On 3-D Point Set Matching With MAE and SAE Cost Criteria

Giuseppe C. Calafiore

***Abstract*—This paper deals with the problem of optimally matching two ordered sets of 3-D points by means of a rigid displacement. Contrary to the standard approach, where a sum-of-squared-errors criterion is minimized in order to obtain the optimal displacement, we here analyze the use of ℓ_∞ [maximum absolute error (MAE)] and ℓ_1 [sum of absolute errors (SAE)] cost criteria for determining the optimal matching. Two numerically efficient (polynomial time) algorithms are developed in this paper to compute an approximately optimal solution for the MAE and SAE matching problems.**

***Index Terms*—Least squares fitting, maximum absolute error (MAE), point set matching, sum of absolute error (SAE).**

I. Introduction

The PROBLEM of superimposing two ordered groups of 3-D points by means of a rigid displacement (rotation and translation) is a classical one in the robotics, manufacturing, and computer vision literature, where it is encountered under various names such as the absolute orientation, pose estimation, point-based registration, or matching problem. In the standard approach, a least squares [or sum of squared errors (SSE)] fitting criterion is employed, and the optimal displacement is determined either by using a rotation matrix and translation vector parameterization (see, e.g., [1] and [20]) or by using quaternions [6], [10].

However, while the least squares matching problem is mainstream and relatively simple to solve [for instance, by using singular value decomposition (SVD)], it also has its drawbacks. First, it is well known that a least squares cost criterion is optimal from a statistical point of view only when the mismatch errors can be modeled as normal (Gaussian) random variables and can therefore be improper and nonrobust in the presence of nonstandard errors or outliers (see [15]). Second, the minimization of a least squares criterion does not permit one to control the matching errors on the individual points and does not guarantee satisfaction of *a priori* fixed bounds on these errors. This issue is of particular relevance in manufacturing when it is necessary to ascertain, for instance, if a machined part can match a design template within *a priori* assigned tolerances; see [3] for a discussion of the matching problem in the constrained case.

A similar issue also arises in guidance systems for surgery based on medical imaging. In this setting, image registration is accomplished by intraoperatively matching marker points on the patient to tomographic images that were obtained preoperatively (see, e.g., [7]). The control over the maximum error instead of the least squares error is, in this case, clearly of paramount importance for an accurate surgical intervention.

In this paper, we propose the use of two alternative cost criteria for point matching and present two efficient polynomial-time algorithms for computing suboptimal solutions of the corresponding problems. Specifically, we consider a cost criterion based on the maximum Euclidean distance among corresponding point pairs (the ℓ_∞ or maximum-absolute-error (MAE) criterion) and a criterion based on the average of the Euclidean distances of the matching point pairs (the ℓ_1 or sum of absolute error (SAE) criterion).

The ℓ_1 criterion is suitable in the presence of nonnormal noise or possible outliers in the data. The motivation for this comes from the fact that the least squares criterion tends to be sensitive to large residuals, while this effect is reduced if a criterion with linear, instead of quadratic, residual terms is used. This point is extensively accounted for in the literature on robust statistics (see, e.g., [12]), and the SAE criterion is commonly used in image analysis and shape matching [16], [17], [21].

The ℓ_∞ criterion, instead, is unsuitable in applications that involve noise-corrupted data (such as image analysis) since its performance is entirely driven by the worst-case matching points, i.e., no error averaging occurs. On the other hand, this criterion is advisable in applications such as tolerance inspection, manufacturing, or robotic surgery, where the registration noise can be controlled and rendered small *a priori* and where a minimal worst-case matching error between point pairs needs to be guaranteed.

