Generalized Penetration Depth Computation based on Kinematical Geometry

Georg Nawratil [a] Helmut Pottmann [a] Bahram Ravani [b]

[a] *Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Wiedner Hauptstrasse 8-10/104, Vienna, A-1040, Austria*

[b] *Department of Mechanical and Aeronautical Engineering, University of California, Davis, CA 95616, USA*

Abstract

The generalized penetration depth PD of two overlapping bodies X and Y is the distance between the given colliding position of X and the closest collision-free Euclidean copy X^{ε} to X according to a distance metric. We present geometric optimization algorithms for the computation of PD with respect to an object-oriented metric S which takes the mass distribution of the moving body X into consideration. We use a kinematic mapping which maps rigid body displacements to points of a 6-dimensional manifold M^6 in the 12-dimensional space $\mathbb{R}^{12}$ of affine mappings equipped with S. We formulate PD as the solution of the constrained minimization problem of finding the closest point on the boundary of the set of all points of M^6 which correspond to colliding configurations. Based on the theory of gliding motions, the closest point with respect to the metric S ($\Rightarrow PD_S$) can be computed with an adapted projected gradient algorithm. We also present an algorithm for the computation of the closest point with respect to the geodesic metric G of M^6 induced by S ($\Rightarrow PD_G$). Moreover we introduce two methods for the computation of a collision-free initial guess and give a physical interpretation of PD_S and PD_G.

Key words: Penetration Depth, Geometric Optimization, Gliding Motions, Kinematics, Distance Function

1 Introduction

Assume two rigid objects X and Y of $\mathbb{R}^3$ are interpenetrating each other. One way to characterize the extent of overlap is the distance between the given

Email address: nawratil@geometrie.tuwien.ac.at (Georg Nawratil).
URL: http://www.geometrie.tuwien.ac.at/nawratil (Georg Nawratil).

colliding position of X and the closest collision-free pose X^{ε} to X according to a distance metric, where ε is a Euclidean map (rigid body motion). Such a measure is called penetration depth. If ε is restricted to the group of spatial translations $ST(3)$ then we speak of translational penetration depth PD^t, where the distance between X and X^{ε} is defined as the Euclidean distance between any corresponding point pair $\mathbf{x}$ and $\mathbf{x}^{\varepsilon}$ of X. Most of the prior work deals with translational penetration depth computation. A detailed review of this topic is given in Zhang et al. (2006).

Zhang et al. (2006) pointed out that PD^t computation is not sufficient for many applications (these include rigid body dynamics simulation, motion planning for dexterous manipulation, workspace analysis and 6-dof haptic rendering) because it does not take the rotational motion into account. Therefore they introduced the nomenclature of generalized penetration depth PD for the case that ε is allowed to be any Euclidean displacement of $SE(3)$.

1.1 Prior work

To the best of our knowledge, the papers of Zhang et al. (2006,2007) are the only works on PD computation. A key issue for the definition of PD is the choice of the distance metric on $SE(3)$, because there exists no naturally introduced metric as in the case of PD^t.

Zhang et al. (2006) used a model dependent distance metric $L(\mathsf{X}^{\alpha}, \mathsf{X}^{\beta})$ for their computation of generalized penetration depth. This metric is defined as follows: Assume $\mathcal{L}_{\alpha}^{\beta}$ denotes the set of all rigid body transformations $\mathbf{c}_t$ with $t \in [0,1]$, $\mathbf{c}_0(\mathsf{X}) = \mathsf{X}^{\alpha}$ and $\mathbf{c}_1(\mathsf{X}) = \mathsf{X}^{\beta}$. If $L_{\alpha}^{\beta}(\mathbf{c}(\mathbf{x}))$ is the trajectory length of a point $\mathbf{x} \in \mathsf{X}$ under $\mathbf{c} \in \mathcal{L}_{\alpha}^{\beta}$, the distance metric of Zhang et al. can be written as:

$$L(\mathsf{X}^{\alpha}, \mathsf{X}^{\beta}) := min\left(\left\{max\left(\left\{L_{\alpha}^{\beta}(\mathbf{c}(\mathbf{x}))\,|\,\mathbf{x} \in \mathsf{X}\right\}\right)\,\middle|\,\mathbf{c} \in \mathcal{L}_{\alpha}^{\beta}\right\}\right). \tag{1}$$

The generalized penetration depth with respect to the above distance metric on $SE(3)$ is denoted by PD_L. Therefore $PD_L(\mathsf{X}, \mathsf{Y})$ equals the minimum of the longest trajectories of X under all possible rigid transformations which separate the overlapping bodies X and Y. The advantage of the used metric is that for convex objects PD_L equals PD^t. Therefore the PD_L computation of two convex bodies can be put down to the computation of PD^t, where many good algorithms are already known (e.g. Cameron (1997), Kim et al. (2002) and Van den Bergen (2001)). These algorithms compute PD^t by calculating the minimum distance from the origin to the surface of the Minkowski sum of the two convex polyhedra.

The L metric is indeed nice in theory but it is not very suitable for practical computation. Due to this fact, Zhang et al. were only able to present algorithms for the computation of the upper and lower bounds of PD_L for non-convex objects. The algorithm for the lower bound of PD_L is based on the convex decomposition of both input models. The maximum value of PD^t between all pairwise combinations of the resulting convex pieces of X and Y gives the lower bound of PD_L. The computation of the upper bound equals a variant of a 3D convex containment problem, which is solved by using linear programming.

Recently, Zhang et al. (2007) also presented an optimization-based algorithm for PD computation with respect to the so called DIST-metric, which is more suitable for computation than the above mentioned one.

1.2 Overview

In section 2 we repeat the kinematic mapping of Hofer et al. (2004), which maps rigid body displacements to points of a 6-dimensional manifold M^6 in the 12-dimensional space $\mathbb{R}^{12}$ of affine mappings equipped with the distance metric S.

In section 3 we define the generalized penetration depth with respect to S, because it allows for a very efficient computation. Therefore this metric was already successfully used for the design of rigid body motions (see e.g. Hofer (2004), Hofer and Pottmann (2004) and Hofer et al. (2004)) as well as for the definition of new performance indices for 6R robots (see Nawratil (2007)).

We formulate generalized penetration depth PD as the solution of the constrained minimization problem of finding the closest point on the boundary of the set of all points of M^6 which correspond to colliding configurations. Based on the theory of gliding motions (repeated in section 2), the closest point with respect to the metric S ($\Rightarrow PD_S$) can be computed with an adapted projected gradient algorithm, which is outlined in subsection 3.1. The complexity analysis is done in subsection 3.2.

In section 4 we also present an algorithm for the computation of the closest point with respect to the geodesic metric G of M^6 induced by S ($\Rightarrow PD_G$). Moreover we introduce two methods for the computation of a collision-free initial guess (see subsection 3.4) and give a physical interpretation of PD_S and PD_G in subsection 3.3 and 4.3, respectively.

2 Fundamentals

Consider a rigid body moving in Euclidean three-space E^3. We use Cartesian coordinates and denote the coordinate vectors of points of the moving system Σ^0 by $\mathbf{x}^0, \mathbf{y}^0, \dots,$ and points of the fixed system Σ by $\mathbf{x}, \mathbf{y}$, and so on. A one-parameter motion Σ^0/Σ is a smooth family of Euclidean congruence transformations depending on a parameter t which can be thought of as time. A point $\mathbf{x}^0$ of Σ^0 is, at time t, mapped to the point $\mathbf{x}(t) = \mathbf{A}(t)\cdot\mathbf{x}^0 + \mathbf{a}_0(t)$ of Σ. If we do not impose orthogonality on the matrix $\mathbf{A}$, we get, for each t, an affine map.

2.1 The kinematic image space

We use a kinematic mapping (see Hofer et al. (2004)) that views affine maps as points in 12-dimensional space $\mathbb{R}^{12}$. For that, we consider the affine map $\mathbf{x} = \alpha(\mathbf{x}^0) = \mathbf{a}_0 + \mathbf{A}\cdot\mathbf{x}^0$ with $\mathbf{x}^0 = (x_1^0, x_2^0, x_3^0)$. If we denote the three column vectors of $\mathbf{A}$ as $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ we can rewrite $\mathbf{x}$ as $\alpha(\mathbf{x}^0) = \mathbf{a}_0 + x_1^0\mathbf{a}_1 + x_2^0\mathbf{a}_2 + x_3^0\mathbf{a}_3$. Now we represent the affine map α by the point $\mathfrak{A} := (\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ in the 12-dimensional affine space $\mathbb{R}^{12}$. The image of the group of Euclidean maps under this kinematic mapping is a 6-dimensional manifold M^6.

Hofer et al. (2004) introduced a meaningful metric in $\mathbb{R}^{12}$, which is based on the following idea. We represent the rigid body X by a finite number of points $\mathbf{x}_1^0, \dots, \mathbf{x}_N^0$. In our case these are the vertices of a triangular mesh $\mathcal{X}$, which approximates the boundary surface of X. By means of this point cloud it is possible to define the distance between any two affine copies X^α and X^β as follows:

$$S(\mathsf{X}^\alpha, \mathsf{X}^\beta)^2 := \sum_{i=1}^{N} \|\alpha(\mathbf{x}_i^0) - \beta(\mathbf{x}_i^0)\|^2. \tag{2}$$

This object-oriented metric only depends on the barycenter $\mathbf{b}_\mathsf{X}$ and the covariance matrix $\mathbf{D}_\mathsf{X}$ with respect to $\mathbf{b}_\mathsf{X}$ given by:

$$\mathbf{b}_\mathsf{X} := N^{-1}\sum_{i=1}^{N}\mathbf{x}_i^0 \quad \text{and} \quad \mathbf{D}_\mathsf{X} := \sum_{i=1}^{N}(\mathbf{x}_i^0 - \mathbf{b}_\mathsf{X})\cdot(\mathbf{x}_i^0 - \mathbf{b}_\mathsf{X})^T. \tag{3}$$

Therefore we can replace the points $\mathbf{x}_1^0, \dots, \mathbf{x}_N^0$ by the six special points $\mathbf{s}_i$ with

$$\mathbf{s}_i := \mathbf{b}_\mathsf{X} + \sqrt{\frac{\lambda_i}{2}}\,\mathbf{d}_i, \qquad \mathbf{s}_{i+3} := \mathbf{b}_\mathsf{X} - \sqrt{\frac{\lambda_i}{2}}\,\mathbf{d}_i, \qquad i = 1, 2, 3 \tag{4}$$

where λ_i denotes an eigenvalue and $\mathbf{d}_i$ a corresponding unit eigenvector of $\mathbf{D}_\mathsf{X}$. This does not change the barycenter, the covariance matrix and the inertia tensor $\mathbf{T}_\mathsf{X} := \mathit{trace}(\mathbf{D}_\mathsf{X})\mathbf{I}_3 - \mathbf{D}_\mathsf{X}$ of X.

The distance between any two points $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathbb{R}^{12}$ is defined as the distance between the corresponding affine copies of the moving body X, i.e.,

$$\|\alpha - \beta\|^2 = \|\mathfrak{A} - \mathfrak{B}\|^2 := S(\mathsf{X}^\alpha, \mathsf{X}^\beta)^2 = \sum_{i=1}^{6} \|\alpha(\mathbf{s}_i) - \beta(\mathbf{s}_i)\|^2. \tag{5}$$

With $\mathfrak{D} := \mathfrak{A} - \mathfrak{B}$ the squared distance can be rewritten as:

$$\|\mathfrak{A} - \mathfrak{B}\|^2 = \|\mathfrak{D}\|^2 = \mathfrak{D}^T \cdot \mathbf{M} \cdot \mathfrak{D} =: \langle \mathfrak{D}, \mathfrak{D} \rangle, \tag{6}$$

where $\mathbf{M}$ is a positive definite 12×12 matrix. $\mathbb{R}^{12}$ equipped with this metric is a Euclidean space E^{12}. If we choose the barycenter as origin in the moving system and the eigenvectors of $\mathbf{D}_{\mathsf{X}}$ as coordinate axes then the six points (4) are given by $(\pm f_1, 0, 0)$, $(0, \pm f_2, 0)$ and $(0, 0, \pm f_3)$. Therefore $\mathbf{M}$ of (6) can be written as

$$\mathbf{M} = diag(6\mathbf{I}_3, 2f_1^2\mathbf{I}_3, 2f_2^2\mathbf{I}_3, 2f_3^2\mathbf{I}_3). \tag{7}$$

In the following we consider tangent spaces of points $\mathfrak{E} \in M^6$. A tangent vector $\mathfrak{T}_{\mathfrak{E}}$ of M^6 at a point $\mathfrak{E} \in M^6$ corresponds in $\mathbb{R}^3$ to a velocity vector field of a Euclidean motion. If we denote the instantaneous screw associated with this velocity vector field by $\underline{\mathbf{q}} = (\mathbf{q}, \widehat{\mathbf{q}})$, the velocity vector of a point $\mathbf{x} \in \mathbb{R}^3$ equals $\mathbf{v}(\mathbf{x}) = \widehat{\mathbf{q}} + \mathbf{q}\times \mathbf{x}$. Therefore the coordinates of $\mathfrak{T}_{\mathfrak{E}}$ in E^{12} at the point $\mathfrak{E} = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in M^6$ are given by:

$$\mathfrak{T}_{\mathfrak{E}} = (\widehat{\mathbf{q}} + \mathbf{q}\times \mathbf{e}_0, \mathbf{q}\times \mathbf{e}_1, \mathbf{q}\times \mathbf{e}_2, \mathbf{q}\times \mathbf{e}_3). \tag{8}$$

The normalized Plücker coordinates $(\mathbf{p}, \widehat{\mathbf{p}})$ of the axis, the angular velocity ω, the translatory velocity $\widehat{\omega}$ and the pitch p of the instantaneous screw $\underline{\mathbf{q}}$ can be reconstructed by

$$\omega = \|\mathbf{q}\|, \quad \widehat{\omega} = \frac{\mathbf{q}\cdot\widehat{\mathbf{q}}}{\omega}, \quad p = \frac{\widehat{\omega}}{\omega} \quad \text{and} \quad (\mathbf{p}, \widehat{\mathbf{p}}) = \frac{1}{\omega}(\mathbf{q}, \widehat{\mathbf{q}} - p\,\mathbf{q}). \tag{9}$$

We denote the 6-dimensional tangent space of $\mathfrak{E} \in M^6$ by $T^6_{\mathfrak{E}}$. A basis of this tangent space is spanned by the six vectors $\mathfrak{T}^i_{\mathfrak{E}} \in \mathbb{R}^{12}$

$$\begin{aligned}
\mathfrak{T}^1_{\mathfrak{E}} &= (0, -e_{03}, e_{02}, 0, -e_{13}, e_{12}, 0, -e_{23}, e_{22}, 0, -e_{33}, e_{32})^T,\\
\mathfrak{T}^2_{\mathfrak{E}} &= (e_{03}, 0, -e_{01}, e_{13}, 0, -e_{11}, e_{23}, 0, -e_{21}, e_{33}, 0, -e_{31})^T,\\
\mathfrak{T}^3_{\mathfrak{E}} &= (-e_{02}, e_{01}, 0, -e_{12}, e_{11}, 0, -e_{22}, e_{21}, 0, -e_{32}, e_{31}, 0)^T,\\
\mathfrak{T}^4_{\mathfrak{E}} &= (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),\\
\mathfrak{T}^5_{\mathfrak{E}} &= (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),\\
\mathfrak{T}^6_{\mathfrak{E}} &= (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
\end{aligned} \tag{10}$$

with $\mathbf{e}_i = (e_{i1}, e_{i2}, e_{i3})^T$ for $i = 0, \ldots, 3$.

