*Journal of Intelligent and Robotic Systems* **18:** 105–126, 1997.



# Obstacle Collision Detection Using Best Ellipsoid Fit

ELON RIMON
*Dept. of Mechanical Engineering Technion, Israel Institute of Technology*

STEPHEN P. BOYD
*Dept. of Electrical Engineering Stanford University*

(Received: 11 September 1995; in final form: 27 August 1996)

**Abstract.** This paper describes a method for estimating the distance between a robot and its surrounding environment using best ellipsoid fit. The method consists of the following two stages. First we approximate the detailed geometry of the robot and its environment by minimum-volume enclosing ellipsoids. The computation of these ellipsoids is a convex optimization problem, for which efficient algorithms are known. Then we compute a conservative distance estimate using an important but little known formula for the distance of a point from and $n$-dimensional ellipse. The computation of the distance estimate (and its gradient vector) is shown to be an eigenvalue problem, whose solution can be rapidly found using standard techniques. We also present an incremental version of the distance computation, which takes place along a continuous trajectory taken by the robot. We have implemented the proposed approach and present some preliminary results.

**Key words:** ellipsoid fit, geometric approximation, collision detection.

## 1. Introduction

Knowledge of the distance between the robot and its environment is central for planning collision-free paths. For example, Khatib [19], and Rimon and Koditschek [31], use distance functions as an artificial potential function, whose gradient vector field indicates a direction of repulsion from the obstacles. They combine repulsive distance functions with an attractive potential at the goal, to generate a navigation vector field suitable for a low-level reactive controller. Another example is the work of Lin and Canny [21] and Rimon and Canny [30]. They construct a network of curves, called a roadmap, in the free configuration space of a robot, by incrementally following the local maxima of distance functions along a suitable sweep direction. Choset and Burdick [4] recently used distance functions to construct a roadmap which generalizes the idea of Voronoi diagram to configuration spaces of any dimension. Distance functions are also useful in computer animation, where the animation software is required to generate smooth, collision-free paths between intermediate frames [1, 16].

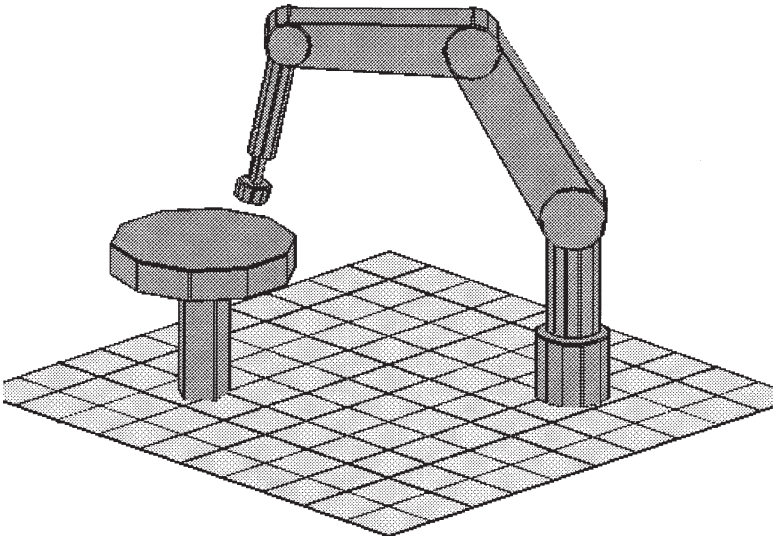

*Figure 1*. Five-link arm serving a rotating feeding table.

The traditional approach to distance computation consists of modeling the various bodies by polyhedra and detecting collision by computing the distance between the polyhedra. This approach has a long history in robotics. Work in this area ranges from rapid numerical techniques [2, 13, 32], through various approximate and hierarchical approaches [6, 8, 15], to closed-form distance estimate expressions [3, 7, 23]. However, the description of complex shapes with a polyhedral model often requires many planar faces and edges. For example, each link in Figure 1 consists of about 30 planar faces. Each face is represented in turn by triangles, and a total of 60–100 are used to describe each link. Higher quality rendering of complex or curved shapes requires hundreds of planar faces. It is therefore important to seek a simplification of polyhedral objects into simpler shapes for which the distance can be estimated efficiently. Traditional approaches use various polyhedral approximations for the simplification [10, 18, 24]. This paper proposes the use of minimum-volume enclosing ellipsoids for shape simplification, and an efficient eigenvalue-based technique for distance estimation.

The minimum-volume enclosing ellipsoid of a convex body (or of the convex hull of a set of measurement points) is called the *Löwner–John ellipsoid*. Fitting bodies with their Löwner–John ellipsoid is an intuitively appealing means to lump their detailed geometry into a single quadratic surface [26]. The robot links, for example, are often convex and their Löwner–John ellipsoid gives a suitable conservative approximation. The obstacles, such as the table in Figure 1, are often non-convex and they should be first decomposed into overlapping convex shapes.

The attractive feature of the Löwner–John ellipsoid is that it can be computed as a *convex optimization* problem, for which efficient general algorithms are known. We give the ingredients necessary for implementing one such algorithm, called the *ellipsoid algorithm*, in the computation of the Löwner–John ellipsoid.

Next we present a technique for computing a conservative distance estimate, or a generalized distance, between the ellipsoids. The computation is based on a formula for the distance of a point from an ellipse of any dimension. The implementation of the formula requires computation of the *minimal eigenvalue* of an auxiliary matrix constructed from the geometrical data. This allows computation of the distance estimate with standard and rapid eigenvalue techniques. Let us describe the distance estimate that we compute.

Let $\mathsf{E}_l$ be an ellipsoid surrounding a link, and let $\mathsf{E}_b$ be an ellipsoid surrounding an obstacle part. Then the solution of the associated eigenvalue problem yields the points $x^*$ on $\mathsf{E}_b$ which is the closest to $\mathsf{E}_l$ with respect to the ellipsoidal level-surfaces surrounding $\mathsf{E}_l$. Second application of the point-to-ellipse formula yields the point $y^*$ on $\mathsf{E}_l$ which is the closest to $x^*$. The function $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b) \triangleq \mathsf{k}x^* - y^*\mathsf{k}$, termed the *free margin* of $\mathsf{E}_l$ with respect to $\mathsf{E}_b$, is positive when $\mathsf{E}_l$ and $\mathsf{E}_b$ are disjoint, and zero when they touch. Moreover, its value provides an estimate for the amount of separation between $\mathsf{E}_l$ and $\mathsf{E}_b$. Another property of $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$ is that it is differentiable (actually smooth), and its gradient vector is also computable as an eigenvalue problem. The gradient vector is often used by reactive planning systems to determine a direction of repulsion from the obstacles, and its rapid computation is another useful property of $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$.

Finally, the robot typically computes its distance from the environment while it moves along a continuous path. We therefore present an approach for the incremental computation of $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$ along a continuous trajectory. It is interesting to compare our incremental distance computation between enclosing ellipsoids with the incremental computation of the distance between polyhedra. A recent incremental algorithm of Lin and Canny [22], computes the Euclidean distance between two polyhedra by tracking their closest two features. Its time complexity is constant, except when the identity of the closest two features changes. Then it becomes linear in the number of vertices (in the worst case). In contrast, the incremental computation of $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$ always takes constant time. Moreover, it does not require the substantial bookkeeping required to manage the polyhedral features. These gains come, of course, at the expense of using approximate shapes and obtaining only a conservative estimate for the exact Euclidean distance.

This paper is organized as follows. Section 2 describes the computation of the Löwner–John ellipsoid as a convex optimization problem. Section 3 shows how to compute $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$ as an eigenvalue problem. Section 4 discusses the incremental computation of $\mathsf{margin}(\mathsf{E}_l, \mathsf{E}_b)$ along a trajectory taken by the robot. The concluding section discusses several open issues associated with this work.

## 2. Computation of the Optimal Ellipsoid

Every compact set with non-empty interior possesses a unique minimum-volume enclosing ellipsoid, called the *Löwner–John ellipsoid*, or the L–J ellipsoid [17]. This section focuses on the computation of the L–J ellipsoid of a convex polyhedron $\mathsf{P}$, a problem which we call the *L–J problem*. After a short review of convex optimization and the ellipsoid algorithm, we show that the L–J problem is convex. Then we give the ingredients necessary for implementing the ellipsoid algorithm to the L–J problem.

### 2.1. REVIEW OF CONVEX OPTIMIZATION

Convex optimization is concerned with computing the minimum of a convex function subject to convex constraints. By definition, a scalar-valued function $f : \mathbb{R}^k \to \mathbb{R}$ is a *convex function* if $f(\lambda x_1 + (1-\lambda)x_2) \leqslant \lambda f(x_1) + (1-\lambda) f(x_2)$ for all $x_1, x_2 \in \mathbb{R}^k$ and $0 \leqslant \lambda \leqslant 1$. Convexity of $f$ implies that for any constant $c$, the sublevel set $\{x \in \mathbb{R}^k : f(x) \leqslant c\}$, if non-empty, is a convex set. Let $\phi : \mathbb{R}^k \to \mathbb{R}$ be a convex function. A *convex optimization* problem is to compute $x^*$ that minimizes $\phi(x)$, subject to the constraint that $x$ be in $\mathsf{K}$, where $\mathsf{K} \subseteq \mathbb{R}^k$ is a convex region. One general algorithm used to solve convex optimization problems is the *ellipsoid algorithm*[⋆] [14, Ch. 3]. It requires that $\mathsf{K}$ be described by a convex function $\psi : \mathbb{R}^k \to \mathbb{R}$ in the form $\mathsf{K} = \{x \in \mathbb{R}^k : \psi(x) \leqslant 0\}$.