To the best of the author's knowledge, no efficient polynomial-time algorithm is currently available for solving either the ℓ_1 or the ℓ_∞ matching problem, even when point correspondence is given, as it is assumed in this paper. In regard to this latter matter and to further clarify the scope of this contribution, we remark that several generalizations exist in the literature dealing with the SSE matching problem in cases where no ordering (point correspondence) is given or the number of points in the two sets is unequal (see, for instance, [8] and [19] and the variations on the iterative closest point algorithm in [5] and the references therein). However, since the matching problem without point ordering is highly nonlinear and intrinsically of a combinatorial nature, all the cited methods rely on optimization heuristics that may be prone to convergence problems and/or convergence to local extrema. In this paper, we shall not consider the problem of determining point correspondences. Instead, we focus on the development of efficient polynomial-time algorithms that approximately solve

Manuscript received April 28, 2006; revised October 2, 2006. This paper was recommended by Associate Editor J. Wu.

The author is with the Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy (e-mail: giuseppe.calafiore@polito.it).

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

Digital Object Identifier 10.1109/TSMCA.2007.914774

the MAE and SAE matching problems under a preassigned point ordering. Once these "kernel" problems are efficiently solved, however, one may develop some "outer" heuristic (such as random sample consensus or iterative closest point) in order to address the more complicated situation when point ordering is unknown. This is left for future work.

This paper is organized as follows. In Section II, the SSE, MAE, and SAE matching problems are formally stated. Section III reviews the solution of the standard SSE problem, whereas our proposed algorithms for the solution of the MAE and SAE problems are reported in Sections IV and V, respectively. Some numerical examples are presented in Section VI, and conclusions are drawn in Section VII.

II. Problem Statement

Let $\mathcal{A} \doteq \{a^{(i)} \in \mathbb{R}^3,\ \ i = 1, \ldots, n\}$ and $\mathcal{B} \doteq \{b^{(i)} \in \mathbb{R}^3,$ $i = 1, \ldots, n\}$ be the two ordered sets of points, and let

$$A \doteq [a^{(1)}, \ldots, a^{(n)}] \quad B \doteq [b^{(1)}, \ldots, b^{(n)}]$$

be matrices containing by columns the points in sets $\mathcal{A}$ and $\mathcal{B}$, respectively. We define a displacement operator $\Gamma(\cdot)$ that acts on a point set by performing a rigid roto-translation (displacement) on it. The displacement operator is described by a rotation, which is expressed by means of a rotation matrix $R \in \mathrm{so}(3)$, where so(3) is the special orthogonal group of real 3×3 rotation matrices, and a translation, expressed by a vector $t \in \mathbb{R}^3$. Thus, if a displacement Γ is applied to the point matrix B, a displaced matrix B_d is obtained

$$B_\mathrm{d} \doteq \Gamma(B) = RB + tu^\top$$

where $u \in \mathbb{R}^n$ is a vector of ones. The columns of B_d are the displaced points

$$b_\mathrm{d}^{(i)} = Rb^{(i)} + t, \qquad i = 1, \ldots, n.$$

The objective of this paper is to present computational techniques that permit the determination of an optimal displacement (i.e., R and t) that "superimposes" the point set $\mathcal{B}$ to the "template" point set $\mathcal{A}$ by minimizing some suitable error measure.

Let us first recall three standard norms on vectors: The ℓ_1, ℓ_2, and ℓ_∞ norms of vector $x \in \mathbb{R}^m$ are defined, respectively, as

$$\|x\|_1 \doteq \sum_{i=1}^{m} |x_i| \quad \|x\|_2 \doteq \sqrt{\sum_{i=1}^{m} x_i^2} \quad \|x\|_\infty \doteq \max_{i=1,\ldots,m} |x_i|.$$

The Euclidean distances between the template points $a^{(i)}$ and the corresponding displaced points $b_\mathrm{d}^{(i)}$ are

$$d_i \doteq \left\|a^{(i)} - b_\mathrm{d}^{(i)}\right\|_2 = \left\|a^{(i)} - Rb^{(i)} - t\right\|_2, \qquad i = 1, \ldots, n.$$