2.2 Instantaneous gliding motions

We assume X^{ε} and Y are in contact, where ε is a Euclidean map which corresponds to the point $\mathfrak{C} \in M^6$. Therefore the surface normals of both objects coincide in the contact points $\mathbf{p}_1, \dots, \mathbf{p}_n$. By a well known result from kinematics (see Pottmann and Wallner (2001)), the instantaneous screw $\underline{\mathbf{q}} = (\mathbf{q}, \widehat{\mathbf{q}})$ of the motion such that X^{ε} glides on Y has to meet the n linear gliding constraints

$$h_i(\underline{\mathbf{q}}): \quad \widehat{\mathbf{n}}_i \cdot \mathbf{q} + \mathbf{n}_i \cdot \widehat{\mathbf{q}} = 0 \quad \text{for} \quad i = 1, \dots, n, \tag{11}$$

where $(\mathbf{n}_i, \widehat{\mathbf{n}}_i)$ denote the Plücker coordinates of the common normal in the contact point $\mathbf{p}_i$. Up to first order these are the only conditions on a gliding surface pair, see Pottmann and Ravani (2000). These linear constraints mean nothing else that the n lines $(\mathbf{n}_i, \widehat{\mathbf{n}}_i)$ are contained in the path normal complex of $\underline{\mathbf{q}}$. The velocity vector fields of all possible gliding motions $\underline{\mathbf{q}}$ determined by (11) correspond to the $(6-g)$-dimensional tangent space $T^{6-g}_{\mathfrak{C}}$ of $\mathfrak{C} \in M^6$ with

$$g = rank((\mathbf{n}_1, \widehat{\mathbf{n}}_1), \dots, (\mathbf{n}_n, \widehat{\mathbf{n}}_n)). \tag{12}$$

3 PD computation with respect to the metric of the ambient space

The generalized penetration depth with respect to the metric (5) is denoted by $PD_S(\mathsf{X}, \mathsf{Y})$ and defined as follows:

Definition 1 $PD_S(\mathsf{X}, \mathsf{Y})$ *equals* $S(\mathsf{X}^{id}, \mathsf{X}^{\varphi})$ *divided by* $\sqrt{6}$*, where* φ *is the Euclidean map causing the minimal distance between* X^{id} *and all Euclidean copies* X^{ε} *of* X^{id} *which do not collide with* Y*.*

$$PD_S(\mathsf{X}, \mathsf{Y}) = min\left(\left\{ \frac{S(\mathsf{X}^{id}, \mathsf{X}^{\varepsilon})}{\sqrt{6}} \,\middle|\, interior(\mathsf{X}^{\varepsilon}) \cap \mathsf{Y} = \varnothing \ \wedge \ \varepsilon \in SE(3) \right\}\right)$$

It should be noted that the metric S depends on the moving body which is given by the first element in the bracket. Therefore the object-oriented metric S of $PD_S(\mathsf{X}, \mathsf{Y})$ depends on X in contrary to the metric S of $PD_S(\mathsf{Y}, \mathsf{X})$ which depends on the mass distribution of Y.

If φ is a translation, then $PD_S(\mathsf{X}, \mathsf{Y})$ equals the translational penetration depth PD^t due to the division by $\sqrt{6}$. Generally we can say that X^{φ} and Y are in contact, otherwise X^{φ} could be translated along the vector $t(\mathbf{b}_{\mathsf{X}} - \varphi(\mathbf{b}_{\mathsf{X}}))$ with $t \in \mathbb{R}^+$ into the position $\mathsf{X}^{\overline{\varphi}}$, where the two objects are touching each other. Due to (5) the inequality $S(\mathsf{X}^{id}, \mathsf{X}^{\varphi}) > S(\mathsf{X}^{id}, \mathsf{X}^{\overline{\varphi}})$ holds, which contradicts the definition of φ.

Zhang et al. (2006) proved that $PD_L(\mathsf{X},\mathsf{Y})$ equals PD^t if both given objects X and Y are convex. This theorem does not hold for the generalized penetration depth with respect to the metric S. The following counter example illustrates this:

Counter example:

Assume the moving object X is an ellipsoidal shell given by $x^2/a^2+y^2/b^2+z^2/c^2=1$. The fixed body Y is the half-space $z \leq -b$. Without loss of generality we assume $c > b > 0$. The upright projection of the situation is given in *Fig. 1.* The coordinates of the points $\mathbf{s}_i$ can be computed from the well known covariance matrix

$$\mathbf{D}_{\mathsf{X}} = diag\left(\frac{a^2}{3},\frac{b^2}{3},\frac{c^2}{3}\right)$$

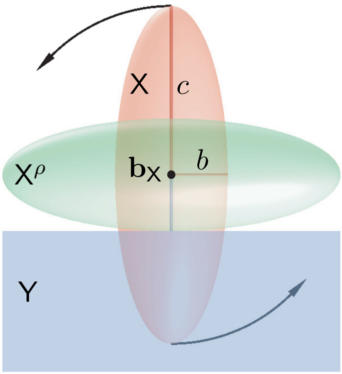


Fig. 1. Counter example

according to (4). If we apply a quarter rotation ρ on X about the x-axis, we get a collision-free configuration X^ρ. Due to the fact that $S(\mathsf{X}^{id},\mathsf{X}^\rho)$ and PD^t only depend on b and c, we have to choose the right values for these variables such that $S(\mathsf{X}^{id},\mathsf{X}^\rho)/\sqrt{6} < PD^t = c-b$ holds. This inequality is fulfilled for e.g. $c=2$ and $b=1$. □

We can also see that $PD_S(\mathsf{X},\mathsf{Y})$ is not necessarily equal to $PD_S(\mathsf{Y},\mathsf{X})$ in this example. If Y is movable, then $S(\mathsf{Y}^{id},\mathsf{Y}^\varepsilon)=\infty$ for all $\varepsilon \in SE(3)\setminus ST(3)$ because Y is unbounded. Therefore $PD_S(\mathsf{Y},\mathsf{X})$ equals $PD^t(\mathsf{Y},\mathsf{X}) = PD^t(\mathsf{X},\mathsf{Y})$, which is for $c=2$ and $b=1$ greater than $PD_S(\mathsf{X},\mathsf{Y})$.

In the following we translate the expression for $PD_S(\mathsf{X},\mathsf{Y})$ given in Def. 1 in terms of the kinematic image space. The kinematic image of X^{id} is the point $\mathfrak{I}=(0,0,0,1,0,0,0,1,0,0,0,1)$ on M^6. $\mathfrak{I}$ belongs to the set $\mathcal{C}$ of points $\mathfrak{E}\in M^6$ which correspond to colliding configurations; i.e.

$$\mathcal{C} = \left\{\mathfrak{E}\,|\, interior(\mathsf{X}^\varepsilon)\cap\mathsf{Y}\neq\varnothing \;\wedge\; \mathfrak{E}\in M^6\right\}. \tag{13}$$

The image point $\mathfrak{F}$ of φ is located on the boundary $\partial\mathcal{C}$ of $\mathcal{C}$, because $\partial\mathcal{C}$ corresponds to all free configurations of X and Y with a surface to surface contact. Therefore $PD_S(\mathsf{X},\mathsf{Y})$ can also be formulated as:

$$PD_S(\mathsf{X},\mathsf{Y}) = min\left(\left\{\frac{\|\mathfrak{I}-\mathfrak{E}\|}{\sqrt{6}}\,\middle|\,\mathfrak{E}\in\partial\mathcal{C}\right\}\right). \tag{14}$$

It follows immediately that the point $\mathfrak{F}$ is the closest normal footpoint on $\partial\mathcal{C}$ with respect to $\mathfrak{I}$. It should also be noted that the tangent space $T^{6-g}_{\mathfrak{E}}$ of $\partial\mathcal{C}$ at a point $\mathfrak{E}\in\partial\mathcal{C}$ is $(6-g)$-dimensional with g from (12) and that $T^{6-g}_{\mathfrak{E}}$

is spanned by the velocity vector fields of all possible gliding motions (see section 2.2). On basis of these preliminary considerations we can formulate the following $PD_S(\mathsf{X},\mathsf{Y})$ algorithm for the computation of $\mathfrak{F}$.

3.1 Algorithm for the computation of PD with respect to the metric of the ambient space

This $PD_S(\mathsf{X},\mathsf{Y})$ algorithm is based on the projected gradient algorithm of Hofer and Pottmann (2004). This is an iterative method consisting of repeated application of the following three steps (see *Fig. 2*). When we discuss one procedure of the iteration, we will denote the current iterate by $\mathfrak{E}^c$, the next iterate by $\mathfrak{E}^+$ and the prior one by $\mathfrak{E}^-$. We assume that a collision-free starting configuration X^{ς} is known. In section 3.4 we present two methods for calculating such an initial guess.

1. If $\mathfrak{E}^c \in \partial\mathcal{C}$ we compute the g^c linearly independent gliding constraints according to (11). We project $\mathfrak{J}$ orthogonally into the tangent space $T^{6-g^c}_{\mathfrak{E}^c}$ of all possible gliding motions, which results in the point $\mathfrak{J}^\perp$.
   If $\mathfrak{E}^c \notin \partial\mathcal{C}$ we project $\mathfrak{J}$ orthogonally into the tangent space $T^6_{\mathfrak{E}^c}$.

2. Compute an appropriate stepsize s and project $\mathfrak{E}^s := \mathfrak{E}^c + s(\mathfrak{J}^\perp - \mathfrak{E}^c)$ back onto M^6, which yields $\mathfrak{E}^\perp$.

3. If $\mathfrak{E}^\perp$ is a collision-free configuration we set $\mathfrak{E}^+ := \mathfrak{E}^\perp$. Otherwise we reduce the stepsize s until $\mathfrak{E}^\perp \in \partial\mathcal{C}$.

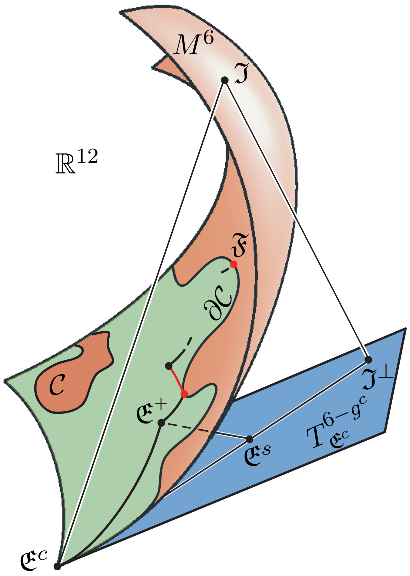


Fig. 2. $PD_S(\mathsf{X},\mathsf{Y})$ algorithm

3.1.1 Preprocessing steps and preliminary considerations

We assume that the two given objects X and Y have smooth boundary surfaces. These boundary surfaces are approximated by triangular meshes $\mathcal{X}$ and $\mathcal{Y}$ with uniformly distributed vertices $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and $\mathbf{y}_1, \ldots, \mathbf{y}_M$, respectively. In preparation for the algorithm we compute the surface normal vectors $\mathbf{n}_{\mathbf{x}_i}$ in each vertex $\mathbf{x}_i$ of $\mathcal{X}$ and $\mathbf{n}_{\mathbf{y}_i}$ in each vertex $\mathbf{y}_i$ of $\mathcal{Y}$. It is assumed that these surface normal vectors are normalized and that they are oriented outward.

The last preparatory work which must be done is the computation of a signed distance field to the boundary surface (triangular mesh $\mathcal{Y}$) of the fixed object Y, where all points inside of Y have a negative distance. We denote the signed distance of a point $\mathbf{z} \in \mathbb{R}^3$ and the object Y by $d(\mathbf{z}, \mathsf{Y})$. Because we approximated the boundary surfaces of X and Y by triangular meshes, we have to define when the moving mesh $\mathcal{X}$ and the fixed mesh $\mathcal{Y}$ are in contact.