The ellipsoid algorithm produces a sequence of $k$-ellipsoids whose centers, denoted $x_i$, converge to $x^*$. At the $i$th step it computes a separating hyperplane passing through $x_i$ for one of the two convex regions

$$\{x : \phi(x) \leqslant \phi(x_i)\} \quad \text{or} \quad \{x : \psi(x) \leqslant \psi(x_i)\}.$$

The separating hyperplane is not necessarily unique, since the boundary of the region may have a sharp corner at $x_i$. In such situations the hyperplane's normal becomes a subgradient. By definition, if $f : \mathbb{R}^k \to \mathbb{R}$ is convex, but not necessarily differentiable, a vector $n \in \mathbb{R}^k$ is a *subgradient* of $f$ at $x$ if it satisfies

$$f(z) \geqslant f(x) + n^T(z - x) \quad \text{for all } z \in \mathbb{R}^k. \tag{1}$$

Note that the set of subgradient vectors reduces to the classical gradient vector when $f$ is differentiable at $x$.

The ellipsoid algorithm is initialized with a $k$-ellipsoid containing $x^*$, for instance, a large ball containing $\mathsf{K}$. At the $i$th step the center $x_i$ is checked against the constraint function $\psi$. If the constraint is violated ($\psi(x_i) > 0$), a separating hyperplane passing through $x_i$ for the region $\{x : \psi(x) \leqslant \psi(x_i)\}$ is computed. Otherwise, a separating hyperplane passing through $x_i$ for the region $\{x : \phi(x) \leqslant \phi(x_i)\}$ is computed. Clearly, one side of the resulting hyperplane contains the entirety of $\mathsf{K}$ in the first case, and it contains the minimizer $x^*$

[⋆] Nesterov and Nemirovsky's recent interior-point algorithms promise to be much faster [25].

in the other case. In both cases the $(i+1)$th $k$-ellipsoid is computed as the minimum-volume ellipsoid that contains the intersection of the $i$th ellipsoid with the halspace determined by the separating hyperplane. A closed-form formula for this $k$-ellipsoid is known, and this formula is used by the algorithm.

A property of convex functions, $\phi$ in our case, allows the algorithm to compute an upper bound on the error $|\phi(x_i) - \phi(x^*)|$. The algorithm terminates when $|\phi(x_i) - \phi(x^*)| \leqslant \varepsilon$ and $\psi(x_i) \leqslant 0$, where $\varepsilon > 0$ is the desired accuracy. In practice it is preferable to minimize $\log(\phi)$ instead of $\phi$. The minimizer is still $x^*$, while $\varepsilon$ becomes a desired relative accuracy: $|\log(\phi(x_i)) - \log(\phi(x^*))| = |\log(\phi(x_i)/\phi(x^*))| \leqslant \varepsilon$. Selecting $\varepsilon = \log(\rho)$ terminates the algorithm with relative error $\rho$.

The run-time of the ellipsoid algorithm can be estimated as follows. At each step the volume of the new $k$-ellipsoid reduces by a fixed factor. Using this factor, it can be shown that the number of steps $K$ required to achieve $\varepsilon$-accurate solution satisfies $K \leqslant 2k^2 \log(c/\varepsilon)$. In this expression $c$ is a constant and $k$, the dimension of the ambient space, is fixed for a given problem. Note that the bound on $K$ grows slowly with the accuracy $\varepsilon$. Note, too, that each step of the algorithm requires an evaluation of $\psi(x_i)$. In the L–J problem this operation is linear in the number of vertices of the polyhedron $\mathsf{P}$. Letting $m$ be the number of vertices, we have that $K \leqslant 2k^2 \log(c/\varepsilon)m$.

## 2.2. THE LÖWNER–JOHN PROBLEM IS CONVEX

We show in this section that the computation of the L–J ellipsoid of a polyhedron $\mathsf{P} \subset \mathbb{R}^n$ is a convex optimization problem. This fact is mentioned in [25, p. 229], but we provide a complete analysis of the problem with new proofs for the key facts. First we introduce some notation. Let $y = (y_1, \ldots, y_n)$ and $x = (x_1, \ldots, x_{n+1})$ be points in $\mathbb{R}^n$ and $\mathbb{R}^{n+1}$, respectively. Any $n$-ellipsoid can be described by its center, a vector $y_0 \in \mathbb{R}^n$, and a matrix $Y \in \mathbb{R}^{n\times n}$, which is symmetric and positive definite (a condition written as $Y > 0$). This ellipsoid, denoted $\mathsf{E}^n(y_0, Y)$, is given by $\mathsf{E}^n(y_0, Y) = \{y \in \mathbb{R}^n\colon\ (y - y_0)^T Y (y - y_0) \leqslant 1\}$.

The basic idea is to embed the polyhedron $\mathsf{P}$ in $\mathbb{R}^{n+1}$, in the $n$-dimensional plane at height $x_{n+1} = 1$. Then to compute the minimum-volume $(n+1)$-ellipsoid centered at $0 \in \mathbb{R}^{n+1}$, $\mathsf{E}^{n+1}(0, X)$, which contains the embedded $\mathsf{P}$. We show below that the latter problem is convex. Finally, the resulting $(n+1)$-ellipsoid determines the optimal $n$-ellipsoid by intersecting it with the hyperplane $x_{n+1} = 1$ (Figure 2). This process is summarized in the following theorem.

THEOREM 1. *Let* $\mathsf{E}^{n+1}(0, X^*)$ *be the* minimum volume $(n+1)$*-ellipsoid centered at the origin which contains the embedded* $\mathsf{P}$. *Then* $\mathsf{E}^n(y^*, Y^*)$*, obtained by intersecting* $\mathsf{E}^{n+1}(0, X^*)$ *with* $x_{n+1} = 1$*, is the* minimum volume $n$*-ellipsoid containing* $\mathsf{P}$.

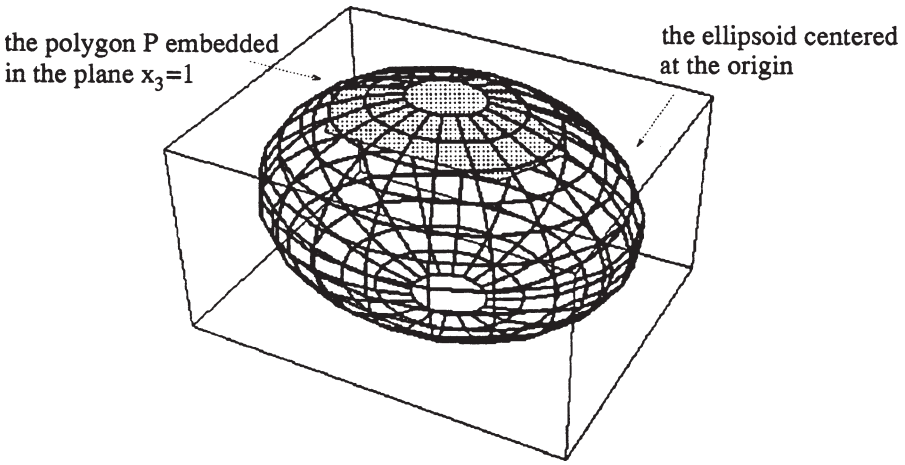


*Figure 2*. The optimal $\mathbf{n}$-ellipsoid is obtained by intersecting the optimal $(\mathbf{n}+1)$-ellipsoid centered at the origin with $\mathbf{x}_{n+1} = 1$ ($\mathbf{n} = 2$ in the figure).

*Moreover, if* $\mathsf{E}^{n+1}(0, X)$ *is* $\varepsilon$-optimal *with respect to* $\mathsf{E}^{n+1}(0, X^*)$, *then* $\mathsf{E}^{n}(y, Y)$, *obtained by intersecting* $\mathsf{E}^{n+1}(0, X)$ *with* $x_{n+1} = 1$, *is* $\varepsilon$-optimal *with respect to* $\mathsf{E}^{n}(y^*, Y^*)$.

A proof of the theorem appears in the appendix. Henceforth we focus on the computation of the optimal $(n+1)$-ellipsoid $\mathsf{E}^{n+1}(0, X)$. The optimization variables are the $k$ distinct entries of the symmetric matrix $X$. Since $X$ is $(n+1)$ ð $(n+1)$, $k = (n+1)(n+2)/2 = 10$ in our case where $n = 3$. The optimization problem is thus to minimize the volume of $\mathsf{E}^{n+1}(0, X)$ over the symmetric matrices $X$, subject to the constraints that $X$ be positive-definite and that the embedded $\mathsf{P}$ be inside $\mathsf{E}^{n+1}(0, X)$.