Let $d \doteq [d_1, \ldots, d_n]^\top$ be the vector of distances between the corresponding point pairs: The overall matching error between A and B_d can be measured by a metric based on a norm of d. Depending on the norm employed, this error is written as

$$\begin{aligned}
e_1(R,t) &\doteq \frac{1}{n}\|d\|_1 \\
&= \frac{1}{n}\sum_{i=1}^{n} \left\|a^{(i)} - Rb^{(i)} - t\right\|_2 \quad \text{(SAE)} \\
e_2(R,t) &\doteq \frac{1}{\sqrt{n}}\|d\| \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} \left\|a^{(i)} - Rb^{(i)} - t\right\|_2^2\right)^{1/2} \quad \text{(SSE)} \\
e_\infty(R,t) &\doteq \|d\|_\infty \\
&= \max_{i=1,\ldots,n} \left\|a^{(i)} - Rb^{(i)} - t\right\|_2 \quad \text{(MAE)}.
\end{aligned}$$

Based on these error criteria, we introduce the following three optimal matching problems:

$$\begin{aligned}
\mathcal{P}_1 &: \min_{R\in\mathrm{so}(3),t\in\mathbb{R}^3} e_1(R,t) \\
\mathcal{P}_2 &: \min_{R\in\mathrm{so}(3),t\in\mathbb{R}^3} e_2(R,t) \\
\mathcal{P}_\infty &: \min_{R\in\mathrm{so}(3),t\in\mathbb{R}^3} e_\infty(R,t).
\end{aligned} \tag{1}$$

Notice that problem $\mathcal{P}_1$ amounts to determining the displacement that minimizes the average distance between the template points $a^{(i)}$ and the corresponding roto-translated points $b_\mathrm{d}^{(i)}$. With $\mathcal{P}_2$, we aim at minimizing the average squared distance between the matching points, whereas with $\mathcal{P}_\infty$, we aim at minimizing the maximum distance between these points.

Problem $\mathcal{P}_2$ is well known in the literature, and an exact closed-form solution for it is available. In Section III, we briefly recall this solution, which also serves as a starting point for the solution algorithms for problems $\mathcal{P}_1$ and $\mathcal{P}_\infty$. To the best of this author's knowledge, no exact and efficient solution method currently exists for these two latter problems. In Sections IV and V, we discuss two ad-hoc iterative algorithms for the approximate solution of $\mathcal{P}_1$ and $\mathcal{P}_\infty$.

III. SSE Displacement Problem

We here briefly review the solution of the classical SSE optimal displacement problem $\mathcal{P}_2$ (see [1] and [20]). The main result is reported in the next theorem, whose proof is sketched for completeness in the Appendix.

Theorem 1 (Optimal Solution of $\mathcal{P}_2$): Let

$$W \doteq I_n - \frac{1}{n}uu^\top \quad \tilde{A} = AW \quad \tilde{B} = AW \tag{2}$$

where I_n denotes the $n \times n$ identity matrix, and let $U\Sigma V^\top$ be the singular value factorization of $\tilde{B}\tilde{A}^\top$, where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$ is a diagonal matrix of the ordered singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. An optimal solution to problem (1) is given by

$$R_{2,\mathrm{opt}} = \begin{cases} VU^\top, & \text{if } \det \tilde{B}\tilde{A}^\top \geq 0 \\ V\mathrm{diag}(1,1,-1)U^\top, & \text{if } \det \tilde{B}\tilde{A}^\top < 0 \end{cases}$$

$$t_{2,\mathrm{opt}} = \frac{1}{n}(A - R_{2,\mathrm{opt}}B)u.$$

IV. MAE Displacement Problem

In this section, we present a sequential method for determining a suboptimal solution to problem $\mathcal{P}_\infty$. The key idea is to first determine an initial optimal SSE displacement (using the result in Theorem 1) and then to iteratively perform additional corrective displacements in order to adjust the solution by moving in the direction of minimizing the ℓ_∞ norm of the distance vector d.