Definition 2 *The moving mesh $\mathcal{X}$ and the fixed mesh $\mathcal{Y}$ are in contact if*

$$0 \leq min\left(\{d(\mathbf{x}, \mathsf{Y}) \,|\, \mathbf{x} \in \mathcal{X}\}\right) \leq w, \tag{15}$$

where $w \in \mathbb{R}^+$ is an appropriate small value, called contact value.

Definition 3 *The point $\mathbf{m}$ of $\mathcal{X}$ is a local minimum of the distance function $d(\mathbf{x}, \mathsf{Y})$ if*

$$d(\mathbf{m}, \mathsf{Y}) \leq min\left(\{d(\mathbf{s}, \mathsf{Y}) \,|\, \mathbf{s} \in \mathcal{S}\}\right) \tag{16}$$

where $\mathcal{S}$ denotes the star neighbourhood of $\mathbf{m}$.

With this framework we are able to define the contact points and the contact normals of the moving mesh $\mathcal{X}$ and the fixed mesh $\mathcal{Y}$ as follows:

Definition 4 *Assume the moving mesh $\mathcal{X}$ and the fixed mesh $\mathcal{Y}$ are in contact (Def. 2). If $\mathbf{p}_i \in \mathcal{X}$ is a local minimum of the distance function $d(\mathbf{x}, \mathsf{Y})$ according to Def. 3 and $d(\mathbf{p}_i, \mathsf{Y}) \leq w$, where w is the contact value of Def. 2, then $\mathbf{p}_i$ is defined as a contact point of the moving mesh $\mathcal{X}$ and the fixed mesh $\mathcal{Y}$. The Plücker coordinates $(\mathbf{n}_i, \widehat{\mathbf{n}}_i)$ of the corresponding contact normal are given by:*

$$\mathbf{n}_i := \mathbf{n}_{p_i} \quad \textit{and} \quad \widehat{\mathbf{n}}_i := \mathbf{p}_i \times \mathbf{n}_i \tag{17}$$

where $\mathbf{n}_{\mathbf{p}_i}$ is the normalized surface normal vector of $\mathbf{p}_i$ with respect to $\mathcal{X}$.

3.1.2 Step 1

In the very first step we check via the signed distance field if the current pose of the moving mesh $\mathcal{X}^{\varepsilon^c}$ and $\mathcal{Y}$ are in contact ($\Longleftrightarrow \mathfrak{E}^c \in \partial\mathcal{C}$) with respect to the contact value $w^c := w^-$. If this is the case we compute the contact points $\mathbf{p}_1, \ldots, \mathbf{p}_n \in \mathcal{X}^{\varepsilon^c}$ and the contact normals $(\mathbf{n}_1, \widehat{\mathbf{n}}_1), \ldots, (\mathbf{n}_i, \widehat{\mathbf{n}}_i)$ according to Def. 4. Without loss of generality we assume $d(\mathbf{p}_1, \mathsf{Y}) \leq d(\mathbf{p}_2, \mathsf{Y}) \ldots \leq d(\mathbf{p}_n, \mathsf{Y})$. In the following two cases we redefine the current contact value w^c:

Case A: $g^c = 6$

If there exists no instantaneous gliding motion, i.e. g^c of (12) is equal to 6, then we redefine the contact value w^c such that

$$d(\mathbf{p}_{j^+}, \mathsf{Y}) < w^c < d(\mathbf{p}_j, \mathsf{Y}) \quad \text{with} \tag{18}$$

$$d(\mathbf{p}_{j^+}, \mathsf{Y}) = max\,(\{d(\mathbf{p}_k, \mathsf{Y}) \,|\, d(\mathbf{p}_k, \mathsf{Y}) < d(\mathbf{p}_j, \mathsf{Y}) \text{ for } k = 1, \ldots, j-1)\} \tag{19}$$

holds. Starting with $j = n$ and iterating this procedure until a gliding motion is possible, i.e. $g^c \leq 5$. The next iterate is given by j^+.

Case B: $\|\mathfrak{C}^c - \mathfrak{C}^-\| < b$ **and** $g^c \geq g^- \neq 0$

If both conditions are fulfilled we redefine the contact value w^c as in case A until $g^c < g^-$. The inequality $\|\mathfrak{C}^c - \mathfrak{C}^-\| < b$ is part of the algorithm's stopping condition, where b is a certain threshold. Due to $\|\mathfrak{C}^c - \mathfrak{C}^-\| < b$ there was only a very small change in the last step. Therefore we give the objects $\mathsf{X}^{\varepsilon^c}$ additional degrees of freedom for the current iteration step by reducing the contact value w^c.

In all other cases w^c remains unchanged. Without loss of generality we assume that the Plücker coordinates of the first $g^c \leq 5$ contact normals are linearly independent. If $\mathcal{X}^{\varepsilon^c}$ and $\mathcal{Y}$ are not in contact, then we set g^c equal to zero.

Remark: There is one exceptional case where the iterative procedure of case A and B fails. Assume $d(\mathbf{p}_1, \mathsf{Y}) = d(\mathbf{p}_2, \mathsf{Y}) = \ldots = d(\mathbf{p}_k, \mathsf{Y}) = 0$ with $k > 5$ and $g^c = 6$. Then there exists no instantaneous motion such that $\mathsf{X}^{\varepsilon^c}$ glides on Y. Therefore we cannot bring $\mathsf{X}^{\varepsilon^c}$ closer to X^{id} and so $\mathfrak{C}^c$ is the searched footpoint $\mathfrak{F}$ on the boundary of $\mathcal{C}$. In this case the local minimizer $\mathfrak{F}$ is a singular point of $\partial\mathcal{C}$.

In the general case we project $\mathfrak{I}$ orthogonally onto the tangent space $T^{6-g^c}_{\mathfrak{C}^c}$. This orthogonal projection is equivalent for searching the point $\mathfrak{X} \in T^{6-g}_{\mathfrak{C}^c}$ such that $\|\mathfrak{I} - \mathfrak{X}\|$ is minimal. Therefore we have to minimize the quadratic objective function

$$f(\underline{\mathbf{q}}): \quad \sum_{i=1}^{6} \|\mathbf{s}_i^{\varepsilon^c} + \mathbf{v}(\mathbf{s}_i^{\varepsilon^c}) - \mathbf{s}_i^{id}\|^2 \quad \text{with} \quad \mathbf{v}(\mathbf{s}_i^{\varepsilon^c}) = \widehat{\mathbf{q}} + (\mathbf{q} \times \mathbf{s}_i^{\varepsilon^c}) \tag{20}$$

under the g^c linear constraint equations $h(\underline{\mathbf{q}})$ of (11). We get the minimizer

$\underline{\mathbf{q}} = (\mathbf{q}, \widehat{\mathbf{q}})$ as the solution of the following system of linear equations:

$$\begin{pmatrix} \sum \mathbf{S}_i \mathbf{S}_i^T & \sum \mathbf{S}_i \\ \sum \mathbf{S}_i^T & 6\mathbf{I}_3 \\ \widehat{\mathbf{n}}_1^T & \mathbf{n}_1^T \\ \vdots & \vdots \\ \widehat{\mathbf{n}}_{g^c}^T & \mathbf{n}_{g^c}^T \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \widehat{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \sum \mathbf{s}_i^{\varepsilon^c} \times \mathbf{s}_i^{id} \\ \sum \mathbf{s}_i^{id} - \mathbf{s}_i^{\varepsilon^c} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{21}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix and $\mathbf{S}_i$ the following 3×3 matrix:

$$\mathbf{S}_i := \begin{pmatrix} 0 & -s_i^3 & s_i^2 \\ s_i^3 & 0 & -s_i^1 \\ -s_i^2 & s_i^1 & 0 \end{pmatrix} \quad \text{with} \quad \mathbf{s}_i^{\varepsilon^c} = (s_i^1, s_i^2, s_i^3) \quad \text{for} \quad i = 1, \ldots, 6. \tag{22}$$

Therefore $\mathfrak{I}^\perp$ is given by $\mathfrak{I}^\perp = \mathfrak{E}^c + \mathfrak{T}$ with $\mathfrak{T}$ of (8).

3.1.3 Step 2

The stepsize s for the computation of $\mathfrak{E}^s := \mathfrak{E}^c + s(\mathfrak{I}^\perp - \mathfrak{E}^c) = \mathfrak{E}^c + s\mathfrak{T}$ is chosen as follows: In the general case we take a small stepsize s, whose validity is tested with the Armijo rule. Only if g^c equals g^- we apply the step size selection of Pottmann and Hofer (2004) (see also Hofer and Pottmann (2004)), which is based on an approximate planar development of a part of the footpoint cone. This can be done for the following reason:

If $g^c = g^- = 0$ we have exactly the same situation as in the cited papers and the part of the footpoint cone Λ_{M^6} is determined by the vertex $\mathfrak{I}$, the footpoints $\mathfrak{E}^c \in M^6$ and $\mathfrak{E}^- \in M^6$ and the direction $\mathfrak{T}$ of the base curve's tangent at the point $\mathfrak{E}^c$. If $g^c = g^- \neq 0$ we can think of a $(6-g)$-dimensional manifold $B^{6-g} \in M^6$ which locally approximates the boundary of $\mathcal{C}$ such that $T^{6-g}_{\mathfrak{E}^c}$ and $T^{6-g}_{\mathfrak{E}^-}$ are the tangent spaces in $\mathfrak{E}^c \in B^{6-g}$ resp. $\mathfrak{E}^- \in B^{6-g}$ with respect to B^{6-g}. Now the stepsize selection can be done by developing the footpoint cone $\Lambda_{B^{6-g}}$ into the plane. The validity of the resulting stepsize must be tested with the Armijo rule, because large changes of the normal curvature of the surface M^6 resp. B^{6-g} unbalance the supposed stepsize selection.

In the next step we project $\mathfrak{E}^s := \mathfrak{E}^c + s\mathfrak{T}$ orthogonally back onto M^6, which yields $\mathfrak{E}^\perp$. In the algorithm this is done by using the helical motion σ which is determined by the instantaneous screw $s\underline{\mathbf{q}} = (s\mathbf{q}, s\widehat{\mathbf{q}})$. More precisely, for $\mathbf{q} \neq 0$ we get $\mathsf{X}^{\varepsilon^\perp}$ by applying a motion to $\mathsf{X}^{\varepsilon^c}$ which is the superposition of a rotation about the axis $(\mathbf{p}, \widehat{\mathbf{p}})$ through an angle of

$$\lambda = \arctan(s\|\mathbf{q}\|) \in \left]0, \frac{\pi}{2}\right[\qquad \text{(or for small } \lambda \text{ just } \lambda = s\|\mathbf{q}\|) \tag{23}$$

and a translation $\lambda\, p\, \mathbf{p}$ with $\mathbf{p} := (p^1, p^2, p^3)^T$ parallel to the axis. Therefore this helical motion σ can be parametrized as

$$\sigma_t(\mathbf{x}^{\varepsilon^c}) = \mathbf{R}(\mathbf{x}^{\varepsilon^c} - \mathbf{p}\times \widehat{\mathbf{p}}) + t\, p\, \mathbf{p} + \mathbf{p}\times \widehat{\mathbf{p}} \tag{24}$$

with

$$\mathbf{R} = \begin{pmatrix} r_0^2 + r_1^2 - r_2^2 - r_3^2 & 2(r_1 r_2 + r_0 r_3) & 2(r_1 r_3 - r_0 r_2) \\ 2(r_1 r_2 - r_0 r_3) & r_0^2 - r_1^2 + r_2^2 - r_3^2 & 2(r_2 r_3 + r_0 r_1) \\ 2(r_1 r_3 + r_0 r_2) & 2(r_2 r_3 - r_0 r_1) & r_0^2 - r_1^2 - r_2^2 + r_3^2 \end{pmatrix}$$

and

$$r_0 := \cos\left(\frac{t}{2}\right), \qquad r_i := p^i \sin\left(\frac{t}{2}\right) \quad \text{for } i = 1, 2, 3. \tag{25}$$

If we set t equal to λ the point $\mathbf{x}^{\varepsilon^c}$ is mapped onto $\sigma_\lambda(\mathbf{x}^{\varepsilon^c}) = \mathbf{x}^{\varepsilon^\perp}$. For the computation of the Plücker coordinates $(\mathbf{p}, \widehat{\mathbf{p}})$ of the axis and the pitch p from $\underline{\mathbf{q}}$ see (9).

In the special case of $\mathbf{q} = 0$ there is no need of an orthogonal projection onto M^6 because $\mathfrak{E}^s$ is already located on M^6 for any stepsize s ($\Longrightarrow$ $\mathfrak{E}^s = \mathfrak{E}^\perp$). This is due to the fact that $\underline{\mathbf{q}}$ is an instantaneous translation and therefore we get $\mathsf{X}^{\varepsilon^\perp}$ by translating $\mathsf{X}^{\varepsilon^c}$ along the vector $s\widehat{\mathbf{q}}$.

Remark: We project $\mathfrak{E}^s$ onto M^6 by using the helical motion σ associated with $\underline{\mathbf{q}}$ instead of solving the registration problem with known correspondences (see e.g. Belta and Kumar (2002) and Horn (1987)) as recommended by Botsch et al. (2006). Therefore we can give a more efficient algorithm for the reduction of the stepsize, if $\mathfrak{E}^\perp$ corresponds to a colliding configuration. This algorithm is outlined in the next section.

3.1.4 Step 3

For the collision test we also use the signed distance field to the boundary surface of the fixed object Y. The two objects are interpenetrating each other if $min\left(\left\{d(\mathbf{x}^{\varepsilon^\perp}, \mathsf{Y}) \,|\, \mathbf{x} \in \mathcal{X}\right\}\right) < 0$. If there is no intersection then we set $\mathfrak{E}^\perp$ equal to $\mathfrak{E}^+$, otherwise we apply the following algorithm for the reduction of the stepsize such that $\mathfrak{E}^\perp$ is located on the boundary of $\mathcal{C}$.