First let us verify that the volume of $\mathsf{E}^{n+1}(0, X)$ is a convex function of the entries of $X$. This volume is given by $\beta_{n+1}/(\det X)^{1/2}$, where $\beta_{n+1}$ is the volume of the unit ball in $\mathbb{R}^{n+1}$. Since $\log\left(\beta_{n+1}/(\det X)^{1/2}\right) = \log \beta_{n+1} - (\log(\det X))/2$, and since $\beta_{n+1}$ is constant, the objective function is $\phi(X) = -\log(\det X)$. To show that $\phi$ is a convex function, we will use the following fact. To check that a function $f(x)$ is convex, it suffices to verify that $f(\frac{1}{2}(x_1+x_2)) \leqslant \frac{1}{2}(f(x_1) + f(x_2))$ [9, p. 118]. Thus it suffices to show that:

$$-\log\left(\det \tfrac{1}{2}(X_1 + X_2)\right) \leqslant -\tfrac{1}{2}\left(\log\det(X_1) + \log\det(X_2)\right). \tag{2}$$

The following lemma asserts that $\phi(X)$ has this property. The proof of the lemma appears in the appendix.

LEMMA 2.1. *If* $A$ *and* $B$ *are positive-definite matrices,*

$$\det\left(\tfrac{1}{2}(A+B)\right) \geqslant \det(AB)^{1/2},$$

*where* $(AB)^{1/2}$ *is the square-root matrix*$^\star$ *of* $AB$.

*The logarithm of both sides gives the inequality* (2). *Hence*

$$\phi(X) = -\log(\det X)$$

*is a* convex function *over the matrices* $X > 0$.

Next we verify that the constraint $X > 0$ is convex. $X > 0$ iff its minimal eigenvalue is positive. This is equivalent to the condition $\lambda_{\max}(-X) < 0$, where $\lambda_{\max}(-X)$ denotes the largest eigenvalue of the negated matrix $-X$. By definition, $\lambda_{\max}(-X) = \max\{v^T[-X]v\}$ over all unit-magnitude vectors $v$. For each fixed vector $v$, $v^T[-X]v$ is linear in the entries of $X$ and is therefore convex with respect to $X$. Moreover, it is known that the maximum of a family of convex functions is itself convex [5, p. 47]. Hence $\lambda_{\max}(-X)$ is a convex function, and the constraint $X > 0$ is therefore convex.

Last we verify that the containment of the embedded $\mathsf{P}$ in $\mathsf{E}^{n+1}(0, X)$ is a convex constraint. Let $v_1, \ldots, v_m$ be the vertices of $\mathsf{P}$, where $v_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$. In general, $\mathsf{P}$ is contained in $\mathsf{E}^{n+1}(0, X)$ iff the vertices of $\mathsf{P}$ are in $\mathsf{E}^{n+1}(0, X)$. Since $\mathsf{E}^{n+1}(0, X) = \{x \in \mathbb{R}^{n+1}\colon\ x^T X x \leqslant 1\}$, the containment condition is $\max_{i=1,\ldots,m}\{\overline{v}_i^T X \overline{v}_i\} \leqslant 1$, where $\overline{v}_i = (v_i, 1)$ is the $i$th vertex of $\mathsf{P}$ embedded in $x_{n+1} = 1$. But this constraint is the maximum of functions each of which is linear in $X$. Hence it is a convex function of $X$, and the containment constraint is also convex. To summarize, the following optimization problem is convex. Minimize $\phi(X)$, where

$$\phi(X) = -\log(\det X), \tag{3}$$

subject to $\psi_1(X) < 0$ and $\psi_2(X) \leqslant 0$, where

$$\psi_1(X) \triangleq \lambda_{\max}(-X) \quad \text{and} \quad \psi_2(X) \triangleq \max_{i=1,\ldots,m} \{\overline{v}_i^T X \overline{v}_i\} - 1. \tag{4}$$

### 2.3. COMPUTING THE LÖWNER–JOHN ELLIPSOID

This section provides the details necessary for implementing the ellipsoid algorithm to the L–J problem. Recall that every step of the ellipsoid algorithm requires the construction of a separating hyperplane. To compute a normal vector to this hyperplane, we need expressions for the gradient of $\phi(X)$ and for the subgradient of $\psi_1(X)$ and $\psi_2(X)$. First note that tangent vectors in the space of symmetric matrices $X$ are also symmetric matrices. Let $V$ denote these tangent matrices. Gradients (and subgradients) in this space are symmetric matrices, denoted $G$, that map the tangent matrices $V$ to the reals. It can be verified that in general $G$

$^\star$ Given a positive-definite matrix $\mathbf{P}$, $\mathbf{P}^{1/2}$ is the positive-definite matrix satisfying $\mathbf{P} = \mathbf{P}^{1/2}\mathbf{P}^{1/2}$.

acts on $V$ according to the rule: $V \mapsto \mathrm{tr}(G^T V)$, where tr denotes the trace of a matrix. First consider the function $\phi$. It is smooth, and the identity

$$X^{-1} = \frac{1}{\det(X)} [X_{ij}]^T,$$

where $[X_{ij}]$ is the matrix of cofactors of $X$, implies that its gradient, denoted $G_\phi$ is:

$$G_\phi = \nabla(-\log(\det X)) = -X^{-1}.$$

Next consider the function $\psi_1$. According to Equation (1), its subgradient, denoted $G_{\psi_1}$, must satisfy

$$\lambda_{\max}(-Z) \geqslant \lambda_{\max}(-X) + \mathrm{tr}(G_{\psi_1}^T(Z - X)) \quad \text{for all symmetric matrices } Z. \quad (5)$$

Let $u_0$ be a unit-magnitude eigenvector of $-X$ corresponding to its *maximal* eigenvalue. The following inequality is satisfied by all symmetric matrices $Z$,

$$\begin{aligned} \lambda_{\max}(-Z) \geqslant u_0^T[-Z]u_0 &= u_0^T[-X]u_0 - u_0^T(Z - X)u_0 \\ &= \lambda_{\max}(-X) - u_0^T(Z - X)u_0. \end{aligned}$$

But $u_0^T(Z - X)u_0$ is a scalar, and is equal to $\mathrm{tr}(u_0^T(Z - X)u_0)$. In general, $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. Hence $\mathrm{tr}(u_0^T(Z - X)u_0) = \mathrm{tr}(u_0 u_0^T(Z - X))$. Comparison with (5) yields that $G_{\psi_1} = -u_0 u_0^T$. Finally consider $\psi_2$. It has the form $\psi_2 = \max_{i=1,\ldots,m}\{\psi_{2i}\} - 1$, where each $\psi_{2i}$ is a convex function in the entries of $X$. In general, the subgradient of the maximum over a family of convex functions $\psi_{2i}$ is the convex combination of the subgradients of those $\psi_{2i}$ which attain the maximum value [5, p. 47]. In our case it suffices to find one vertex $\overline{v}_{i_0}$ at which $\overline{v}_i^T X \overline{v}_i$ attains its maximum, and a subgradient for $\psi_2$ would be the matrix $G_{\psi_2} = \overline{v}_{i_0}\overline{v}_{i_0}^T$.

A summary of the implementation of the ellipsoid algorithm to the L–J problem follows. We will use the following notation. $(X_i, \Pi_i) \in (\mathbb{R}^k, \mathbb{R}^{k\times k})$ denotes the center and matrix of the $i$th $k$-ellipsoid in the ellipsoid algorithm. For simplicity, $X_i$ also represents the matrix of the $i$th $(n+1)$-ellipsoid, $\mathcal{E}^{n+1}(0, X_i)$. We replace $\lambda_{\max}(-X_i)$ by the equivalent expression $\lambda_{\min}(X_i)$. We will also use the following stack notation: if $G \in \mathbb{R}^{n\times n}$ is a matrix, $G^s$ denotes the $n^2 \times 1$ vector obtained by stacking the columns of $G$ over each other. The $n^2 \times 1$ vector $h_i$ denotes the subgradient of $\psi_1$ or $\psi_2$, the vector $g_i$ denotes the gradient of $\phi$, and $\tilde{g}$ denotes the normal vector to the separating hyperplane. Some additional details concerning the algorithm appear at the end of this section.

$X_1, \Pi_1$ an initial $k$-ellipsoid that contains the minimum;

$i \leftarrow 0$;

repeat {

$i \quad i+1$;
if $(\lambda_{\min}(X_i) \leqslant 0)$ { /* $X_i$ is not positive definite */
  compute eigenvector $u$ for $\lambda_{\min}(X_i)$;
  $h_i = (-uu^T)^s$; /* compute the subgradient $G_{\psi_1}$ of $\psi_1$ */
  $\tilde{g} \quad h_i/\sqrt{h_i^T \Pi_i h_i}$;
} else {
  if $((v_{j_0}^T, 1)X_i \begin{pmatrix} v_{j_0} \\ 1 \end{pmatrix} > 1$ for some $1 \leqslant j_0 \leqslant m)$ { /* $X_i$ is infeasible */
    $h_i = \left( \begin{pmatrix} v_{j_0} \\ 1 \end{pmatrix} (v_{j_0}^T, 1) \right)^s$; /* compute the subgradient $G_{\psi_2}$ of $\psi_2$ */
    $\tilde{g} \quad h_i/\sqrt{h_i^T \Pi_i h_i}$;
  } else { /* $X_i$ is feasible */
    $g_i = \nabla(-\log(\det X_i)) = (-X_i^{-1})^s$; /* compute the gradient $G_\phi$ of $\phi$ */
    $\tilde{g} \quad g_i/\sqrt{g_i^T \Pi_i g_i}$;
  }
}
$X_{i+1} \quad X_i - \frac{1}{K+1}\Pi_i \tilde{g}$; /* compute the $(i+1)$th $k$-ellipsoid */
$\Pi_{i+1} \quad \frac{K^2}{K^2-1}\left(\Pi_i - \frac{2}{K+1}\Pi_i \tilde{g}\tilde{g}^T \Pi_i\right)$;
} until $((\lambda_{\min}(X_i) > 0)$ and (no $j_0$ exists) and $(\sqrt{g_i^T \Pi_i g_i} \leqslant \varepsilon))$.