If (R_1, t_1) is the initial displacement, the initial displaced matrix $B_{\mathrm{d}1}$ is given by

$$B_{\mathrm{d}1} = R_1 B + t_1 u^\top.$$

Applying a second corrective displacement (R_2, t_2) (we shall discuss later in Section IV-A how such a corrective displacement is computed) to $B_{\mathrm{d}1}$ yields

$$B_{\mathrm{d}2} = R_2 B_{\mathrm{d}1} + t_2 u^\top = R_2 R_1 B + R_2 t_1 u^\top + t_2 u^\top.$$

Proceeding in this way, at the kth iteration, the overall displaced matrix would be

$$B_{\mathrm{d}k} = (R_k \cdots R_1)B + ((R_k \cdots R_2)t_1 + (R_k \cdots R_3)t_2 + \cdots + R_k t_{k-1} + t_k)u^\top$$

corresponding to a total rotation matrix $(R_k \cdots R_1)$ and translation vector $(R_k \cdots R_2)t_1 + (R_k \cdots R_3)t_2 + \cdots + R_k t_{k-1} + t_k$.

The logical scheme of the algorithm is as follows. The key step 5) in the algorithm is explained in detail in Section IV-A.

Algorithm 1 (MAE Point Matching): Set exit relative level $\eta \in (0, 1)$. Given matrices A, B, compute the optimal SSE displacement and initialize

$$k = 1, \quad R_k = R_{2,\mathrm{opt}}, \quad t_k = t_{2,\mathrm{opt}}, \quad R = R_k,$$
$$t = t_k, \quad e = \text{some large number}.$$

repeat

1) build the displaced matrix

$$B_{\mathrm{d}k} = RB + tu^\top$$

2) evaluate the cost function

$$e_+ \doteq \max_{i=1,\ldots,n} \left\| a^{(i)} - b_{\mathrm{d}k}^{(i)} \right\|_2$$

3) exit condition: if $(e - e_+)/e < \eta$, stop and return R, t

4) set $k = k + 1$

5) compute (R_k, t_k) by solving the corrective ℓ_∞ displacement subproblem (see Algorithm 2 in Section IV-A)

6) update the overall displacement parameters: $R = R_k R$, $t = R_k t + t_k$

7) set $e = e_+$.

Algorithm 1 iteratively computes adjustment displacements until no further significant improvement is observed in the objective cost. Clearly, the central phase in the algorithm is the computation of the corrective displacement in step 5). This subproblem is analyzed in detail in the next section.

A. Computing ℓ_∞ Adjustment Displacements

Suppose that we are given the template matrix A and the displaced matrix $B_{\mathrm{d}k-1}$ at the $(k-1)$th iteration of Algorithm 1. The corrective displacement subproblem amounts to determining a rotation matrix R_k of "small" angle $\theta_k \leq \gamma$ and translation t_k such that

$$\max_{i=1,\ldots,n} \left\| a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 \tag{3}$$

is approximately minimized. To this purpose, we introduce an approximate parameterization of small rotations in the next section.

1) Exponential Mapping, Approximation, and Reconstruction: A classical result from a group theory (see, e.g., [9]) states that any rotation matrix $R \in \mathrm{so}(3)$ can be parameterized in terms of a matrix $S \in \mathrm{sk}(3)$, where $\mathrm{sk}(3)$ denotes the space of 3×3 skew-symmetric matrices. Specifically, if $h \in \mathbb{R}^3$ is a unit vector (i.e., such that $\|h\| = 1$) denoting the rotation axis, and θ is the angle of rotation around h, then the rotation matrix representing this rotation is given by the exponential mapping[1]

$$\mathrm{e}^S \doteq I + S + \frac{1}{2!}S^2 + \frac{1}{3!}S^3 + \cdots \tag{4}$$

where

$$S \doteq H\theta \quad H = \mathrm{skew}(h) \doteq \begin{bmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix}$$

with $\|H\|_F = \sqrt{2}$ being the Frobenius norm of H (the Frobenius norm of a matrix is the square root of the sum of the squares of all the elements of the matrix).

Unfortunately, due to its nonlinearity, the exponential parameterization (4) is not very useful for our subsequent developments. We shall hence introduce a working scheme that consists in two steps.