Stepsize reduction algorithm for $\mathbf{q} \neq 0$

Because the orthogonal projection is done via the uniform helical motion of (24) associated with $\underline{\mathbf{q}}$, a reduction of the stepsize s is equal to the reduction

λ of (23). In the following we give an algorithm that would describe how a parameter value $\lambda^{\partial\mathcal{C}}$ of the helical motion can be computed, such that $\sigma_{\lambda^{\partial c}}(\mathsf{X}^{\varepsilon^c})$ and Y are in contact with respect to the contact value w^c. The given iterative procedure consists of repeated application of the following steps, where the current iterate is denoted by λ^c and the next one by λ^+ (see *Fig. 3*).

Due to the applied collision test the points $\mathbf{d}_i^{\varepsilon^\perp} \in \mathcal{X}^{\varepsilon^\perp}$ with

$$d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y}) = min\left(\left\{d(\mathbf{x}^{\varepsilon^\perp}, \mathsf{Y}) \,|\, \mathbf{x} \in \mathcal{X}\right\}\right). \tag{26}$$

can be computed in linear time. These points are global minima of the distance function $d(\mathbf{z}, \mathsf{Y})$ on the mesh $\mathcal{X}^{\varepsilon^\perp}$. Then we compute the distance k_i of $\mathbf{d}_i^{\varepsilon^\perp}$ to the axis $(\mathbf{p}, \widehat{\mathbf{p}})$ of the helical motion σ according to

$$k_i = \|\widehat{\mathbf{p}} + (\mathbf{d}_i^{\varepsilon^\perp} \cdot \mathbf{p})\mathbf{p} - \mathbf{d}_i^{\varepsilon^\perp}\|. \tag{27}$$

Without loss of generality we assume $k_1 \leq k_2 \ldots \leq k_k$. The basic idea of this algorithm is the computation of the parameter λ_i^+ such that the chord length between the local minimum $\mathbf{d}_i^{\varepsilon^\perp}$ and the new iterate of this point $\sigma_{\lambda_i^+}(\mathbf{d}_i^{\varepsilon^c})$ equals the absolute value of the distance $d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y})$ plus a constant u_i, i.e.

$$\|\sigma_{\lambda_i^+}(\mathbf{d}_i^{\varepsilon^c}) - \mathbf{d}_i^{\varepsilon^\perp}\| = u_i - d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y}). \tag{28}$$

In order to guarantee that the solution of this iteration $\sigma_{\lambda^{\partial c}}(\mathsf{X}^{\varepsilon^c})$ and Y are in contact with respect to the contact value w^c, we choose u_i as follows

$$\begin{array}{ll} \bullet & u_i = w^c \quad \text{for } d(\mathbf{d}_i^{\varepsilon^c}, \mathsf{Y}) > w^c \text{ and} \\ \bullet & u_i = a \cdot d(\mathbf{d}_i^{\varepsilon^c}, \mathsf{Y}) \quad \text{with } a \in \,]0,1[\, \text{ for } d(\mathbf{d}_i^{\varepsilon^c}, \mathsf{Y}) \leq w^c. \end{array} \tag{29}$$

If we set $\lambda_i^+ = \lambda^c - \mu_i$ we obtain μ_i by solving the following equation

$$p^2\mu_i^2 + 2k_i^2(1 - \cos(\mu_i)) - (u_i - d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y}))^2 = 0. \tag{30}$$

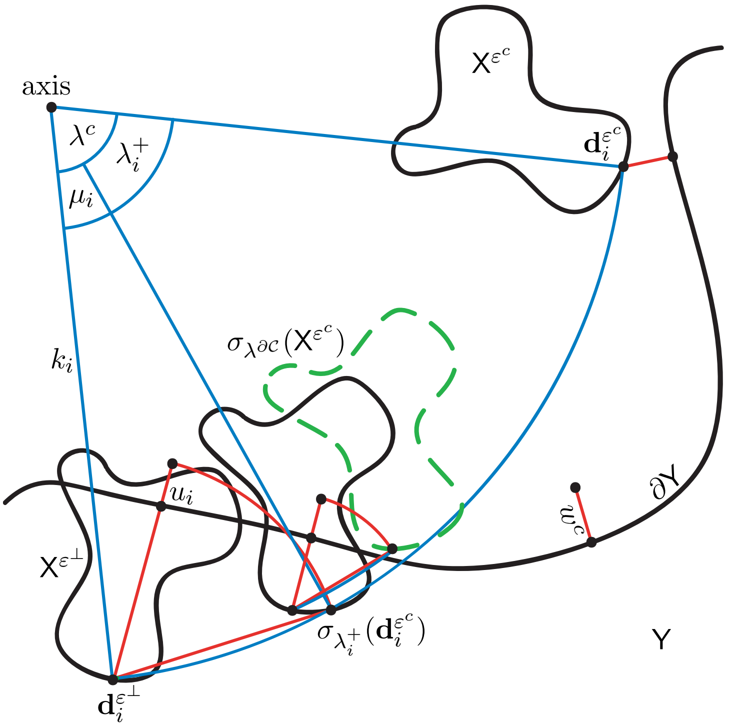
Because of the validity of the trivial inequality

$$u_i - d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y}) \leq d(\mathbf{d}_i^{\varepsilon^c}, \mathsf{Y}) - d(\mathbf{d}_i^{\varepsilon^\perp}, \mathsf{Y}) \leq \|\mathbf{d}_i^{\varepsilon^c} - \mathbf{d}_i^{\varepsilon^\perp}\| \tag{31}$$

and $\lambda^c \in \,]0, \pi/2[$ due to (23) the last equation has a unique solution for μ_i within the interval $]0, \lambda^c]$. If we assume $u_i = c$ for all $i = 1, \ldots, k$ then from (30) follows immediately $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_k$. In order to reduce computational costs we suppose to solve (30) only for $i = 1$ if there is more than one global minimum. It should be noted that in the border case $k_1 = 0$ equation (30) can be solved explicitly for μ_1 which yields

$$\mu_1 = \left|\frac{u_1 - d(\mathbf{d}_1^{\varepsilon^\perp}, \mathsf{Y})}{p}\right|. \tag{32}$$

Now we set $\lambda^+ = \lambda^c - \mu_1$ and run again the collision test for $\sigma_{\lambda^+}(\mathsf{X}^{\varepsilon^c})$ and Y. If there is no interpenetration, then the meshes must be in contact due to the construction of the stepsize reduction algorithm. Otherwise we iterate the procedure.



axis
$\mathsf{X}^{\varepsilon^c}$
λ^c
λ_i^+
μ_i
$\mathbf{d}_i^{\varepsilon^c}$
k_i
$\sigma_{\lambda^{\partial C}}(\mathsf{X}^{\varepsilon^c})$
u_i
$\mathsf{X}^{\varepsilon^\perp}$
w^c
$\partial\mathsf{Y}$
$\sigma_{\lambda_i^+}(\mathbf{d}_i^{\varepsilon^c})$
Y
$\mathbf{d}_i^{\varepsilon^\perp}$

Fig. 3. Stepsize reduction algorithm for $\mathbf{q} \neq 0$. This algorithm describes how a parameter value $\lambda^{\partial C}$ of the helical motion σ can be computed, such that $\sigma_{\lambda^{\partial C}}(\mathsf{X}^{\varepsilon^c})$ and Y are in contact with respect to the contact value w^c.

Remark: Instead of using the stepsize reduction algorithm one can also apply methods for continuous collision detection under screwing motions like the one proposed by Kim and Rossignac (2003). It should be noted that such methods are time consuming tasks, because the exact collision time (stepsize) is computed by testing all vertex/face and edge/edge collision cases which cannot be excluded by a series of rejection tests.

In contrast to the cited method the proposed one is based on an iterative algorithm using the signed distance field of the fixed object. For the computation of the generalized penetration depth it is advantageous to use such a distance field because it only must be computed once. Moreover we are not interested in the exact collision time but only in a stepsize such that the surfaces are in contact with respect to the contact value w.

Stepsize reduction algorithm for $\mathbf{q} = 0$

Due to the equation $\mathsf{X}^{\varepsilon^s} = \mathsf{X}^{\varepsilon^c} + \tau\widehat{\mathbf{q}}_0$ where $\widehat{\mathbf{q}}_0$ denotes the normalized translation vector and $\tau := s\|\widehat{\mathbf{q}}\|$, a reduction of the stepsize s corresponds to a reduction of τ. We give again an iterative procedure, where the current iterate is denoted by τ^c and the next one by τ^+. As in the case $\mathbf{q} \neq 0$ we compute the global minima $\mathbf{d}_i^{\varepsilon^\perp}, \ldots, \mathbf{d}_k^{\varepsilon^\perp}$ according to (26). Without loss of generality we assume $d(\mathbf{d}_1^{\varepsilon^c}, \mathsf{Y}) \geq d(\mathbf{d}_2^{\varepsilon^c}, \mathsf{Y}) \geq \ldots \geq d(\mathbf{d}_k^{\varepsilon^c}, \mathsf{Y})$. Now we reduce τ^c by $\nu_i \in \mathbb{R}^+$ given by

$$\nu_i := u_i - d(\mathbf{d}_k^{\varepsilon^\perp}, \mathsf{Y}) \tag{33}$$

with u_i of (29). Because of $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_k$ we set $\tau^+ := \tau^c - \nu_1$ and run again the collision test. If there is no intersection, then the meshes must be in contact due to the applied stepsize reduction algorithm. Otherwise we iterate the procedure.

If $\|\mathfrak{C}^+ - \mathfrak{C}^c\| < b$ and if the current contact value w^c also falls below a certain threshold v, then the $PD_S(\mathsf{X}, \mathsf{Y})$ algorithm is stopped.

Remark: The presented algorithm can easily be adapted for the computation of the translational penetration depth. We only have to solve the system of linear equations given in (21) under the side condition that $\mathbf{q} = \mathbf{o}$. Furthermore we have to consider that there exists no translational gliding motion if g^c is greater than 2 which affects the conditions in case A and case B of step 1.

3.2 Computational complexity

As preprocessing step we have to compute the distance field of Y. This can for example be done with the fast sweeping algorithm of Zhao (2005) which is of complexity $O(N)$, where N denotes the number of grid points.

The collision test is based on this distance field. There are different possibilities to compute $d(\mathbf{x}, \mathsf{Y})$ from the distances of the neighboring grid points of $\mathbf{x}$, but it can be done with a computational complexity depending linearly on the number $N_{\mathbf{x}}$ of vertices of $\mathcal{X}$. If these distances are known, the computation of the minimal value $d_{min} = min(d(\mathbf{x}, \mathsf{Y}) | \mathbf{x} \in \mathcal{X})$, which already indicates the collision, is of complexity $O(N_{\mathbf{x}})$. Moreover the point which causes this minimal value is also used for the stepsize reduction algorithm and so we need no further computation for this part.

It can also be seen immediately from d_{min} whether the non-penetrating objects are in contact with respect to the contact value w. If $0 \leq d_{min} \leq w$ holds we must check if there are also other contact points. Now the time complexity for computing the local minima depends linearly on the number of points $\mathbf{x} \in \mathcal{X}$ with $d(\mathbf{x}, \mathsf{Y}) \leq w$. Computing this set of points is again of complexity $O(N_{\mathbf{x}})$.

3.3 Physical interpretation of PD_S

In the following we present a physical interpretation of $PD_S(\mathsf{X},\mathsf{Y})$. Assume the pose X^{φ} which cause $PD_S(\mathsf{X},\mathsf{Y})$ is known. This pose corresponds to the local minimizer $\mathfrak{F} \in \partial\mathcal{C}$ with $\mathfrak{F} = (\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$. It is well known that the eigenvector of $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$ with respect to the eigenvalue 1 equals the direction of the rotation axis $\mathbf{a}$ (unit vector) and that the rotation angle ρ is given by $2\cos\rho = trace(\mathbf{F}) - 1$. Under this considerations $PD_S(\mathsf{X},\mathsf{Y})$ can be rewritten as follows:

$$PD_S(\mathsf{X},\mathsf{Y})^2 = \frac{\|\mathfrak{E} - \mathfrak{I}\|^2}{6} = \|\mathbf{f}_0\|^2 + \frac{\mathbf{d}^T \cdot \mathbf{T}_{\mathsf{X}} \cdot \mathbf{d}}{6} \quad \text{with} \tag{34}$$

$$\mathbf{d}^T = \left(\pm\sqrt{2f_{11} - 2\cos\rho}, \pm\sqrt{2f_{22} - 2\cos\rho}, \pm\sqrt{2f_{33} - 2\cos\rho}\right) \tag{35}$$

where $\mathbf{T}_{\mathsf{X}}$ denotes the inertia tensor of the moving body X. Two of the eight possibilities of $\mathbf{d}$ are linearly dependent with the direction $\mathbf{a}$ of the rotation axis. [1] Due to the well known formula of the total kinetic energy K

$$K = \frac{mv^2}{2} + \frac{\mathbf{T}\omega^2}{2}, \tag{36}$$

where m denotes the mass of the body, v the body's velocity, $\mathbf{T}$ the body's moment of inertia and ω the body's angular velocity, we can interpret $PD_S(\mathsf{X},\mathsf{Y})$ as follows:

Theorem 1 $PD_S(\mathsf{X},\mathsf{Y})$ *equals* $\sqrt{K}$ *where* K *is the total kinetic energy of* S *induced by the instantaneous screw* $\underline{\mathbf{q}} = (\omega\mathbf{a}, \mathbf{f}_0)$ *with*

$$\omega = \sqrt{2(1 - \cos\rho)} = \sqrt{3 - trace(\mathbf{F})}. \tag{37}$$

S *denotes an ellipsoidal shell of mass 2, whose vertices are identical with the six special points* $\mathbf{s}_i$ *(*$i = 1, \ldots, 6$*) of the moving body* X *given in* (4).