*Example*. We have implemented the ellipsoid algorithm and computed the L–J ellipsoids for the objects in the scene of Figure 1. The scene includes a 5-link robot arm serving a rotating feeding table. The L–J ellipsoid was computed for each link with relative error of $\rho = 1.001$, and the results are shown in Figure 3. The table was first decomposed into two convex pieces, consisting of its leg and top. The L–J ellipsoids for these pieces was computed with the same relative error of $\rho = 1.001$, and the results are also shown in Figure 3. In our implementation, we have made no special effort to optimize our code, and a typical run time is 2–3 seconds per object. It is worth mentioning an exact algorithm for computing the L–J ellipsoid by Post [27], which is quadratic in the number of vertices of $\mathsf{P}$. The $\varepsilon$-accurate algorithm used here is linear in the number of vertices.

We conclude with a mention of several details concerning the algorithm. First we discuss the formula for the $(i+1)$th $k$-ellipsoid used in the algorithm. Recall that this is the minimum-volume ellipsoid surrounding the intersection of the $i$th $k$-ellipsoid with the halfspace determined by the separating hyperplane. A formula for this ellipsoid is known [14], and we used this formula to compute

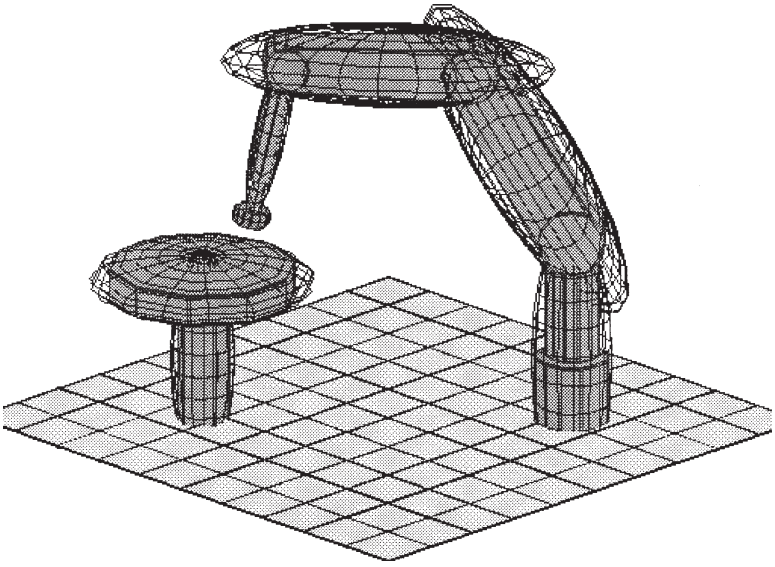

*Figure 3*. The Löwner–John ellipsoids of the scene in Figure 1.

the center, $X_{i+1}$, and the matrix, $\Pi_{i+1}$, of the $(i+1)$th $k$-ellipsoid. Note that this formula depends on the data of the $i$th $k$-ellipsoid, $X_i$ and $\Pi_i$, and on the normal to the separating hyperplane, $\tilde{g}$. Next we give a convenient choice of an initial $k$-ellipsoid which contains the minimum. Let $0 < r < R$ be the radii of two balls in $\mathbb{R}^n$, such that the $r$-ball is contained in the polyhedron $\mathsf{P}$ and the $R$-ball contains it. Then a choice of the initial center $X_1$ as the $(n+1)$ ð $(n+1)$ matrix $X_1 = \mathsf{diag}(1/(1+R^2))$, would give a feasible $(n+1)$-ellipsoid containing the embedded $\mathsf{P}$. As for the initial $k$ ð $k$ matrix $\Pi_1$, it is shown in [29] that a good initial choice for it would be $\Pi_1 = \mathsf{diag}(((n+1)\mathsf{max}\{1, 1/r^2\})^2)$. Finally, the termination condition is derived as follows. For any convex problem, the algorithm stops when $|\phi(x_i) - \phi(x^*)| \leqslant \varepsilon$ and the constraints are satisfied. An upper bound on the error $|\phi(x_i) - \phi(x^*)|$ is obtained as follows. Let $g_i$ be a subgradient of $\phi$ at $x_i$. By definition of a subgradient (Equation (1)),

$$\phi(x^*) \geqslant \phi(x_i) + g_i^T(x^* - x_i),$$

where $x^*$ is the point where $\phi$ attains its minimum. But $x^*$ lies in the $i$th $k$-ellipsoid $\mathsf{E}_i$. Hence

$$\phi(x_i) - \phi(x^*) \leqslant -g_i^T(x^* - x_i) \leqslant \max_{x \in \mathcal{E}_i} \{-g_i^T(x - x_i)\}.$$

The expression for the maximum on the right side is $(g_i^T \Pi_i g_i)^{1/2}$, and this expression is used in the algorithm.

## 3. A Generalized Distance Between Ellipsoids

The generalized distance between two ellipsoids $\mathsf{E}_1$ and $\mathsf{E}_2$, $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_1)$, is based on the distance of a point from an $n$-dimensional ellipse. We first show that the distance of a point from an ellipse can be computed as an eigenvalue problem. Then we show how to compute $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_1)$. Simulation results at the end of the section show that $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_1)$ can be effectively computed using a standard eigenvalue routine.

### 3.1. THE DISTANCE OF A POINT FROM AN $n$-ELLIPSE

We wish to compute the minimal Euclidean distance of a point $x_0 \in \mathbb{R}^n$ from an $n$-dimensional ellipsoid, $\mathsf{E}(a, A) = \{x \in \mathbb{R}^n : (x - a)^T A(x - a) \leqslant 1\}$, where $A > 0$. It is convenient to assume that $x_0$ lies outside of $\mathsf{E}^n(a, A)$, so that the closest point necessarily lies on the boundary of $\mathsf{E}^n(a, A)$. Further, we may assume that $x_0$ is at the origin of $\mathbb{R}^n$. The problem is thus to minimize the function $\phi(x) = \|x\|$ subject to the constraint $(x - a)^T A(x - a) = 1$. The solution $x^* \in \mathsf{E}^n(a, A)$ must satisfy the following Lagrange multiplier rule:

$$x^* = \lambda A(x^* - a) \quad \text{for some scalar } \lambda. \tag{6}$$

Equivalently, $x^* = \lambda[\lambda A - I]^{-1} Aa$, where $I$ is the identity matrix. It follows that $x^* - a = [\lambda A - I]^{-1} a$. Substituting for $x^* - a$ in the constraint $(x - a)^T A(x - a) = 1$ gives the following $2n$-degree polynomial in the unknown $\lambda$,

$$\tilde{a}^T \big[\widetilde{A} - \lambda I\big]^{-2} \tilde{a} = 1, \tag{7}$$

where $\widetilde{A} = A^{-1}$ and $\tilde{a} = A^{-1/2} a$. The two Equations (6) and (7) form a system of $n + 1$ scalar equations in the unknowns $x^* \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. The particular root $\lambda$ which corresponds to $x^*$ is characterized by the following identity.

LEMMA 3.1 ([11]). *Let* $(x_i, \lambda_i)$ *and* $(x_j, \lambda_j)$ *be two solutions of Equations* (6)–(7). *Then*

$$\phi^2(x_j) - \phi^2(x_i) = \|x_j\|^2 - \|x_i\|^2 = \tfrac{1}{2}(\lambda_j - \lambda_i)(x_j - x_i)^T A(x_j - x_i).$$

Since $(x_j - x_i)^T A(x_j - x_i) \geqslant 0$, the root of the polynomial (7) which corresponds to $x^*$ must be the *minimal* real root of the polynomial (7). Our goal now is to show that this minimal root can be computed as an eigenvalue problem. We will use for this purpose the method of Gander *et al*. [12]. Consider the following two new variables $w$ and $z$,

$$w \triangleq \big[\widetilde{A} - \lambda I\big]^{-2} \tilde{a} \quad \text{and} \quad z \triangleq \big[\widetilde{A} - \lambda I\big]^{-1} \tilde{a}. \tag{8}$$

Expressing Equations (7)–(8) in terms of $w$, $z$, and $\lambda$, gives the following system of equations

$$\begin{aligned} &\tilde{a}^T w = 1 \\ &[\widetilde{A} - \lambda I] z = \tilde{a} \\ &[\widetilde{A} - \lambda I] w - z = 0. \end{aligned} \tag{9}$$

Substituting $\tilde{a}^T w = 1$ into the right side of the second equation gives

$$[\widetilde{A} - \lambda I] z = [\tilde{a}\tilde{a}^T] w. \tag{10}$$

Combining the third equation in (9) with (10) yields an eigenvalue problem:

$$\begin{bmatrix} \widetilde{A} & -I \\ -\tilde{a}\tilde{a}^T & \widetilde{A} \end{bmatrix} \begin{pmatrix} w \\ z \end{pmatrix} = \lambda \begin{pmatrix} w \\ z \end{pmatrix}. \tag{11}$$

Let $M$ denote the $2n \times 2n$ matrix in (11), and let $\lambda_{\min}(M)$ be its eigenvalue with minimal real part, termed the *minimal eigenvalue*. We show in the appendix that $\lambda_{\min}(M)$ is precisely the minimal real root of the polynomial (7). Thus, the distance of a point $x$ from an $n$-dimensional ellipse $\mathsf{E}^n(a, A)$ is given by the formula

$$\mathsf{dst}(x, \mathsf{E}^n(a, A)) = \|x - x^*\|,$$

where $x^* = \lambda_{\min}(M)[\lambda_{\min}(M)A - I]^{-1} A a$.