First, for small angles of rotation, rotation matrices are approximately parameterized by a first-order truncation of (4)

$$\tilde{R} \doteq I + S \tag{5}$$

where $S \doteq \mathrm{skew}(s)$, with $s = \theta h$. The quality of the approximation is controlled by the norm of the matrix parameter $S \in \mathrm{sk}(3)$, $\|S\|_F = \sqrt{2}\|s\|_2$.

Second, since matrix $\tilde{R}$ is not a rotation, we introduce a reconstruction step in order to retrieve an actual rotation matrix from (5). To this end, we pose and solve the following problem:

[1] An explicit expression for e^S is provided by the Rodrigues formula: $\mathrm{e}^S = I + H\sin\theta + H^2(1 - \cos\theta)$.

Determine a matrix $R \in \mathrm{so}(3)$ so that $\|R - \tilde{R}\|_F$ is minimized. This problem can actually be solved by the SVD of $\tilde{R}$ (see, e.g., [11]). Let $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$ be the SVD of $\tilde{R}$, then the rotation matrix closest to $\tilde{R}$ in the Frobenius norm is

$$R = \tilde{U}\tilde{V}^\top.$$

Letting $\Delta R \doteq \tilde{R} - R$, the (squared) reconstruction error can be quantified as

$$\|\Delta R\|_F^2 = \sum_{i=1}^{3} (\tilde{\sigma}_i - 1)^2$$

where $\tilde{\sigma}_i$'s are the singular values of $\tilde{R}$, which are explicitly given by

$$\begin{aligned}\tilde{\sigma}_1 = \tilde{\sigma}_2 &= \sqrt{1 + \|s\|_2^2}\\ \tilde{\sigma}_3 &= 1\end{aligned}$$

and therefore, the reconstruction error is

$$\begin{aligned}\|\Delta R\|_F &= \sqrt{2}\left(\sqrt{1 + \|s\|_2^2} - 1\right)\\ &= \frac{1}{\sqrt{2}}\|s\|_2^2 + O\left(\|s\|_2^4\right)\\ &\simeq \frac{1}{\sqrt{2}}\|s\|_2^2, \quad \text{for small } \|s\|_2. \end{aligned} \tag{6}$$

2) Approximate ℓ_∞ Adjustment Displacement Problem: We are now in a position to state and solve the approximate adjustment displacement problem. Let $\gamma > 0$ be a bound on the correction angle, and let $\tilde{R}_k = I + S_k$ be the approximate adjustment rotation. The set of possible adjustments over which we search for the optimum is the convex set

$$\mathcal{R}(\gamma) \doteq \left\{\tilde{R}_k : \tilde{R}_k = I + S_k,\ S_k \in \mathrm{sk}(3), \|S_k\|_F \leq \gamma\sqrt{2}\right\}$$

where the constraint $\|S_k\|_F \leq \gamma\sqrt{2}$ imposes that the adjustment rotation angle be no larger than γ. The approximate adjustment problem is then written as

$$\min_{\tilde{R}_k \in \mathcal{R}(\gamma), t_k \in \mathbb{R}^3} \max_{i=1,\ldots,n} \left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2.$$

Since $\max_{i=1,\ldots,n} x_i \leq \tilde{e}_+$ if and only if $x_i \leq \tilde{e}_+$ for all $i = 1, \ldots, n$, introducing a slack variable $\tilde{e}_+$, the aforementioned problem can be expressed in the equivalent form as

$$\min_{\tilde{R}_k \in \mathcal{R}(\gamma), t_k \in \mathbb{R}^3, \tilde{e}_+} \tilde{e}_+ \quad \text{subject to:}$$

$$\left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 \leq \tilde{e}_+, \qquad i = 1, \ldots, n.$$

Making all constraints explicit, we finally obtain

$$\min_{S_k \in \mathrm{sk}(3), t_k \in \mathbb{R}^3, \tilde{e}_+} \tilde{e}_+, \ \text{subject to:}$$

$$\left\| a^{(i)} - b_{\mathrm{d}k-1}^{(i)} - S_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 \leq \tilde{e}_+, \quad i = 1, \ldots, n \tag{7}$$

$$\|S_k\|_F \leq \gamma\sqrt{2}. \tag{8}$$

A key observation at this point is that problem (7) is a convex optimization program, and in particular, it is a convex second-order cone program (SOCP). The global optimum of (7) can therefore be computed with great computational efficiency via specialized interior-point methods for SOCP optimization (see, e.g., [2] and the references therein). An assessment of the numerical effort required for solving (7) using a general-purpose algorithm for SOCP is provided in Section IV-A3.