The angular velocity ω of (37) is the chord length approximation of the arc length of ρ and not ρ itself. This is due to the used metric S of the ambient space. Therefore $PD_S(\mathsf{X},\mathsf{Y})$ equals the square root of the minimal needed kinetic energy of S such that S^{id} moves affinely into the pose S^{φ} within one time unit. It should be noted that this ellipsoidal shell S was also used to define a new performance index for 6R robots (see Nawratil (2007)).

[1] Due to the symmetry of the inertia ellipsoid there exist eight possibilities for choosing $\mathbf{d}$.

In the following we present two methods for calculating an initial guess.

3.4.1 Method 1

We replace the moving body X by the smallest ellipsoidal shell E that encloses X. E should be centered at the barycenter $\mathbf{b}_{\mathsf{X}}$ and its axis lengths ratio should equal $a_1 : a_2 : a_3 = f_1 : f_2 : f_3$. Because E encloses X, E and Y are interpenetrating each other. Now we want to compute the closest collision-free pose E^{ς} to E according to the distance metric (2) where ς is a superposition of a rotation $\rho_i \in I$ and a translation $\mathbf{t} \in \mathbb{R}^3$. I denotes the icosahedral group of rotations which contains 60 elements. This group serve us as a fair discretization of $SO(3)$.

The map ς can be computed as follows: If E^{ρ_i} is a collision-free configuration we are done. Otherwise we have to compute the shortest translation vector $\mathbf{t}_i$ which separates E^{ρ_i} and Y. This can easily done as follows: We apply to E^{ρ_i} and Y the affine mapping α_i which maps E^{ρ_i} onto a sphere of radius a_1. Assume $\mathbf{A}_i$ denotes the matrix of this linear mapping α_i. Then the offset mesh $\mathcal{OY}^{\alpha_i}$ of distance a_1 from $\mathcal{Y}^{\alpha_i}$ is given by:

$$\mathcal{OY}^{\alpha_i} := \left\{ \mathbf{z}_i := \mathbf{A}_i \cdot \mathbf{y}_i + \frac{\mathbf{A}_i^{-T} \cdot \mathbf{n}_{\mathbf{y}_i}}{\|\mathbf{A}_i^{-T} \cdot \mathbf{n}_{\mathbf{y}_i}\|} a_1 \,\middle|\, \mathbf{y}_i \in \mathcal{Y} \right\}. \tag{38}$$

If Y is not convex we have to trim $\mathcal{OY}^{\alpha_i}$ which yields the trimmed offset mesh $\mathcal{TY}^{\alpha_i}$ with

$$\mathcal{TY}^{\alpha_i} := \{ \mathbf{z}_i \,|\, \mathbf{z}_i \in \mathcal{OY}^{\alpha_i} \ \wedge \ d(\mathbf{z}_i, \mathsf{Y}^{\alpha_i}) \geq a_1 \}, \tag{39}$$

where the distance function $d(\mathbf{z}_i, \mathsf{Y}^{\alpha_i})$ can, for example, be computed by the Approximate Nearest Neighbors (ANN) algorithm. Now we apply to $\mathcal{TY}^{\alpha_i}$ the mapping α_i^{-1} which yields the point cloud $\mathcal{TY}$. Trivially the position vector of that point of $\mathcal{TY}$ which is closest to the origin is the searched translation vector $\mathbf{t}_i$. Now ς equals those of the 60 mappings ς_i which causes the minimal distance $S(\mathsf{X}^{id}, \mathsf{X}^{\varsigma_i})$. Because E encloses X we can be sure that the starting configuration X^{ς} is collision-free.

Remark: One must not stick to the icosahedral group but could also take a refined discretization of $SO(3)$. Subdivision schemes for a fair discretization of the spherical motion group are given in Nawratil and Pottmann (in press).

The limitations of this conservative method are illustrated in Fig. 6 of section 5. Moreover it should be noted that this method works in all cases in contrast to the second method, presented next.

3.4.2 Method 2

The key idea of this more sophisticated method for generating an initial guess is that we compute corresponding point pairs $\{\mathbf{x}_i, \mathbf{c}_i\}$ where $\mathbf{c}_i$ is the closest point (footpoint) of $\mathcal{Y}$ with respect to $\mathbf{x}_i \in \mathcal{X}$ (see Fig. 4). Based on the resulting known correspondence it is possible to formulate the following algorithm which tries to minimize the extent of overlap in each iteration step:

1. We choose a chord length $cl > 0$.

2. Compute the screw $\underline{\mathbf{q}}_{max}$ which maximizes the objective function

$$\zeta(\underline{\mathbf{q}}): \quad \sum_{i=1}^{N} w(\mathbf{x}_i)\mathbf{v}(\mathbf{x}_i)\cdot\mathbf{n}_{\mathbf{c}_i} \quad \text{with} \quad d(\mathbf{x}_i, \mathsf{Y}) \le cl \tag{40}$$

   and

$$w(\mathbf{x}_i) = \frac{d(\mathbf{x}_i, \mathsf{Y}) + cl}{min(d(\mathbf{x}, \mathsf{Y})|\mathbf{x} \in \mathcal{X}) - cl} \in [0, 1] \tag{41}$$

   under the side condition

$$\nu(\underline{\mathbf{q}}): \quad \sum \|v(\mathbf{s}_i)\|^2 = 1. \tag{42}$$

   Hence, we maximize nothing else than the weighted sum of the components of the velocity vectors $\mathbf{v}(\mathbf{x}_i)$ in direction of the surface normal vector $\mathbf{n}_{\mathbf{c}_i}$ in $\mathbf{c}_i$ of $\mathcal{Y}$. Moreover it should be noted that the weighting function $w(\mathbf{x}_i)$ preferring points with a deeper penetration, depends linearly on the distance $d(\mathbf{x}_i, \mathsf{Y})$. The side condition normalizes the screw according to the metric of the ambient space. As $\nu(\underline{\mathbf{q}})$ is a quadratic form and $\zeta(\underline{\mathbf{q}})$ a linear equation in the unknowns of the screw, we get $\underline{\mathbf{q}}_{max}$ by solving a system of linear equations.

   Remark: In the unlucky case that $\mathbf{x}_i$ is located on the cut locus of the fixed object, we get several footpoints $\mathbf{c}_i^j$ with $j > 1$. For those points we replace $\mathbf{v}(\mathbf{x}_i)\cdot\mathbf{n}_{\mathbf{c}_i}$ of the objective function by the mean value $\sum_j \mathbf{v}(\mathbf{x}_i)\cdot\mathbf{n}_{\mathbf{c}_i^j}/j$.

3. Compute the helical motion σ associated with $\underline{\mathbf{q}}_{max}$ and displace the object X such that $max(\|\sigma(\mathbf{x}_i) - \mathbf{x}_i\|) = cl$. Note that $max(\|\sigma(\mathbf{x}_i) - \mathbf{x}_i\|) = \|\sigma(\mathbf{x}_{max}) - \mathbf{x}_{max}\|$ holds where $\mathbf{x}_{max}$ is the point of $\mathcal{X}$ which has the biggest distance k_{max} from the helical axis or axis of rotation, respectively. Then the corresponding stepsize can be computed as in subsection 3.1.4 for the point $\mathbf{x}_{max}$. If $\underline{\mathbf{q}}_{max}$ is a translation the stepsize s can be computed from $cl = s\|\widehat{\mathbf{q}}_{max}\|$. We end up with the new iterate X^+.

It should be noted that the above given stepsize selection guarantees that points $\mathbf{x}_i \in \mathcal{X}$ which have no influence on the objective function due to $d(\mathbf{x}_i, \mathsf{Y}) > cl$ do not penetrate the fixed object in the next iterate. The reason for the restriction of the objective function to this set of points is the following:

In most practical cases, the extent of penetration is small, and therefore the set of non-penetrating points far away from the fixed object would affect the objective function too much despite of the natural weighting function $w(\mathbf{x}_i)$. Therefore the input data reduction as well as the weighting function $w(\mathbf{x}_i)$ improves the quality of the screw $\underline{\mathbf{q}}_{max}$ which can be evaluated by the function

$$\kappa(\underline{\mathbf{q}}) : \quad \zeta(\underline{\mathbf{q}}) / \sum_{i=1}^{N} w(\mathbf{x}_i) \|\mathbf{v}(\mathbf{x}_i)\| \in [0,1] \quad \text{with} \quad d(\mathbf{x}_i, \mathsf{Y}) \leq cl. \qquad (43)$$

A value of 1 indicates a velocity field $\underline{\mathbf{q}}$ which exactly fits our requirements.

We suggest to set the only free parameter cl equal to $w - min(d(\mathbf{x}, \mathsf{Y}) | \mathbf{x} \in \mathcal{X})$ where w is the contact value of Def 2 (see Fig. 4). Additionally we set $cl = b$ if cl exceeds a predefined upper bound b. This choice ensures that the obtained initial guess and the fixed object are in contact with respect to the contract value w, if the algorithm yields a solution. In section 5 we give examples which show that this algorithm works very well for most practical cases where the extent of penetration is small.

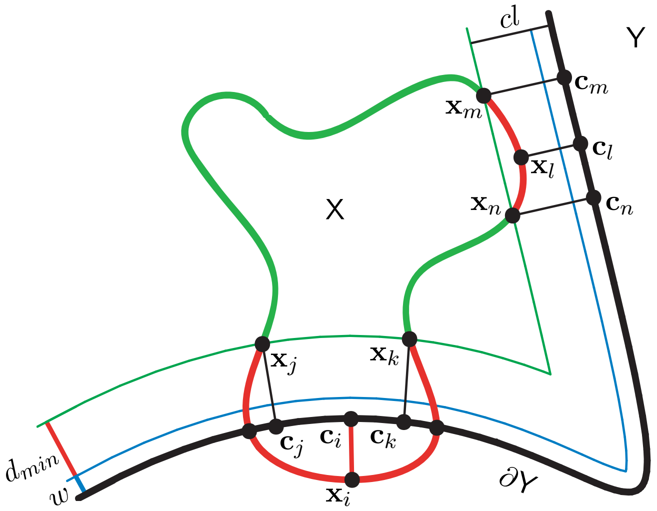


Fig. 4. Method 2 for the computation of a collision-free starting configuration: We compute for each point $\mathbf{x}_i \in \mathcal{X}$ with $d(\mathbf{x}_i, \mathsf{Y}) \leq cl$ the corresponding closest point (footpoint) $\mathbf{c}_i \in \mathcal{Y}$. Moreover the selection of the chord length cl as $w - min(d(\mathbf{x}, \mathsf{Y}) | \mathbf{x} \in \mathcal{X})$ is illustrated.

4 PD computation with respect to the geodesic metric of M^6

In this section we want to compute the footpoint $\mathfrak{G}$ on $\partial\mathcal{C}$ with respect to the geodesic metric of M^6 which is defined as follows:

Definition 5 *The geodesic distance $G(\mathsf{X}^\alpha, \mathsf{X}^\beta) = \|\mathfrak{A} - \mathfrak{B}\|_G$ between two points $\mathfrak{A}$ and $\mathfrak{B}$ on M^6 is defined as the length of the shortest (geodesic) path $\mathfrak{g} \in M^6$ with respect to the metric from the ambient space* (5) *connecting $\mathfrak{A}$ and $\mathfrak{B}$.*

The following results about geodesics on M^6 are known: Pottmann et al. (2004) proved that motions $\mathbf{c}_t$ which join two given positions X^α resp. X^β and arise from minimization of the functional

$$E_1 = \int_0^1 \|\dot{\mathbf{c}}_t\|^2 dt \quad \text{with} \quad \mathbf{c}_0 = \mathsf{X}^\alpha \quad \text{and} \quad \mathbf{c}_1 = \mathsf{X}^\beta \tag{44}$$

correspond to geodesics $\mathfrak{g}$ on M^6, parametrized by a constant multiple of arc length. The meaning of minimizing E_1 is that the total first energy of the feature point trajectories is minimized. Moreover they proved, that the trajectory of the barycenter under the geodesic motion is a straight line traced with constant speed. These motions are well-known in mechanics as free motions of a body (see Arnol'd (1989)).

We can define the generalized penetration depth with respect to the geodesic metric of M^6 analogously to (14) as:

$$PD_G(\mathsf{X}, \mathsf{Y}) = min\left(\left\{\frac{\|\mathfrak{I} - \mathfrak{E}\|_G}{\sqrt{6}} \,\middle|\, \mathfrak{E} \in \partial\mathcal{C}\right\}\right). \tag{45}$$

It should be noted that $PD_G(\mathsf{X}, \mathsf{Y})$ is not necessarily equal to PD^t if both given objects X and Y are convex, because the counter example also holds for the geodesic metric. This can easily be verified by taking into account that the geodesic motion between X and X^ρ is the rotation about the x-axis (see Arnol'd (1989)). It follows immediately that $PD_G(\mathsf{Y}, \mathsf{X})$ is greater than $PD_G(\mathsf{X}, \mathsf{Y})$ in the given example, which shows that $PD_G(\mathsf{Y}, \mathsf{X})$ and $PD_G(\mathsf{X}, \mathsf{Y})$ are not equal in the general case. Trivially $PD_G(\mathsf{X}, \mathsf{Y}) \geq PD_S(\mathsf{X}, \mathsf{Y})$ where the equality only holds if the Euclidean map φ which causes $PD_S(\mathsf{X}, \mathsf{Y})$ is a pure translation. In the following we want to give an algorithm for the computation of the closest point $\mathfrak{G}$ on $\partial\mathcal{C}$ with respect to the geodesic metric of Def. 5.

It is well known that the computation of geodesics requires discretization. The unknown curve $\mathfrak{g} \in M^6$ must pass through the only fixed point $\mathfrak{I} \in M^6$ and a point $\mathfrak{E}_N$ on the boundary of $\mathcal{C}$ due to (45). The curve $\mathfrak{g}$ itself is represented by a point sequence $\mathfrak{I}, \mathfrak{E}_1, \mathfrak{E}_2, \ldots, \mathfrak{E}_N$ which contains the points $\mathfrak{E}_N$ and $\mathfrak{I}$ (see *Fig. 5*).