### 3.2. CONVERSION TO DISTANCE ESTIMATE BETWEEN ELLIPSOIDS

We have shown that the distance of a point from an $n$-dimensional ellipse can be computed as an eigenvalue problem. Now we compute the following distance estimate between two $n$-ellipsoids, $\mathsf{E}_1 = \mathsf{E}^n(b, B)$ and $\mathsf{E}_2 = \mathsf{E}^n(c, C)$. First we compute the point $x^* \in \mathsf{E}_2$ at which the ellipsoidal level surfaces surrounding $\mathsf{E}_1$ first touch the ellipsoid $\mathsf{E}_2$. Next we compute the point $y^* \in \mathsf{E}_1$ which is the closest to $x^*$. The resulting distance estimate is then $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2) = \|x^* - y^*\|$.

To compute $x^*$, we assume that $b$, the center of $\mathsf{E}_1$, lies outside of $\mathsf{E}_2$. In that cae $x^*$ lies on the boundary of $\mathsf{E}_2$. We therefore minimize the function $\phi(x) = (x - b)^T B (x - b)$ subject to the constraint $(x - c)^T C (x - c) = 1$. First we apply a coordinate transformation that translates $b$ to the origin and deforms $\mathsf{E}_1$ to a unit ball:

$$\bar{x} = B^{1/2}(x - b) \quad \text{or} \quad x = b + B^{-1/2}\bar{x}, \tag{12}$$

where $B^{1/2}$ is the square-root matrix of $B$.

In the new coordinates, $\bar{x}$, the problem is to minimize the function $\phi(\bar{x}) = \|\bar{x}\|^2$ such that $(\bar{x} - \bar{c})^T \overline{C} (\bar{x} - \bar{c}) = 1$, where $\overline{C} = B^{-1/2} C B^{-1/2}$ and $\bar{c} = B^{1/2}(c - b)$. This is exactly the point-to-ellipse distance problem of the previous section, which yields the point $x^* \in \mathsf{E}_2$. The point $y^* \in \mathsf{E}_1$ closest to $x^*$ is obtained by second application of the point-to-ellipse distance formula. The computation of $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$ is summarized in the following proposition.

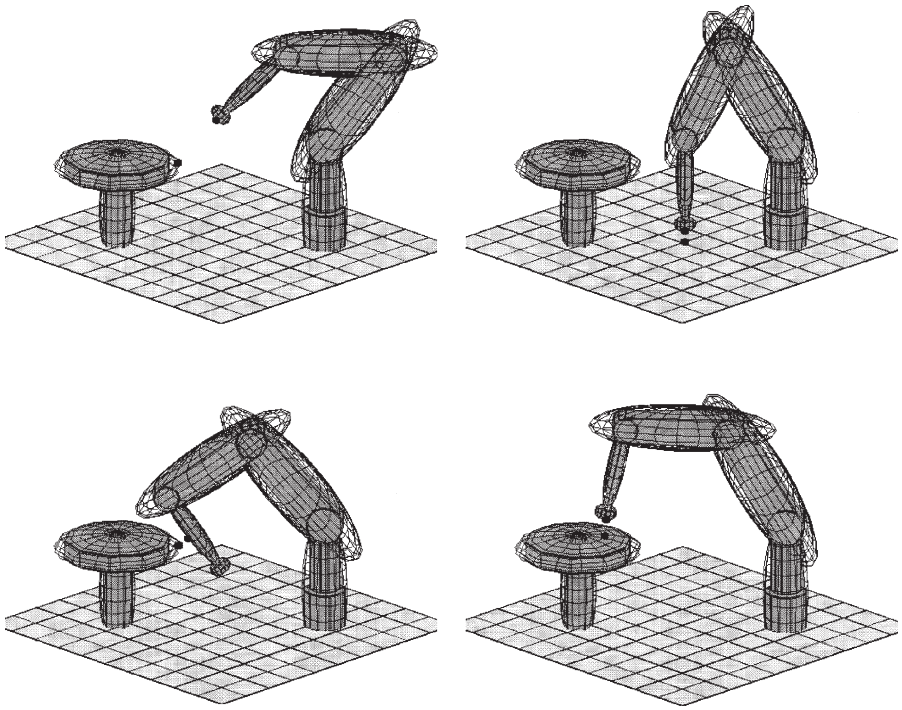

*Figure 4*. Snapshots showing the points x* and y*, that determine margin($\mathsf{E}_1; \mathsf{E}_2$).

PROPOSITION 3.2. *Given two n-ellipsoids* $\mathsf{E}_1 = \mathsf{E}^n(b, B)$ *and* $\mathsf{E}_2 = \mathsf{E}^n(c, C)$, *the point* $x^* \in \mathsf{E}_2$ *is:* $x^* = b + \lambda_1 B^{-1/2}[\lambda_1 I - \widetilde{C}]^{-1}\bar{c}$, *where* $\widetilde{C} = \overline{C}^{-1}$ *and* $\lambda_1$ *is the minimal eigenvalue of the* $2n \times 2n$ *matrix*

$$\begin{bmatrix} \widetilde{C} & -I \\ -\tilde{c}\tilde{c}^T & \widetilde{C} \end{bmatrix} \quad \text{such that } \tilde{c} = \overline{C}^{-1/2}\bar{c}. \tag{13}$$

*The point* $y^* \in \mathsf{E}_1$ *is:* $y^* = x^* + \mu_1[\mu_1 I - \widetilde{B}]^{-1}b$, *where* $\widetilde{B} = B^{-1}$ *and* $\mu_1$ *is the minimal eigenvalue of the* $2n \times 2n$ *matrix*

$$\begin{bmatrix} \widetilde{B} & -I \\ -\tilde{b}\tilde{b}^T & \widetilde{B} \end{bmatrix} \quad \text{such that } \tilde{b} = \overline{B}^{-1/2}b. \tag{14}$$

*The resulting distance estimate,* margin$(\mathsf{E}_1, \mathsf{E}_2) = \|x^* - y^*\|$, *is positive when* $\mathsf{E}_1$ *and* $\mathsf{E}_2$ *are disjoint, and zero when* $\mathsf{E}_1$ *and* $\mathsf{E}_2$ *touch such that their interiors are disjoint*.

Another useful property of margin$(\mathsf{E}_1, \mathsf{E}_2)$ is that its gradient can also be computed as an eigenvalue problem. We discuss this computation in the next section.

*Example*. We have implemented the formula for $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$. In our implementation, we used the standard *QR method* [28, pp. 385–392] to compute the minimal eigenvalue of the two matrices appearing in Equations (13)–(14). Then we computed the distance estimate between the L–J ellipsoids surrounding the robot links and the ones surrounding the table in Figure 3. Snapshots showing the points $x^*$ and $y^*$ for several arm positions are shown in Figure 4. The QR method computes all the eigenvalues of each $2n \times 2n$ matrix in roughly $4(2n)^3$ operations, where $n = 3$ in our case. The average time for one distance computation was 1.5 msec, of which 1.0 msec were taken by the minimal eigenvalue computation (on a Silicon Graphics Indigo machine).

## 4. Incremental Computation of $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$

The computation time of $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$ can be significantly improved by tracking the minimal eigenvalue of the matrices associated with $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$, along a continuous trajectory taken by the robot. To explain the principle of the method, we focus on the incremental computation of the distance from the origin of $\mathbb{R}^n$ to an ellipsoid $\mathsf{E}^n(a, A)$ whose data, $(a, A)$, varies continuosly with time. Let $M$ be the $2n \times 2n$ matrix associated with $(a, A)$ (Equation (11)):

$$M = \begin{bmatrix} \widetilde{A} & -I \\ -\tilde{a}\tilde{a}^T & \widetilde{A} \end{bmatrix} \quad \text{where } \widetilde{A} = A^{-1},\ \tilde{a} = A^{-1/2}a. \tag{15}$$

Suppose that the minimal eigenvalue of $M$ at the previous trajectory point has been computed. We wish to use it in the computation of the minimal eigenvalue at the current trajectory point.