Once (7) is solved, we let $\tilde{e}_+^*$ denote the objective value at the optimum, and reconstruct a rotation matrix R_k from $\tilde{R}_k = I + S_k$, using the technique described in Section IV-A1. That is, we compute the SVD $\tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ of $\tilde{R}_k$, and set $R_k = \tilde{U}_k \tilde{V}_k^\top$.

Notice that when the reconstructed R_k is plugged into the objective (3), an objective value that is different from the computed $\tilde{e}_+^*$ is obtained due to the reconstruction error. However, this error can be bounded *a priori* as follows. Letting $\Delta R_k \doteq \tilde{R}_k - R_k$, we have that

$$\begin{aligned}&\left\| a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2\\ &\quad= \left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k + \Delta R_k b_{\mathrm{d}k-1}^{(i)} \right\|_2\\ &\quad\leq \left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 + \left\| \Delta R_k b_{\mathrm{d}k-1}^{(i)} \right\|_2\\ &\quad\leq \left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 + \|\Delta R_k\|_F \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2.\end{aligned}$$

Now, since $\tilde{R}_k$ is an optimal solution of (7), and since $\|\Delta R_k\|_F \leq \sqrt{2}(\sqrt{1+\gamma^2} - 1)$ [this follows from (6) and the bound (8)], we conclude the previous chain of inequalities with

$$\left\| a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 \leq \tilde{e}_+^* + \sqrt{2}\left(\sqrt{1+\gamma^2} - 1\right) \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2$$

from which we derive the following global *a priori* bound on the discrepancy between $\tilde{e}_+^*$ and $e_+ \doteq \max_{i=1,\ldots,n} \|a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k\|_2$:

$$\begin{aligned}e_+ - \tilde{e}_+^* &\leq \sqrt{2}\left(\sqrt{1+\gamma^2} - 1\right) \max_{i=1,\ldots,n} \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2\\ &= \left(\frac{1}{\sqrt{2}}\gamma^2 + O(\gamma^4)\right) \max_{i=1,\ldots,n} \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2.\end{aligned}$$

Thus, $e_+ - \tilde{e}_+^*$ can be controlled *a priori* to be small by choosing a sufficiently small γ.

We can now outline the algorithm to be used for the correction displacement step.

Algorithm 2 (ℓ_∞ Corrective Displacement Subproblem): Fix a suitably small $\gamma > 0$. Given matrices A, $B_{\mathrm{d}k-1}$ do:

1) compute $\tilde{R}_k = I + S_k$, t_k, by solving SOCP problem (7)

2) compute SVD: $\tilde{R}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$

3) reconstruct a rotation matrix $R_k = \tilde{U}_k \tilde{V}_k^\top$

5) return R_k and t_k and finish.

3) Numerical Complexity of the ℓ_∞ Adjustment Displacement Problem: The numerical effort required by Algorithm 2 is mainly concentrated in steps 1) and 2). Step 2) simply requires the singular value factorization of a 3×3 real matrix, which can be performed in $O(1)$ operations.

Step 1) requires, instead, the numerical solution of an SOCP with a fixed number of variables (namely, seven variables: the three free entries of the skew-symmetric matrix S_k, the three elements of t_k, and the scalar $\tilde{e}_+$) and $n+1$ second-order cone constraints. The effort required for this step depends on the specific type of numerical optimization algorithm that one employs. For instance, by using a general-purpose primal-dual interior-point solution method, the numerical effort for solving (7) grows essentially as $O(\sqrt{n})$, which is according to the analysis developed in [13] (Section IV). This shows that the proposed approach is numerically efficient and actually subpolynomial in the number n of data points.