We can view the point sequence $\mathfrak{E}_1, \ldots, \mathfrak{E}_N$ on M^6 as a point $\mathsf{P} \in \Phi$ in $\mathbb{R}^{12N}$, where Φ denotes the set of all points, such that each single $\mathfrak{E}_i$'s are contained in M^6. Therefore the dimension of Φ is equal to $6N$. Due to the fact that the tangent space $T^6_{\mathfrak{E}_i}$ of M^6 for each point $\mathfrak{E}_i$ of P is spanned by the six vectors $\mathfrak{T}_i^1, \ldots, \mathfrak{T}_i^6$ of (10), the tangent space in P of Φ is given by the $6N$ vectors

$$\mathsf{B}_i^j := (0, \ldots, 0, \mathfrak{T}_i^j, 0, \ldots, 0) \quad \text{with} \quad i = 1, \ldots, N \quad \text{and} \quad j = 1, \ldots, 6. \tag{46}$$

In order to simplify the computation we apply to M^6 a translation δ such that $\delta(\mathfrak{I})$ equals the origin $\mathfrak{O} = (0, \ldots, 0)$. If we denote the translated points $\delta(\mathfrak{E}_i)$ by $\mathfrak{K}_i \in \delta(M^6)$, the point sequence $\mathfrak{K}_1, \ldots, \mathfrak{K}_N$ can be seen as a point $\mathsf{K} := \Delta(\mathsf{P}) \in \Delta(\Phi)$ where Δ is a translation along the vector $-\mathsf{I} := -(\mathfrak{I}, \ldots, \mathfrak{I})$. It should be noted that due to the applied translation δ the tangent spaces of $\mathfrak{E}_i$ and $\mathfrak{K}_i$ on M^6 resp. $\delta(M^6)$ are spanned by the same basis vectors and therefore they are parallel.

If we use the difference of successive points as a discrete first derivative and replace integration by summation, the functional E_1 converts into

$$E_1 = \|\mathfrak{K}_1\|^2 + \sum_{i=1}^{N-1} \|\mathfrak{K}_i - \mathfrak{K}_{i+1}\|^2. \tag{47}$$

Because E_1 of (47) is a quadratic function it can be written as

$$E_1 : \ \mathbb{R}^{12N} \mapsto \mathbb{R}, \qquad E_1(\mathsf{K}) = \mathsf{K}\cdot\mathbf{Q}\cdot\mathsf{K} + 2\mathsf{l}^T\cdot\mathsf{K} + c \tag{48}$$

where $\mathbf{Q}$ is a symmetric positive definite matrix. Now we want to minimize E_1 under the constraint that K lies in the surface $\Delta(\Phi) \in \mathbb{R}^{12N}$. The geometric approach to this minimization problem views the matrix $\mathbf{Q}$ as the matrix of the inner product $\langle \mathsf{K}, \mathsf{K} \rangle := \mathsf{K}^T\cdot\mathbf{Q}\cdot\mathsf{K}$, of a Euclidean metric in $\mathbb{R}^{12N}$. Generally E_1 assumes its minimum in the point $-\mathbf{Q}^{-1}\cdot\mathbf{l}$. In our case this point is identical with the origin $\mathsf{O} = (\mathfrak{O}, \ldots, \mathfrak{O})$ because $E_1(\mathsf{O}) = 0$. As a consequence $\mathbf{l} = \mathbf{o}$

and $c = 0$, thus E_1 is equal to the length of the vector K; i.e.

$$E_1(\mathsf{K}) = \mathsf{K}^T \cdot \mathbf{Q} \cdot \mathsf{K} = \|\mathsf{K}\|^2_{\mathbf{Q}} \quad \text{with} \quad \mathbf{Q} = \begin{pmatrix} 2\mathbf{M} & -\mathbf{M} & & & \\ -\mathbf{M} & 2\mathbf{M} & -\mathbf{M} & & \\ & \ddots & \ddots & \ddots & \\ & & -\mathbf{M} & 2\mathbf{M} & -\mathbf{M} \\ & & & -\mathbf{M} & \mathbf{M} \end{pmatrix} \tag{49}$$

and $\mathbf{M}$ of (7). O is the solution of our minimization problem because this point belongs to $\Delta(\Phi)$. But we have an additional constraint namely that the point $\mathfrak{K}_N$ is located on the translated boundary $\delta(\partial\mathcal{C})$. Therefore we have to compute the closest point $\Delta(\mathsf{F})$ of $\Delta(\Gamma)$ to O with respect to the metric defined by $\mathbf{Q}$, where Γ denotes the set of all points $\mathsf{P} = (\mathfrak{C}_1, \ldots, \mathfrak{C}_N) \in \Phi$ with $\mathfrak{C}_N \in \partial\mathcal{C}$. Consequently, the point $\mathsf{F} \in \Gamma$ is the solution of our optimization problem, which can be computed with the following projected gradient algorithm.

4.2 Algorithm for the computation of PD with respect to the geodesic metric of M^6

The given $PD_G(\mathsf{X}, \mathsf{Y})$ algorithm is again based on the work done by Hofer and Pottmann (2004). When we discuss one procedure of the iteration, we will denote the current iterate by $\mathsf{P}^c = (\mathfrak{C}^c_1, \ldots, \mathfrak{C}^c_N)$, the next iterate by $\mathsf{P}^+ = (\mathfrak{C}^+_1, \ldots, \mathfrak{C}^+_N)$ and the prior one by P^-. We assume that the initial guess is a discretized geodesic connecting $\mathfrak{I}$ and the footpoint $\mathfrak{F}$, computed with the $PD_S(\mathsf{X}, \mathsf{Y})$ algorithm of the last section. The computation of this initial guess can be done with the algorithm given in the above cited paper.

1. If $\mathfrak{C}^c_N \in \partial\mathcal{C}$ we compute the g^c linearly independent gliding constraints according to (11). Otherwise $g^c = 0$. Then we compute the tangent space $\mathsf{T}^{6N-g^c}_{\mathsf{P}^c}$ of Φ at the current iterate P^c and project the point O orthogonally into the translated tangent space $\mathsf{T}^{6N-g^c}_{\mathsf{K}^c}$ of $\Delta(\Phi)$ at the point $\Delta(\mathsf{P}^c) = \mathsf{K}^c$, which results in $\mathsf{O}^\perp$.

2. Compute an appropriate stepsize s and project $\mathfrak{C}^s_N$ of $\mathsf{P}^s := \mathsf{P}^c + s(\mathsf{O}^\perp - \mathsf{K}^c)$ onto M^6, which yields the point $\mathfrak{C}^\perp_N$.

3. If $\mathfrak{C}^\perp_N$ corresponds to a colliding configuration we reduce s until both surfaces are in contact. Then we project the points $\mathfrak{C}^s_1, \ldots, \mathfrak{C}^s_{N-1}$ onto M^6 which yields our new iterate P^+.

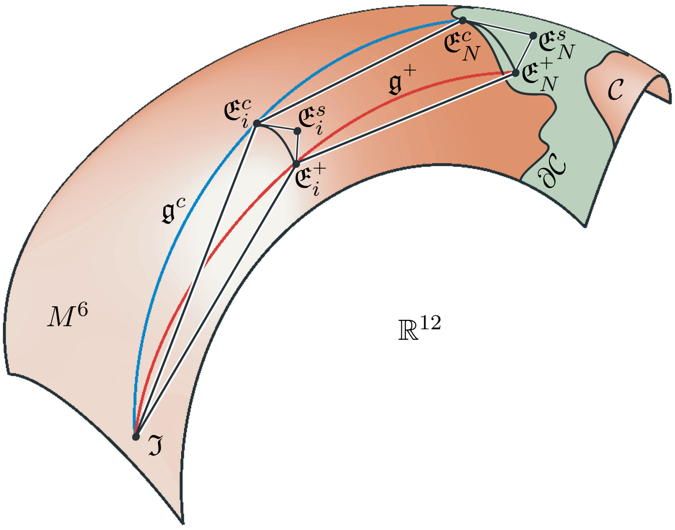


Fig. 5. $PD_G(\mathsf{X},\mathsf{Y})$ algorithm

4.2.1 Step 1

First of all we want to give an elementary view of this step: We displace each point $\mathfrak{E}_i^c$ of the polygon P^c in the tangent space $\mathfrak{T}_\mathsf{i}$ of M^6, such that the generated polygon $\mathfrak{E}_1^s = \mathfrak{E}_1^c + \mathfrak{T}_1, \ldots, \mathfrak{E}_N^s = \mathfrak{E}_N^c + \mathfrak{T}_N$ with $\mathfrak{T}_i$ of (8) minimizes the objective function E_1 under the g^c constraints that $\mathsf{X}^{\varepsilon_N^c}$ glides instantaneously on Y (see *Fig. 5*).

We compute the g^c gliding constrains according to (11) with respect to the contact value $w^c := w^-$. We redefine w^c in the same two cases as in the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm, where we replace $\|\mathfrak{E}^c - \mathfrak{E}^-\|$ by $\|\mathfrak{E}_N^c - \mathfrak{E}_N^-\|$. Further we compute the matrix $\mathbf{Q}$ of (48) and the basis vectors $\{\mathsf{B}_1^1, \ldots, \mathsf{B}_1^6, \ldots, \mathsf{B}_N^1, \ldots, \mathsf{B}_N^6\}$ of Φ's tangent space at the current iterate P^c, which are given by (46). Because these vectors also span the tangent space of $\Delta(\Phi)$ at the point K^c, we can compute the Gramian matrix $\mathbf{G}_{\mathsf{K}^c} := \left(\langle \mathsf{B}_j^i, \mathsf{B}_k^l \rangle_{\mathbf{Q}}\right)$. Then $\mathsf{O}^\perp$ equals $\mathsf{K}^c + \mathsf{T}$ with

$$\mathsf{T} := \sum_{i=1}^{N} \sum_{j=1}^{3} q_i^j \mathsf{B}_i^j + \widehat{q}_i^j \mathsf{B}_i^{j+3} \tag{50}$$

where $\mathsf{Q} := (\mathbf{q}_1, \widehat{\mathbf{q}}_1, \ldots, \mathbf{q}_N, \widehat{\mathbf{q}}_N)^T$ is the solution of the following systems of linear equations:

$$\begin{pmatrix} \mathbf{G}_{K^c} \\ \mathbf{N} \end{pmatrix} \cdot \mathsf{Q} = \begin{pmatrix} \mathbf{r} \\ \mathbf{o} \end{pmatrix} \quad \text{with} \quad \mathbf{N} := \begin{pmatrix} 0 \ldots 0 & \widehat{\mathbf{n}}_1^T & \mathbf{n}_1^T \\ 0 \ldots 0 & \vdots & \vdots \\ 0 \ldots 0 & \widehat{\mathbf{n}}_{g^c}^T & \mathbf{n}_{g^c}^T \end{pmatrix} \tag{51}$$

where $\mathbf{r} := (r_1^1, \ldots, r_1^6, \ldots, r_N^1, \ldots, r_N^6)^T$ is given by $r_i^j := \left(\langle \mathsf{K}^c, \mathsf{B}_i^j \rangle_{\mathbf{Q}}\right)$.

4.2.2 Step 2

The stepsize s for the computation of $\mathsf{P}^s := \mathsf{P}^c + s(\mathsf{O}^\perp - \mathsf{K}^c) = \mathsf{P}^c + s\mathsf{T}$ is chosen as in the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm. In the general case we take a small stepsize s, whose validity is tested with the Armijo rule. Only if g^c equals g^- we apply the step size selection of Pottmann and Hofer (2004) (see also Hofer and Pottmann (2004)).

If $g^c = g^- = 0$ the part of the footpoint cone Λ_Φ is determined by the vertex I, the footpoints $\mathsf{P}^c \in \Phi$ and $\mathsf{P}^- \in \Phi$ and the direction T of the tangent of the base curve in P^c. If $g^c = g^- \neq 0$ we can think again of a $(6N-g)$-dimensional manifold $\Psi \in \Phi$ which locally approximate Γ such that $T^{6N-g}_{\mathsf{P}^c}$ and $T^{6N-g}_{\mathsf{P}^-}$ are the tangent spaces in $\mathsf{P}^c \in \Psi$ resp. $\mathsf{P}^- \in \Psi$ with respect to Ψ. Now the stepsize selection can be done by developing the footpoint cone Λ_Ψ into the plane. The validity of the resulting stepsize must again be tested with the Armijo rule.

In the next step we project $\mathfrak{E}^s_N$ orthogonally onto M^6. This is done as in the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm by using the helical motion σ determined by the instantaneous screw $s\underline{\mathbf{q}}^N = (s\mathbf{q}^N, s\widehat{\mathbf{q}}^N)$.

4.2.3 Step 3

If $\mathsf{X}^{\varepsilon^\perp_N}$ and Y are colliding then we reduce the stepsize s with the stepsize reduction algorithm, which yields $\mathfrak{E}^+_N$. In the next step we project the remaining points $\mathfrak{E}^s_1, \dots, \mathfrak{E}^s_{N-1}$ orthogonally onto M^6 by solving the registration problem with known correspondences. This is a known algebraic problem of degree four, which was explicitly solved by Horn (1987) using unit quaternions. For further details see e.g. Belta and Kumar (2002). We denote the resulting points by $\mathfrak{E}^\perp_1, \dots, \mathfrak{E}^\perp_{N-1}$.