One particular eigenvalue technique, called the *inverse iteration method* [28, p. 394], is ideally suited for this purpose. It is initialized with an estimate for the minimal eigenvalue of $M$, denoted $\widehat{\lambda}_1$, and an estimate for the corresponding eigenvector, denoted $\widehat{v}(0)$. The eigenvector estimate is then repetitively updated according to the rule: $\widehat{v}(k) = [M - \widehat{\lambda}_1 I]^{-1}\widehat{v}(k-1)$. Since $M$ and $[M - \widehat{\lambda}_1 I]^{-1}$ share the same eigenvectors, an eigenvalue $\lambda_i$ of $M$ corresponds to an eigenvalue $1/(\lambda_i - \widehat{\lambda}_1)$ of $[M - \widetilde{\lambda}_1 I]^{-1}$. Hence, if $\lambda_1$ is the true minimal eigenvalue of $M$ and $\widehat{\lambda}_1$ is closer to $\lambda_1$ than to any other eigenvalue of $M$, the eigenvector estimate, $\hat{v}(k)$, converges exponentially to the true eigenvector. However, for this method to be of practical use, we must characterize the distance between $\lambda_1$ and the other eigenvalues of $M$. The following theorem, which is new to our knowledge, asserts that $\lambda_1$ is the only eigenvalue of $M$ in the left-hand side of the complex plane. In the following, $\mathsf{Re}\{z\}$ denotes the real part of a complex number $z$.

THEOREM 2. *Let $\lambda_1$ be the minimal eigenvalue of the matrix $M$ defined in Equation* (15) (*i.e. $\lambda_1$ is the eigenvalue with minimal real part*). *Then $\lambda_1$ is* nega-

tive real *whenever the center, $a$, of $\mathsf{E}^n(a, A)$ lies outside the unit ball. Moreover, all the other eigenvalues of $M$ satisfy*

$$\mathsf{Re}\{\lambda_i\} \geqslant \kappa_1 > 0 \quad \textit{for } i = 2, \ldots, 2n,$$

*where $\kappa_1$ is the minimal eigenvalue of $\widetilde{A} = A^{-1}$.*

The proof of the theorem appears in the appendix. The theorem guarantees that $\lambda_1$ is isolated from the other eigenvalues of $M$ by a disc of radius larger than $|\lambda_1| + \kappa_1$. Thus, all the initial guesses of $\widehat{\lambda}_1$ in the disc of half that radius will converge to $\lambda_1$. The theorem also implies the following corollary, which asserts that $\lambda_1$ is a smooth function of the geometrical data.

COROLLARY 4.1. *The minimal eigenvalue of $M$, $\lambda_1$, (and consequently* margin $(\mathsf{E}_1, \mathsf{E}_2)$*) is a* real analytic function *of the geometrical data, $(a, A)$. Moreover, the gradient of $\lambda_1$ is:*

$$\frac{\partial \lambda_1(a, A)}{\partial a} = -\frac{1}{\alpha}\,\widetilde{A}T^{-2}a \quad \textit{and} \quad \frac{\partial \lambda_1(a, A)}{\partial A} = \frac{1}{\alpha}\,T^{-2}\tilde{a}\tilde{a}^T T^{-2}, \tag{16}$$

*where $T = \widetilde{A} - \lambda_1 I$ and $\alpha = \tilde{a}^T T^{-3}\tilde{a}$.*

*Proof.* According to Theorem 2, $\lambda_1$ is an isolated root of the characteristic polynomial of $M$. It is known from function theory that an isolated root of a polynomial is an analytic function of its coefficients [20, p. 125]. The formula for the gradient is derived by implicit differentiation of the polynomial in Equation (7), $\tilde{a}^T[\widetilde{A} - \lambda_1(a, A)I]^{-2}\tilde{a} = 1$, which is satisfied by $\lambda_1$ according to Lemma B.1 in the appendix. □

The gradient of $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$ can be computed by applying the chain rule to $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_2)$ and then using (16). While we do not explicitly derive the gradient formula, it is clearly forthcoming. Let us discuss now an important implementation detail of the inverse iteration method. Efficient implementation of the inverse iteration method requires the following formula:

$$[M - \widehat{\lambda}_1 I]^{-1} = \begin{bmatrix} PQ & P \\ QPQ - I & QP \end{bmatrix},$$

where $Q = \widetilde{A} - \widehat{\lambda}_1 I$ and

$$P = \left[Q^2 - \tilde{a}\tilde{a}^T\right]^{-1} = Q^{-2} + \frac{1}{1 + \tilde{a}^T Q^{-2}\tilde{a}}\,Q^{-2}\tilde{a}\tilde{a}^T Q^{-2}.$$

If $R$ is the orthogonal matrix that diagonalizes $\widetilde{A}$ into a diagonal matrix $\Lambda$, $\Lambda = R^T\widetilde{A}R$, knowledge of $R$ allows computation of $Q^{-2}$ as $Q^{-2} = R(\Lambda - \widehat{\lambda}_1 I)^{-2}R^T$. Hence the inverse iteration method requires only a series of matrix-vector multiplications, each taking $n^2$ operations. The number of steps required

by the inverse iteration method to achieve an $\varepsilon$-accurate estimate is $c\log(1/\varepsilon)$, where $c$ is a constant which depends on $\widehat{\lambda}_1$. The total number of operations required by the inverse iteration method is thus $cn^2\log(1/\varepsilon)$, compared with $4(2n)^3$ for the QR method.

*Example*. We have implemented the inverse iteration method and tested it on the L–J ellipsoids of Figure 3. We took the robot configurations at the snapshots shown in Figure 4 as vertices in the robot's joint space. Then we connected these vertices by straight lines and arbitrarily discretized each line by 100 equally spaced points. The robot was then moved along the resulting joint-space trajectory, and we computed its distance from the obstacles using the inverse iteration method. The computation at each trajectory point was terminated when the magnitude of the error between successive eigenvector estimates became less than $10^{-6}$. Using an Indigo machine, the average time for one distance computation was .7 msec, of which 0.35 msec were taken by the eigenvalue computation.

To summarize, the incremental computation of the minimal eigenvalue with the inverse iteration method is significantly faster than the static computation. However, we have merely depicted the principle of the method, and other improvements can be introduced. For example, the derivative of $\lambda_1$ as a function of the robot configuration can be computed from Equation (16). The derivative can then provide an estimate for the variation in $\lambda_1$ between successive trajectory points. Comparison of the variation with the conversion disc will ensure that the step size is sufficiently small for the inverse iteration method to converge to the correct solution.

## 5. Concluding Discussion

We have shown that the optimal enclosing ellipsoid, $\mathsf{E}$, of a polyhedron, $\mathsf{P}$, can be effectively computed as a convex optimization problem, and that it often provides a suitable approximation for convex polyhedra (Figure 3). An important topic for further research is concerned with measures for the tightness of the optimal $\mathsf{E}$ about $\mathsf{P}$. The following two properties of the optimal ellipsoid provide some measure for the tightness of the fitting. The first is that $\mathsf{P}$ always contains the ellipsoid formed by shrinking $\mathsf{E}$ from its center by a factor of $1/n$ [17]. The other is that $\mathsf{E}$ touches at least $n+1$ vertices of $\mathsf{P}$, such that the center of $\mathsf{E}$ lies at the geometrical center of these vertices [17]. While there is a limit to the tightness of an approximation by ellipsoids, its advantage lies in the rapid computation of the distance estimate.

We have also shown that a generalized distance between two ellipsoids can be computed as an eigenvalue problem. Standard techniques for computing eigenvalues are so rapid that they are generally considered to be 'closed form' solutions. Moreover, the classical collision detection approaches which compute the distance between polyhedra take time which is linear in the number of vertices (in

worst case) [22], while $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_1)$ always takes constant time to compute, regardless of the complexity of the underlying polyhedra. The ellipsoid approach is thus useful for real-time applications, where basic computations are expected to end within a specific time interval.

Further, polyhedral approaches which compute the exact distance must retain the entire data structure of the polyhedra. However, typical polyhedra contain hundreds of faces, while the ellipsoid approach uses a single quadratic enclosing surface as the data structure. Of course, the simpler data structure comes at the price of obtaining only a conservative distance estimate. In the future, collision-detection systems should be able to exploit the advantages of each representational approach according to the task characteristics, taking into account competing factors such as computation time and desired accuracy.

## Appendix A: Details Concerning the Optimal Ellipsoid

This appendix contains proofs of statements made about the optimal ellipsoid. We begin with a proof of Theorem 1, for which the following lemma will be needed. The lemma gives a formula for the $n$-ellipsoid obtained by intersecting the $(n+1)$-ellipsoid $\mathsf{E}^{n+1}(0, X)$ with $x_{n+1} = 1$. We will use the following block partition of the $(n+1)$ ð $(n+1)$ matrix $X$:

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix},$$

where $X_{11}$ is $n$ ð $n$, $X_{12}$ is $n$ ð 1, and $X_{22}$ is a scalar.