V. SAE Displacement Problem

This section discusses a suboptimal sequential method for approximating the solution of problem $\mathcal{P}_1$. We employ the same framework developed in Section IV for the $\mathcal{P}_\infty$ problem. The overall scheme of the algorithm is given next. The key step 5) in the algorithm is detailed in Section V-A.

Algorithm 3 (SAE Point Matching): Set exit relative level $\eta \in (0,1)$. Given matrices A, B, compute the optimal SSE displacement and initialize

$$k=1, \quad R_k = R_{2,\mathrm{opt}}, \quad t_k = t_{2,\mathrm{opt}},$$
$$R = R_k, \quad t = t_k, \quad e = \text{some large number}.$$

repeat

1) build the displaced matrix

$$B_{\mathrm{d}k} = RB + tu^\top$$

2) evaluate the cost function

$$e_+ \doteq \sum_{i=1}^n \left\| a^{(i)} - b_{\mathrm{d}k}^{(i)} \right\|_2$$

3) exit condition: if $(e - e_+)/e < \eta$, stop and return R, t

4) set $k = k+1$

5) compute (R_k, t_k) by solving the corrective ℓ_1 displacement subproblem (see Algorithm 4 in Section V-A)

6) update the overall displacement parameters: $R = R_k R$, $t = R_k t + t_k$

7) set $e = e_+$.

A. Computing ℓ_1 Adjustment Displacements

Suppose that we are given the template matrix A and the displaced matrix $B_{\mathrm{d}k-1}$ at the $(k-1)$th iteration of Algorithm 3. The ℓ_1 corrective displacement subproblem amounts to determining a rotation matrix R_k of "small" angle $\theta_k \le \gamma$ and translation t_k such that

$$\sum_{i=1}^n \left\| a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2$$

is approximately minimized.

By using the approximation and reconstruction approach discussed in Section IV-A1, we let $\gamma > 0$ be a bound on the correction angle, and let $\tilde{R}_k = I + S_k$ be the approximate adjustment rotation. The approximate ℓ_1 adjustment problem is then written as

$$\min_{\tilde{R}_k \in \mathcal{R}(\gamma), t_k \in \mathbb{R}^3} \sum_{i=1}^n \left\| a^{(i)} - \tilde{R}_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 .$$

Introducing a slack variable vector $\xi = [\,\xi_1\, \xi_2 \ldots, \xi_n\,]^\top$, this problem can be expressed explicitly in the form of a convex SOCP

$$\begin{aligned} &\min_{S_k \in \mathrm{sk}(3), t_k \in \mathbb{R}^3, \xi \in \mathbb{R}^n} \sum_{i=1}^n \xi_i \quad \text{subject to :} \\ &\left\| a^{(i)} - b_{\mathrm{d}k-1}^{(i)} - S_k b_{\mathrm{d}k-1}^{(i)} - t_k \right\|_2 \le \xi_i, \qquad i = 1, \ldots, n \\ &\|S_k\|_F \le \gamma\sqrt{2}. \end{aligned} \tag{9}$$

We observe again that the global optimum of problem (9) can be computed with great computational efficiency using standard algorithms for SOCP optimization. Once problem (9) is solved, we reconstruct a rotation matrix R_k from $\tilde{R}_k = I + S_k$ by computing the SVD $\tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ of $\tilde{R}_k$ and setting $R_k = \tilde{U}_k \tilde{V}_k^\top$.

The error in the objective, which is induced by the reconstruction, can be analyzed in a way analogous to that illustrated in Section IV-A2. The result is the following: Let $\tilde{e}_+^*$ denote the objective value at the optimum of problem (9), and let $e_+ \doteq \sum_{i=1}^n \|a^{(i)} - R_k b_{\mathrm{d}k-1}^{(i)} - t_k\|_2$. Then, it holds that

$$\begin{aligned} e_+ - \tilde{e}_+^* &\le \sqrt{2}(\sqrt{1+\gamma^2} - 1) \sum_{i=1}^n \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2 \\ &= \left(\frac{1}{\sqrt{2}}\gamma^2 + O(\gamma^4) \right) \sum_{i=1}^n \left\| b_{\mathrm{d}k-1}^{(i)} \right\|_2 . \end{aligned}$$

We next outline the algorithm to be used for the ℓ_1 correction displacement step.