We can improve our iterate by taking into account that the trajectory of the barycenter under a geodesic motion has to be a straight line. In the general case the barycenters $\varepsilon^\perp_1(\mathbf{b}_\mathsf{X}), \dots, \varepsilon^\perp_{N-1}(\mathbf{b}_\mathsf{X})$ are not located on the line spanned by $\mathbf{b}_\mathsf{X}$ and $\varepsilon^+_N(\mathbf{b}_\mathsf{X})$ due to the two different kinds of used back projections onto M^6. Therefore we translate each point $\mathfrak{E}^\perp_i$ for $i = 1, \dots, N-1$ within the surface M^6 by $\mathfrak{H}_i := (\mathbf{h}_i, \mathbf{o}, \mathbf{o}, \mathbf{o})$ such that the approximated first energy

$$E_1 = \|\mathfrak{I} - \mathfrak{E}^\perp_1 - \mathfrak{H}_1\|^2 + \sum_{i=1}^{N-2} \|\mathfrak{E}^\perp_i + \mathfrak{H}_i - \mathfrak{E}^\perp_{i+1} - \mathfrak{H}_{i+1}\|^2 + \|\mathfrak{E}^\perp_{N-1} + \mathfrak{H}_{N-1} - \mathfrak{E}^+_N\|^2$$

is minimized. This quadratic function in $3(N-1)$ unknowns can be rewritten as

$$E_1(\mathsf{H}) = \mathsf{H}^T \cdot \mathbf{W} \cdot \mathsf{H} + \mathbf{k}^T \cdot \mathsf{H} + c, \tag{52}$$

with $\mathsf{H} := (\mathbf{h}_1, \dots, \mathbf{h}_{N-1})$. Because $\mathbf{W}$ is a symmetric positive definite matrix the unique global minimizer equals $-\mathbf{W}^{-1} \cdot \mathbf{k}$. By setting $\mathfrak{E}^+_i := \mathfrak{E}^\perp_i + \mathfrak{H}_i$ for

$i = 1, \dots, N-1$ we get the new iterate $\mathsf{P}^+ := (\mathfrak{E}_1^+, \dots, \mathfrak{E}_N^+)$.

If $\|\mathfrak{E}_N^+ - \mathfrak{E}_N^c\| < b$ and if the current contact value w^c also falls below a certain threshold v, then the $PD_G(\mathsf{X}, \mathsf{Y})$ algorithm is stopped.

4.3 Physical interpretation of PD_G

The pose X^γ which cause $PD_G(\mathsf{X}, \mathsf{Y})$ corresponds to the local minimizer $\mathfrak{G} = (\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3) \in \partial\mathcal{C}$. Due to considerations of section 3.3 and the well known result (see Arnol'd (1989)), that geodesic motions are motions of a free rigid body under its own inertia outside of any force field (⇒ kinetic energy is conserved), we can give the following interpretation of $PD_G(\mathsf{X}, \mathsf{Y})$:

Theorem 2 $PD_G(\mathsf{X}, \mathsf{Y})$ *equals* $\sqrt{K}$ *where* K *is the minimal needed total kinetic energy of* S *such that the rigid body* S^{id} *moves under its own inertia into the pose* S^γ *within one time unit.* S *denotes the ellipsoidal shell of Theorem 1.*

Remark: Assume that $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$ describes a rotation about one of the axis of the inertia ellipsoid of S through the angle ρ. Now the superposition of this uniform rotation and the uniform translation $\mathbf{g}_0$ is a geodesic motion. In such cases the only difference between $PD_S(\mathsf{X}, \mathsf{Y})$ and $PD_G(\mathsf{X}, \mathsf{Y})$ is that the angular velocity ω of (37) (chord length approximation of ρ) is replaced by the arc length of ρ.

5 Examples

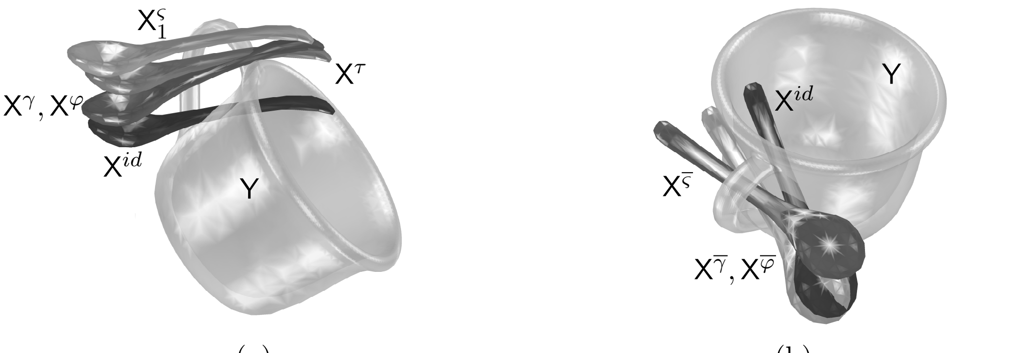
The numerical experiments have been performed with Matlab implementations on a AMD 64 Athlon Processor with 1 GB RAM. The main focus in implementing our algorithms was on the demonstration of the functionality and not on the improvement of the computation time. We are aware of the fact that our prototype implementations are not optimized and that there is a very large potential for speedup, especially in performing the collision test. Moreover some routines can be improved by additional preprocessing steps.

X^{id} denotes the given interpenetrating configuration and X_i^ς the collision-free inital guess, which was computed by method i ($i = 1, 2$) of section 3.4. As the result of the $PD_S(\mathsf{X}, \mathsf{Y})$ algorithm we obtain the pose X^φ which corresponds to the local minimizer $\mathfrak{F} \in \partial\mathcal{C}$. With the adapted version of this algorithm, we computed the pose X^τ which causes PD^t. As the result of the $PD_G(\mathsf{X}, \mathsf{Y})$ algorithm we obtain the pose X^γ which corresponds to the local minimizer $\mathfrak{G} \in \partial\mathcal{C}$.

Here for simplicity and comparability of the results the $PD_G(\mathsf{X},\mathsf{Y})$ algorithm was always performed with 10 intermediate positions. Clearly, an increase of these positions improves the accuracy at the cost of computation time. As the computation of the geodesic distance on M^6 can be decomposed into a translational and a rotational part (see Hofer and Pottmann (2004)) a more sophisticated choice of the number of intermediate positions should be based on the spherical distance of X^{id} and X^{φ}.

The cup examples

This example can also be found in the paper of Zhang et al. (2006), where the boundary surfaces of the fixed object (cup) Y and the moving body (spoon) X were approximated by triangular meshes $\mathcal{Y}$ resp. $\mathcal{X}$ with about 4200 resp. 170 vertices. We increased the number of vertices of the triangular meshes $\mathcal{Y}$ resp. $\mathcal{X}$ to 12500 resp. 1500 in order to improve the representation of the objects and therefore the accuracy of our results. Computing the collision-free starting configuration X_1^{ς} lasts about 2.1sec, where performing 60 times the ANN-algorithm requires about 87% of the total cputime.


X_1^{ς}
X^{τ}
$\mathsf{X}^{\gamma}, \mathsf{X}^{\varphi}$
X^{id}
Y
Y
X^{id}
$\mathsf{X}^{\bar\varsigma}$
$\mathsf{X}^{\bar\gamma}, \mathsf{X}^{\bar\varphi}$

(a)

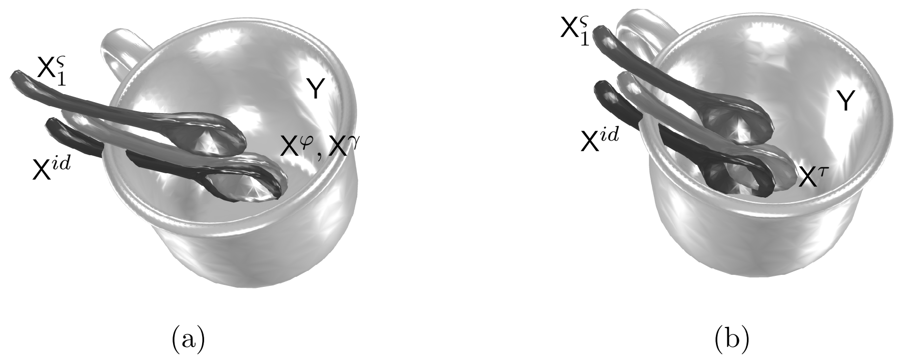
(b)

Fig. 6. Cup: Ex. 1

Table 1	distance	initial guess	calls	total time(tt)	collision test	Fig.
PD^t	1	X_1^{ς}	43	4.1719sec	87% of tt	6(a)
PD_S	0.8277	X_1^{ς}	43	4.4375sec	96% of tt	6(a)
PD_G	0.8301	X_1^{ς}	60	5.8430sec	90% of tt	6(a)
PD_S	0.6339	$\mathsf{X}^{\bar\varsigma}$	53	5.2031sec	81% of tt	6(b)
PD_G	0.6359	$\mathsf{X}^{\bar\varsigma}$	57	6.0781sec	79% of tt	6(b)

Example 1: The starting configuration X_1^{ς} as well as the solutions X^{φ} and X^{τ} of the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm resp. of the adapted one are displayed in *Fig. 6* (a). We also tested our $PD_G(\mathsf{X},\mathsf{Y})$ algorithm for the same starting configuration X_1^{ς} but its solution X^{γ} is too close to X^{φ} that one can see a difference. These two minimization problems seem to have the same local minimizer. The performance of these three algorithms is shown in table 1.

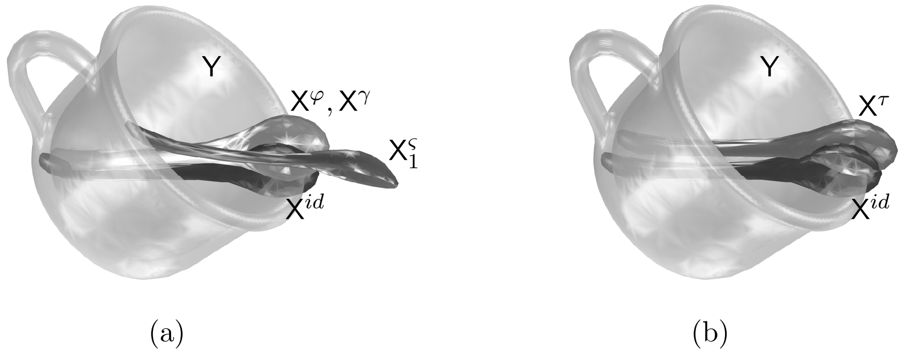
The obtained solutions are only local minima because running the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm resp. $PD_G(\mathsf{X},\mathsf{Y})$ algorithm starting with the pose $\mathsf{X}^{\overline{\varsigma}}$ illustrated in *Fig. 6* (b) yields another solution $\mathsf{X}^{\overline{\varphi}}$ resp. $\mathsf{X}^{\overline{\gamma}}$, which is closer to X^{id} as X^{φ} resp. X^{γ} (see table 1). The starting configuration $\mathsf{X}^{\overline{\varsigma}}$ could not be obtained by method 1 of section 3.4 because the ellipsoid E, which encloses X is to big for the hole of the handle. This example points out the limitations of the presented algorithm. The $PD_G(\mathsf{X},\mathsf{Y})$ algorithm with respect to the starting configuration $\mathsf{X}^{\overline{\varsigma}}$ has again a nearly identical solution to $\mathsf{X}^{\overline{\varphi}}$. The performance of the $PD_G(\mathsf{X},\mathsf{Y})$ resp. $PD_S(\mathsf{X},\mathsf{Y})$ algorithm with respect to $\mathsf{X}^{\overline{\varsigma}}$ is also shown in table 1.



X_1^{ς}
Y
$\mathsf{X}^{\varphi}, \mathsf{X}^{\gamma}$
X^{id}
X_1^{ς}
Y
X^{id}
X^{τ}

(a) (b)

Fig. 7. Cup: Ex. 2



Y
$\mathsf{X}^{\varphi}, \mathsf{X}^{\gamma}$
X_1^{ς}
X^{id}
Y
X^{τ}
X^{id}

(a) (b)

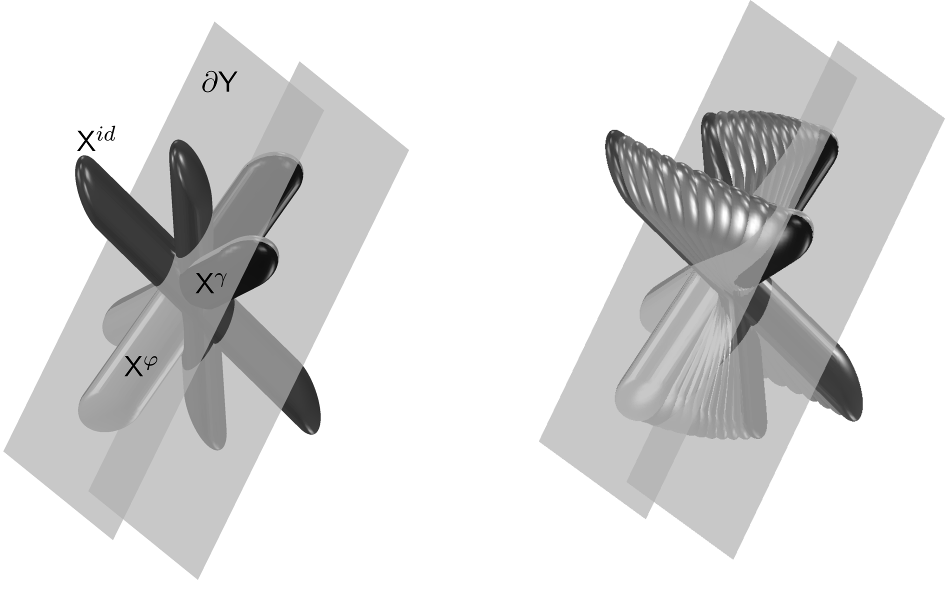
Fig. 8. Cup: Ex. 3

Examples 2 and 3: In example 2 and 3 illustrated in *Fig. 7* and *Fig. 8*, respectively, we run the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm and the $PD_G(\mathsf{X},\mathsf{Y})$ algorithm with respect to the starting configuration X_1^{ς}. As in example 1 the obtained poses X^{φ} and X^{γ} are too close to see a difference. The performance of the algorithms is shown in table 2, where the values for $PD_S(\mathsf{X},\mathsf{Y})$ and $PD_G(\mathsf{X},\mathsf{Y})$ are given with respect to PD^t. The pose X^{τ} which causes PD^t is illustrated in *Fig. 7* (b) and *Fig. 8* (b), respectively.