LEMMA A.1. *Let $\mathsf{E}^n(y_0, Y)$ be the $n$-ellipsoid obtained by intersecting the $(n+1)$-ellipsoid $\mathsf{E}^{n+1}(0, X)$ with $x_{n+1} = 1$. Then $y_0$ and $Y$ are given by*

$$y_0 = -X_{11}^{-1} X_{12} \quad \textit{and} \quad Y = \frac{1}{1 - \rho(X)} X_{11},$$

*where $\rho(X) = X_{22} - X_{12}^T X_{11}^{-1} X_{12}$. Moreover, $\rho(X)$ satisfies $0 < \rho(X) < 1$.*

*Proof.* By definition, all the points $x = (x_1, \ldots, x_{n+1})$ in $\mathsf{E}^{n+1}(0, X)$ at height $x_{n+1} = 1$ satisfy

$$(y^T \ 1) \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \begin{pmatrix} y \\ 1 \end{pmatrix} \leqslant 1 \quad \text{where } y = (x_1, \ldots, x_n).$$

Expanding this inequality gives

$$\begin{aligned} &(y^T \ 1) \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \begin{pmatrix} y \\ 1 \end{pmatrix} \\ &\quad = \mathsf{k} X_{11}^{1/2} y \mathsf{k}^2 + 2 (X_{11}^{1/2} y)^T X_{11}^{-1/2} X_{12} + X_{22} \\ &\quad = (y + X_{11}^{-1} X_{12})^T X_{11} (y + X_{11}^{-1} X_{12}) + X_{22} - X_{12}^T X_{11}^{-1} X_{12} \leqslant 1, \end{aligned}$$

or equivalently,

$$(y + X_{11}^{-1}X_{12})^T \left[ \frac{1}{1 - (X_{22} - X_{12}^T X_{11}^{-1} X_{12})} X_{11} \right] (y + X_{11}^{-1}X_{12}) \leqslant 1.$$

A proof that $\rho(X) = X_{22} - X_{12}^T X_{11}^{-1} X_{12}$ satisfies $0 < \rho(X) < 1$ appears in [29]. □

We are now ready to prove the theorem.

THEOREM 1. *Let* $\mathsf{E}^{n+1}(0, X^*)$ *be the* minimum volume $(n+1)$*-ellipsoid centered at the origin which contains the enbedded* $\mathsf{P}$. *Then* $\mathsf{E}^n(y^*, Y^*)$, *obtained by intersecting* $\mathsf{E}^{n+1}(0, X^*)$ *with* $x_{n+1} = 1$, *is the* minimum volume $n$*-ellipsoid containing* $\mathsf{P}$.

*Moreover, if* $\mathsf{E}^{n+1}(0, X)$ *is* $\varepsilon$-optimal *with respect to* $\mathsf{E}^{n+1}(0, X^*)$, *then* $\mathsf{E}^n(y, Y)$, *obtained by intersecting* $\mathsf{E}^{n+1}(0, X)$ *with* $x_{n+1} = 1$, *is* $\varepsilon$-optimal *with respect to* $\mathsf{E}^n(y^*, Y^*)$.

*Proof.* According to Lemma A.1, each $(n+1)$-ellipsoid $\mathsf{E}^{n+1}(0, X)$ determines an $n$-ellipsoid $\mathsf{E}^n(y(X), Y(X))$ at height $x_{n+1} = 1$, as well as a scalar $\rho(X)$ such that $0 < \rho(X) < 1$. On the other hand, every $n$-ellipsoid $\mathsf{E}^n(y, Y)$ and a scalar $0 < \rho < 1$ determine an $(n+1)$-ellipsoid $\mathsf{E}^{n+1}(0, X)$ which coincides with $\mathsf{E}^n(y, Y)$ in the hyperplane $x_{n+1} = 1$. It can be verified that the matrix $X$ is given in terms of $\mathsf{E}^n(y, Y)$ and $\rho$ by the formula:

$$X(y, Y, \rho) = \begin{bmatrix} (1 - \rho)Y & -(1 - \rho)Yy \\ -(1 - \rho)(Yy)^T & \rho + (1 - \rho)y^T Yy \end{bmatrix}. \tag{17}$$

Let us compute the determinant of $X(y, Y, \rho)$,

$$\begin{aligned} &\det\left(X(y, Y, \rho)\right) \\ &\quad = (1 - \rho)^n \det Y \left\{ \rho + (1 - \rho)y^T Yy - (1 - \rho)y^T Yy \right\} \\ &\quad = \rho(1 - \rho)^n \det Y, \end{aligned} \tag{18}$$

where we have used the identity

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A) \det(D - CA^{-1}B).$$

Recall that the $(n+1)$-ellipsoid attains minimal volume when $\det(X)$ is maximal. Using Equation (18), the maximum of $\det(X)$ can be written in the following way

$$\max_{\rho, Y} \left\{ \rho(1 - \rho)^n \det Y \right\} = \max_{\rho} \left\{ \rho(1 - \rho)^n \right\} \max_{Y} \left\{ \det Y \right\}.$$

It follows that $\det X(y, Y, \rho)$ attains its maximal value precisely when $\det Y$ attains its maximal value. Hence $\mathsf{E}^{n+1}(0, X)$ attains minimal volume precisely

when the corresponding $n$-ellipsoid, $\mathsf{E}^n(y, Y)$, attains its minimal volume. A proof that and $\varepsilon$-optimal $(n+1)$-ellipsoid gives an $\varepsilon$-optimal $n$-ellipsoid appears in [29]. □

LEMMA 2.1. *If $A$ and $B$ are positive-definite matrices,*

$$\det\left(\tfrac{1}{2}(A+B)\right) \geqslant \det(AB)^{1/2}.$$

*Proof.* Since $A > 0$, we may write it as $A = A^{1/2}A^{1/2}$. Use of the identity $\det(P_1P_2) = \det(P_1)\det(P_2)$ gives

$$\det\left(\tfrac{1}{2}(A+B)\right) = \det(A^{1/2})\det\left(\tfrac{1}{2}(I + A^{-1/2}BA^{-1/2})\right)\det(A^{1/2}).$$

Since $\det(AB)^{1/2} = \det(B^{1/2}A^{1/2}) = \det(B^{1/2})\det(A^{1/2})$, it suffices to show that

$$\det\left(\tfrac{1}{2}(I + A^{-1/2}BA^{-1/2})\right) \geqslant \det(B^{1/2})\det(A^{-1/2}). \tag{19}$$

But $A^{-1/2}BA^{-1/2} = (B^{1/2}A^{-1/2})^T B^{1/2}A^{-1/2}$ and $\det(B^{1/2})\det(A^{-1/2}) = \det(B^{1/2}A^{-1/2})$. Hence the matrices in both sides of (19) share the same eigenvectors. Letting $\sigma_1, \ldots, \sigma_n$ denote the eigenvalues of the matrix $B^{1/2}A^{-1/2}$, (19) can be written as: $1/2\prod_{i=1}^n(1+\sigma_i^2) \geqslant \prod_{i=1}^n \sigma_i$. But $\prod_{i=1}^n(1+\sigma_i^2) \geqslant 1+\prod_{i=1}^n \sigma_i^2$. Hence it suffices to show that $1+\prod_{i=1}^n \sigma_i^2 \geqslant 2\prod_{i=1}^n \sigma_i$. But this is the inequality $(1+\prod_{i=1}^n \sigma_i)^2 \geqslant 0$, which is always satisfied. □

## Appendix B: Details Concerning $\mathsf{margin}(\mathsf{E}_1, \mathsf{E}_1)$

This appendix contains the proof of Theorem 2, for which we will need the following lemma.

LEMMA B.1. *The characteristic polynomial of the $2n \times 2n$ matrix $M$ defined in Equation* (15) *is: $q(\lambda) = q_1^2(\lambda)q_2(\lambda)$, where $q_1(\lambda)$ is the characteristic polynomial of $\widetilde{A}$, and $q_2(\lambda) = 1 - \tilde{a}^T[\widetilde{A} - \lambda I]^{-2}\tilde{a}$.*

*Proof.* We have to compute the polynomial $\det(M - \lambda I)$ where $\lambda$ is a scalar. First we exchange the top $n$ rows with the bottom $n$ rows,

$$\det(M - \lambda I) = (-1)^n \det\begin{bmatrix} -\tilde{a}\tilde{a}^T & \widetilde{A} - \lambda I \\ \widetilde{A} - \lambda I & -I \end{bmatrix}.$$

Next we use the identity

$$\det\begin{bmatrix} X & Y \\ W & Z \end{bmatrix} = \det(Z)\det(X - YZ^{-1}W),$$

$$\begin{aligned} \det(M - \lambda I) &= \det\left(-\tilde{a}\tilde{a}^T + [\widetilde{A} - \lambda I]^2\right) \\ &= \det^2(\widetilde{A} - \lambda I)\det\left(-[\widetilde{A} - \lambda I]^{-1}\tilde{a}\tilde{a}^T[\widetilde{A} - \lambda I]^{-1} + I\right). \end{aligned}$$

Finally, we use the identity $\det(uv^T + I) = 1 + u \cdot v$ to obtain the result. □

Note that the polynomial $q_2(\lambda)$ in the lemma is precisely the $2n$-degree polynomial (7). Moreover, it is shown below that $q_2(\lambda)$ has a root which lies to the left of all the roots of the polynomial $q_1(\lambda)$. Hence the minimal eigenvalue of $M$ is the minimal root of the polynomial (7).