Table 2	distance	initial guess	calls	total time(tt)	collision test	Fig.
PD_S	$0.8070{\cdot}PD^t$	X_1^{ς}	55	5.1250sec	88% of tt	7
PD_G	$0.8071{\cdot}PD^t$	X_1^{ς}	63	6.4219sec	85% of tt	7
PD_S	$0.8276{\cdot}PD^t$	X_1^{ς}	38	5.0313sec	88% of tt	8
PD_G	$0.8285{\cdot}PD^t$	X_1^{ς}	46	6.1250sec	79% of tt	8

The gap example

We constructed this example in order to illustrate the slight difference between the poses X^{φ} and X^{γ}. The fixed object Y consists of two disjoint half-spaces and the moving object X looks like the character 'X'. Due to the fact that the boundary surface $\partial\mathsf{Y}$ of Y are two planes we can be sure that our algorithms do not only reach local minima but global ones. Moreover it should be said that the Euclidean maps φ and γ are pure rotations. The angle enclosed by these two spherical motions is only about 0.64°. The corresponding poses X^{φ} and X^{γ} as well as $\partial\mathsf{Y}$ and the 10 intermediate positions for the computation of $PD_G(\mathsf{X},\mathsf{Y})$ are displayed in *Fig. 9.*



(a) (b)

Fig. 9. The gap example

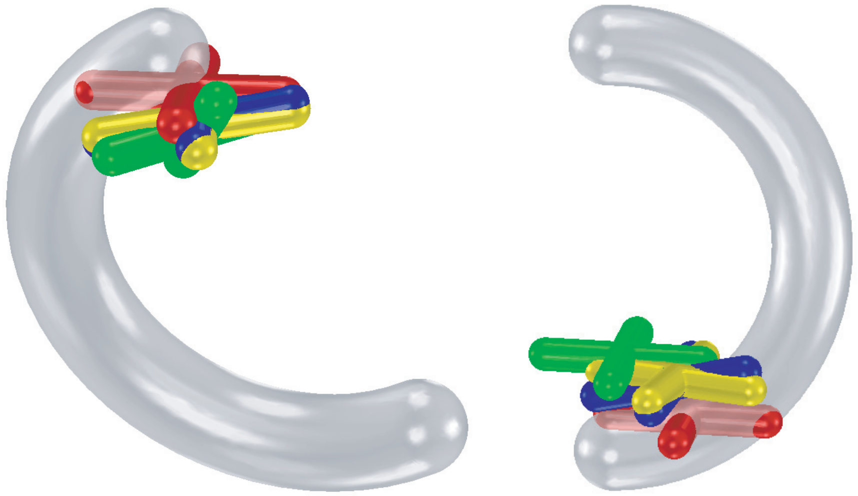
The cross examples

The triangular meshes $\mathcal{Y}$ and $\mathcal{X}$ of the half torus and cross, respectively, consists of 7000 and 11000 vertices. The computation of the collision-free initial guess X_1^{ς} lasts about 5.1sec, where performing 60 times the ANN-algorithm requires about 94% of the total cputime. In the following four examples we tested the $PD_S(\mathsf{X},\mathsf{Y})$ algorithm (see table 3) starting with X_1^{ς}.

Table 3	distance	initial guess	calls	total time(tt)	collision test	Fig.
PD_S	1	X_1^{ς}	58	17.9375sec	80% of tt	10(a)
PD_S	1.5396	X_1^{ς}	71	16.6406sec	79% of tt	10(b)
PD_S	1.1079	X_1^{ς}	79	21.5313sec	84% of tt	11(a)
PD_S	1.6650	X_1^{ς}	49	12.4219sec	72% of tt	11(b)

Moreover we tested the second method of section 3.4 for computing an initial guess. We run the corresponding X_2^{ς} algorithm for $b = -min(d(\mathbf{x},\mathsf{Y})|\mathbf{x} \in \mathcal{X}^{id})/5$. The performance of this algorithm is shown in table 4 and Fig. 12, respectively. The given examples demonstrate that the proposed algorithm works very well for examples, where the extent of overlap is small.

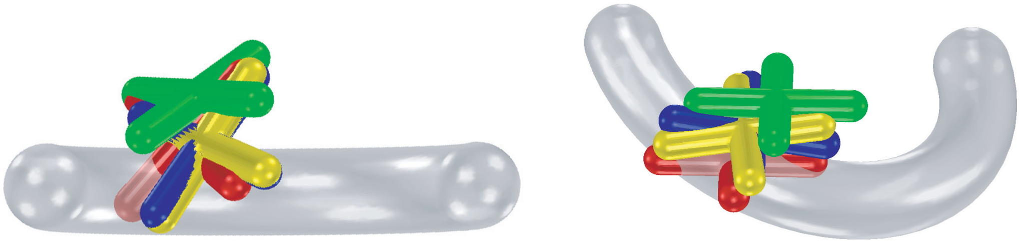
Table 4	calls	total time(tt)	collision test	$S(\mathsf{X}^{\varphi},\mathsf{X}_2^{\varsigma})$	Fig.
X_2^{ς}	10	1.5156sec	63% of tt	$0.10 \cdot S(\mathsf{X}^{\varphi},\mathsf{X}_1^{\varsigma})$	10(a)
X_2^{ς}	12	1.6094sec	63% of tt	$0.39 \cdot S(\mathsf{X}^{\varphi},\mathsf{X}_1^{\varsigma})$	10(b)
X_2^{ς}	9	1.6093sec	67% of tt	$0.07 \cdot S(\mathsf{X}^{\varphi},\mathsf{X}_1^{\varsigma})$	11(a)
X_2^{ς}	13	1.7813sec	59% of tt	$0.58 \cdot S(\mathsf{X}^{\varphi},\mathsf{X}_1^{\varsigma})$	11(b)



(a) (b)

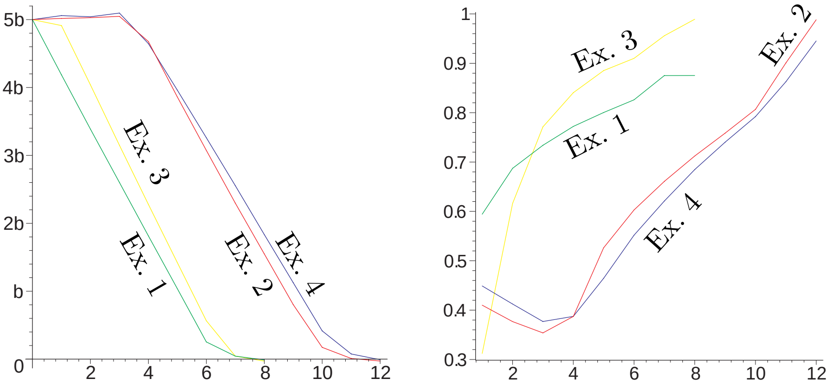
Fig. 10. Cross: (a) Ex. 1 (b) Ex. 2. Illustration of X^{id} (red), X_1^{ς} (green), X_2^{ς} (yellow) and X^{φ} (blue). (For interpretation of the references to colour in these figures, the reader is referred to the web version of this article.)

If we run the $PD_G(\mathsf{X},\mathsf{Y})$ algorithm with the initial guess X^{φ} we cannot achieve a significant improvement of the geodesic distance because the algorithm stops after a few iteration steps. The resulting local minimum is again too close to



(a) (b)

Fig. 11. Cross: (a) Ex. 3 (b) Ex. 4. Illustration of X^{id} (red), X_1^{ς} (green), X_2^{ς} (yellow) and X^{φ} (blue). (For interpretation of the references to colour in these figures, the reader is referred to the web version of this article.)



(a) (b)

Fig. 12. Performance of the X_2^{ς} algorithm: (a) The extent of overlap measured by $-min(d(\mathbf{x}, \mathsf{Y})|\mathbf{x} \in \mathcal{X})$ for each iteration step. (b) Evaluation of the screw $\mathbf{q}_{max}$ by the function $\kappa(\underline{\mathbf{q}})$ of (43).

X^{φ} that a graphical representation would make sense. It seems that in the general case these minima X^{φ} and X^{γ} are much closer to each other than in the synthetically generated example illustrated in *Fig. 9*.

6 Conclusion and future work

In this paper we defined the generalized penetration depth PD of two colliding rigid bodies X and Y with respect to the object-oriented metric S introduced by Hofer et al. (2004) which takes the mass distribution of the moving body X into consideration. We used a kinematic mapping which maps rigid body displacements to points of a 6-dimensional manifold M^6 in the 12-dimensional space of affine mappings. This space equipped with the above mentioned metric is a Euclidean space. We formulated PD as the solution of the constrained minimization problem of finding the closest footpoint on the boundary of the set of all points of M^6 which correspond to colliding configurations. Based on the theory of gliding motions, the closest footpoint with respect to the metric

S ($\Rightarrow PD_S$) can be computed with an adapted projected gradient algorithm. The outlined PD_S algorithm can easily be modified for the computation of the translational penetration depth as well. We also presented a geometric optimization algorithm which computes the closest footpoint with respect to the geodesic metric G of M^6 induced by the metric of the ambient space ($\Rightarrow PD_G$). Moreover we introduced two algorithms for the computation of a collision-free initial guess and gave a physical interpretation of PD_S and PD_G. We also tested the functionality of the presented geometric optimization algorithms based on prototype implementations and gave several examples.

A drawback of our approach is that our algorithms can only compute local minima, and thus they depend on the choice of the initial guess. Therefore we are working on a global approach by calculating a distance field on M^6 based on a fair discretization of $SO(3)$. Subdivision schemes for generating such fair discretizations of the spherical motion group were already presented (see Nawratil and Pottmann (in press)).

Acknowledgment

This research was carried out as part of the project S9206-N12 which was supported by the Austrian Science Fund (FWF). The authors would like to thank Michael Hofer for providing us some of his Matlab implementations, as well as Liangjun Zhang for placing the dataset of the cup and spoon example at our disposal.

The authors would also like to thank the reviewers for their useful comments and suggestions.

References

[1] Arnol'd, V.I., 1989. *Mathematical methods of classical mechanics.* 2^{nd} Edition, Springer, New York.

[2] Belta, C., and Kumar, V., 2002. *An SVD-projection method for interpolation on SE(3).* IEEE Trans. Robotics Automation 18 (3), 334–345.

[3] Botsch, M., Pauly, M., Gross, M., and Kobbelt, L., 2006. *PriMo: Coupled Prisms for Intuitive Surface Modeling.* Eurographics Symposium on Geometry Processing (K. Polthier and A. Sheffer, eds), 11–20.

[4] Cameron, S., 1997. *Enhancing GJK: Computing minimum and penetration distance between convex polyhedra.* In Proc. of IEEE International Conference on Robotics and Automation, 3112–3117.

[5] Hofer, M., 2004. *Variational motion design in the presence of obstacles.* Ph.D. Dissertation, Vienna University of Technology.

[6] Hofer, M., and Pottmann, H., 2004. *Energy-Minimizing Splines in Manifolds.* Transactions on Graphics 23 (3), 284–293 (Proceedings of ACM SIGGRAPH 2004).

[7] Hofer, M., Pottmann, H., and Ravani, B., 2004. *From curve design algorithms to the design of rigid body motions.* The Visual Computer 20 (5), 279–297.

[8] Horn, B.K.P., 1987. *Closed form solution of absolute orientation using unit quaternions.* J. Optical Society A 4, 629–642.

[9] Kim, B., and Rossignac, J., 2003. *Collision Prediction for Polyhedra under Screw Motions.* In Proc. of ACM Symposium on Solid Modeling and Applications, 4–10.

[10] Kim, Y., Lin, M., and Manocha, D., 2002. *Deep: Dual-space expansion for estimating penetration depth between convex polytopes.* In Proc. of IEEE International Conference on Robotics and Automation, 921–926.

[11] Nawratil, G., 2007. *New performance indices for 6R robots.* Mechanism and Machine Theory, 42 (11), 1499-1511.

[12] Nawratil, G., and Pottmann, H., in press. *Subdivision Schemes for the fair Discretization of the Spherical Motion Group.* Journal of Computational and Applied Mathematics.

[13] Pottmann, H., and Hofer, M., 2004. *Algorithms for constrained minimization of quadratic functions - geometry and convergence analysis.* Technical Report 121, Geometry Preprint Series, Vienna Univ. of Technology.

[14] Pottmann, H., Hofer, M., and Ravani, B., 2004. *Variational Motion Design.* On Advances in Robot Kinematics (J. Lenarcic and C. Galletti, eds.), 361–370.

[15] Pottmann, H., and Ravani, B., 2000. *Singularities of motions constrained by contacting surfaces.* Mechanism and Machine Theory 35 (7), 963–984.

[16] Pottmann, H., and Wallner, J., 2001. *Computational Line Geometry.* Springer-Verlag.

[17] Zhang, L., Kim, Y.J., Varadhan, G., and Manocha, D., 2006. *Generalized Penetration Depth Computation.* Symposium on Solid and Physical Modeling, 173–184.

[18] Zhang, L., Kim, Y.J., and Manocha, D., 2007. *A Fast and Practical Algorithm for Generalized Penetration Depth Computation.* In Proc. of Robotics: Science and Systems Conference (RSS07), 2007.

[19] Zhao, H.K., 2005. *Fast sweeping method for eikonal equations.* Math. Comp. 74, 603–627.

[20] Van den Bergen, G., 2001. *Proximity queries and penetration depth computation on 3D game objects.* In Proc. Game Developers Conference, 821–837.