THEOREM 2. *Let $\lambda_1$ be the minimal eigenvalue of $M$ (i.e. the eigenvalue with minimal real part). Then $\lambda_1$ is* negative real *whenever the center, $a$, of $\mathsf{E}^n(a, A)$ lies outside the unit ball. Moreover, all the other eigenvalues of $M$ satisfy*

$$\mathrm{Re}\{\lambda_i\} \geqslant \kappa_1 > 0 \quad \textit{for } i = 2, \ldots, 2n,$$

*where $\kappa_1$ is the minimal eigenvalue of $\widetilde{A} = A^{-1}$.*

*Proof.* First we establish that $\det(M) < 0$. This would imply that $M$ has at least one negative real eigenvalue. Using the characteristic polynomial given by Lemma B.1, $\det(M) = q(\lambda)|_{\lambda=0} = \det^2(\widetilde{A})(1 - \tilde{a}^T A^2 \tilde{a})$, where we substituted $A^{-1}$ for $\widetilde{A}$. Substituting $A^{-1/2}a$ for $\tilde{a}$ gives: $\det(M) = \det^2(\widetilde{A})(1 - \|a\|^2)$. Since $\|a\| > 1$ by hypothesis, $\det(M) < 0$. Next we show that $\lambda_1$ is isolated. Let $\kappa_1, \ldots, \kappa_n$ be the eigenvalues of $\widetilde{A}$, and let $R$ be the orthogonal matrix which diagonalizes $\widetilde{A}$, $R^T \widetilde{A} R = \mathrm{diag}(\kappa_1, \ldots, \kappa_n)$. Then the characteristic polynomial of $M$, written in terms of a complex variable $z$, is:

$$\begin{aligned} q(z) &= q_1^2(z)(1 - \tilde{a}^T R[R^T \widetilde{A} R - \lambda I]^{-2} R^T \tilde{a}) \\ &= \left( \prod_{i=1}^{n} (z - \kappa_i)^2 \right) \left( 1 - \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(z - \kappa_i)^2} \right), \end{aligned}$$

where $\bar{a} = R^T \tilde{a}$. Let $\mathsf{R}$ be the region of the complex plane defined by $\mathrm{Re}\{z\} < \kappa_1$ where, recall, $\kappa_1 > 0$ is the minimal eigenvalue of $\widetilde{A}$ and $\mathrm{Re}\{z\}$ is the real part of $z$. Since $\mathrm{Re}\{z\} - \kappa_i < 0$ for $i = 1, \ldots, n$ and $z \in \mathsf{R}$, $q_1$ is non-zero in $\mathsf{R}$. Only $q_2$ can vanish in $\mathsf{R}$. Let $\mathrm{Im}\{z\}$ denote the imaginary part of $z$, and let $\bar{z}$ denote the complex conjugate of $z$. The $i$th summand of $q_2$ can be written as

$$\begin{aligned} \frac{\bar{a}_i^2}{(z - \kappa_i)^2} &= \bar{a}_i^2 \frac{(\bar{z} - \kappa_i)^2}{(|z|^2 + \kappa_i^2)^2} \\ &= \bar{a}_i^2 \frac{(\mathrm{Re}\{z\} - \kappa_i)^2 - \mathrm{Im}\{z\}^2 + 2j(\mathrm{Re}\{z\} - \kappa_i)\mathrm{Im}\{z\}}{(|z|^2 + \kappa_i^2)^2}, \end{aligned}$$

where $j = \sqrt{-1}$. It follows that the imaginary part of $q_2(z)$ is

$$\mathrm{Im}\{q_2(z)\} = 2\mathrm{Im}\{z\} \sum_{i=1}^{n} \bar{a}_i^2 \frac{\mathrm{Re}\{z\} - \kappa_i}{(|z|^2 + \kappa_i^2)^2}. \tag{20}$$

Since $\mathrm{Re}\{z\} - \kappa_i < 0$ for $i = 1, \ldots, n$, it follows from (20) that the roots of $q_2$ in $\mathbb{R}$ must be real. Let $q_2(s) = 1 - \sum_{i=1}^{n} \bar{a}_i^2/(s - \kappa_i)^2$ be the restriction of $q_2$ to the reals. Then $q_2(s)$ has exactly one root in the interval $(-1, \kappa_1)$. This observation is made in [12] and is a consequence of the following two facts. The first is that $\lim_{s \to -\infty} q_2(s) = 1$ and $\lim_{s \to \kappa_1^-} q_2(s) = -1$. It implies that $q_2(s)$ has at least one root in $(-1, \kappa_1)$. The second is that

$$\frac{\mathrm{d}}{\mathrm{d}s} q_2(s) = 2 \sum_{i=1}^{n} \bar{a}_i^2/(s - \kappa_i)^3.$$

Since $s - \kappa_i < 0$ for $s \in (-1, \kappa_1)$, the derivative is negative and $q_2(s)$ is strictly decreasing. Hence $q_2(s)$ has exactly one root in $(-1, \kappa_1)$, which must be negative since $\det(M) < 0$. □

## References


1. Baraff, D.: Curved surfaces and coherence for nonpenetrating rigid body simulation, *Computer Graphics (Proc. SIGGRAPH)* **24**(4) (1990), 19–28.
2. Bobrow, J. E.: A direct minimization approach for obtaining the distance between convex polyhedra, *International Journal of Robotics Research* **8**(3) (1989), 65–76.
3. Canny, J. F.: Collision detection for moving polyhedra, *IEEE Transactions on PAMI* **8** (1986), 200–209.
4. Choset, H. and Burdick, J. W.: Sensor based planning, Part ii: Incremental construction of the generalized voronoi graph, in: *IEEE Int. Conference on robotics and Automation*, Nagoya, Japan, 1995, pp. 1643–1649.
5. Clarke, F. H.: *Optimization and Nonsmooth Analysis*, SIAM Publications, 1990.
6. Dobkin, D. P. and Kirpatrick, D. G.: Fast detection of polyhedral intersection, *Theoretical Computer Science* **27** (1983), 241–253.
7. Donald, B. R.: Motion planning with six degrees of freedom, Research Report AI-TR-791, Artificial Intelligence Lab., MIT, 1984.
8. Faverjon, B.: Hierarchical object models for efficient anti-collision algorithms, in: *IEEE Int. Conference on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 333–340.
9. Fleming, W.: *Functions of Several Variables*, Springer-Verlag, New York, 1987.
10. Flynn, P. J. and Jain, A. K.: CAD-based computer vision – from CAD models to relational graphs, *IEEE Transactions on PAMI* **13**(2) (1991), 114–132.
11. Gander, W.: Least squares with a quadratic constraint, *Numerische Mathematik* **36** (1981), 291–307.
12. Gander, W., Golub, G. H., and Matt, U.: A constrained eigenvalue problem, *Linear Algebra and Its Applications*, 1989, pp. 815–839.
13. Gilbert, E. G., Johnson, D. W., and Keerthi, S. S.: A fast procedure for computing the distance between objects in three-dimensional space, *IEEE Transactions on Robotics and Automation* **4** (1988), 193–203.
14. Grotschel, M., Lovasz, L., and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.
15. Henrich, D. and Cheng, X.: Fast distance computation for on-line collision detection with multi-arm robots, in: *IEEE Int. Conference on Robotics and Automation*, Nice, France, 1992, pp. 2514–2519.
16. Hubbard, P.: Interactive collision detection, in: *IEEE Symposium on Research Frontiers in Virtual Reality*, 1993, pp. 24–31.
17. John, F.: Extremum problems with inequalities as subsidiary conditions (1948), in: J. Moser (ed.), *Fritz John Collected Papers,* Ch. 2, Birkhauser, Boston, 1985, pp. 543–560.

18. Kalvin, A. D. and Taylor, R. H.: Superfaces: Polyhedral approximation with bounded error, in: *SPIE Conference on Medical Imaging*, Vol. 2164, Newport Beach, CA, 1994, pp. 1–13.
19. Khatib, O.: Real time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research* **5**(1) (1986), 90–99.
20. Knopp, K.: *Theory of Functions II*, Dover, New York, 1947.
21. Lin, M. C. and Canny, J. F.: An opportunistic global path planner, in: *IEEE Int. Conference on Robotics and Automation*, Cincinnati, OH, 1990, pp. 1554–1558.
22. Lin, M. C. and Canny, J. F.: A fast algorithm for incremental distance calculation, in: *IEEE Int. Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 1008–1014.
23. Lozano-Pérez, T.: Spatial planning: A configuration space approach, *IEEE Transactions on Computers* **32**(2) (1983), 108–120.
24. Martin, R. R. and Stephenson, P. C.: Containment algorithm for objects in rectangular boxes, in: *Theory and Practice of Geometric Modeling*, Springer-Verlag, New York, 1989, pp. 307–325.
25. Nesterov, Y. E. and Nemirovsky, A. S.: *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, Springer-Verlag, New York, 1992.
26. Pentland, A. P.: Perceptual organization and the representation of natural form, in: *Readings in Computer Vision*, Kaufmann, Los Altos, CA, 1987, pp. 680–699.
27. Post, M. J.: Minimum spanning ellipsoids, in: *ACM Symposium on Theory of Computing*, Washington, DC, 1984, pp. 108–116.
28. Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T.: *Numerical Recipes in C*, Cambridge Univ. Press, NY, 1988.
29. Rimon, E. and Boyd, S. P.: Efficient distance computation using best ellipsoid fit, Technical report, Information Systems Laboratory, Stanford University, 1992.
30. Rimon, E. and Canny, J. F.: Construction of c-space roadmaps from local sensory data: What should the sensors look for?, in: *IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, 1994, pp. 117–123.
31. Rimon, E. and Koditschek, D. E.: Exact robot navigation using artificial potential functions, *IEEE Transactions on Robotics and Automation* **8**(5) (1992), 501–518.
32. Zeghloul, S., Rambeaud, P., and Lallemand, J.: A fast distance calculation between convex objects by optimization approach, in: *IEEE Int. Conference on Robotics and Automation* Nice, France, 1992, pp. 2520–2525.