Algorithm 4 (ℓ_1 Corrective Displacement Subproblem): Fix a suitably small $\gamma > 0$. Given matrices A, $B_{\mathrm{d}k-1}$ do:

1) compute $\tilde{R}_k = I + S_k$, t_k, by solving SOCP problem (9)

2) compute SVD: $\tilde{R}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$

3) reconstruct a rotation matrix $R_k = \tilde{U}_k \tilde{V}_k^\top$

4) return R_k and t_k and finish.

1) Numerical Complexity of the ℓ_1 Adjustment Displacement Problem: Following the reasoning that is similar to the one reported in Section IV-A3, we see that the numerical effort required by Algorithm 4 is mainly due to the solution of the SOCP (9). In this case, the SOCP has $n+6$ variables and $n+1$ second-order cone constraints. A general-purpose primal-dual interior-point solution algorithm, such as the one described and analyzed in [13], would then require $O(\sqrt{n})O(n^3)$ operations to numerically solve (9).

VI. Numerical Examples

The examples in this section have been coded in Matlab ver. 7.2 and run on a Windows XP platform with an AMD Dual

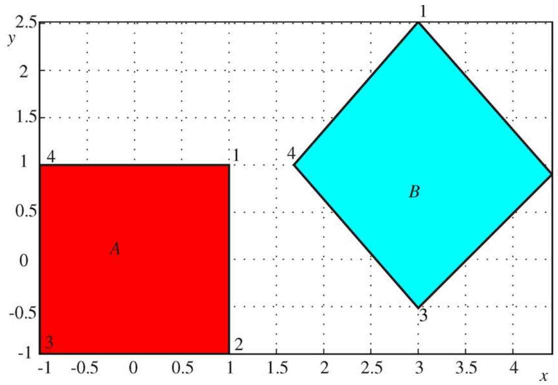

Fig. 1. Point sets A and B for Example 1.

Opteron 280 CPU equipped with 3 G-B of RAM. For the solution of SOCP, we used the freely available YALMIP interface, with SeDuMi solver (see [14]). The purpose of these examples is not to demonstrate which error measure is more suitable than another, since this issue is dependent on the application at hand. Such kind of discussion is, however, available in several papers (see, e.g., [18] and the references therein). Instead, the examples here are aimed at highlighting the functioning and numerical efficiency of the proposed algorithms.

A. Example 1

We start with a simple example consisting of $n = 4$ points, as shown in Fig. 1, with

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$B = \begin{bmatrix} 3.0000 & 4.4142 & 3.0000 & 1.6858 \\ 2.5142 & 0.9000 & -0.5142 & 1.0000 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We would like to determine a displacement that superimposes B on A, minimizing the ℓ_∞ matching error. Notice that the ℓ_2 optimal displacement gives in this case

$$R_{2,\mathrm{opt}} = \begin{bmatrix} 0.7193 & 0.6947 & 0 \\ -0.6947 & 0.7193 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$
$$t_{2,\mathrm{opt}} = \begin{bmatrix} -2.8532 \\ 1.4002 \\ 0 \end{bmatrix}$$

corresponding to an ℓ_∞ matching error that is equal to 0.1347. This error can, indeed, be reduced using Algorithm 1. Setting $\gamma = 0.0175$ and exit tolerance $\eta = 10^{-6}$, Algorithm 1 converged in three iterations to the optimal solution

$$R_{\infty,\mathrm{opt}} = \begin{bmatrix} 0.7071 & 0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$
$$t_{\infty,\mathrm{opt}} = \begin{bmatrix} -2.8284 \\ 1.4142 \\ 0 \end{bmatrix}$$

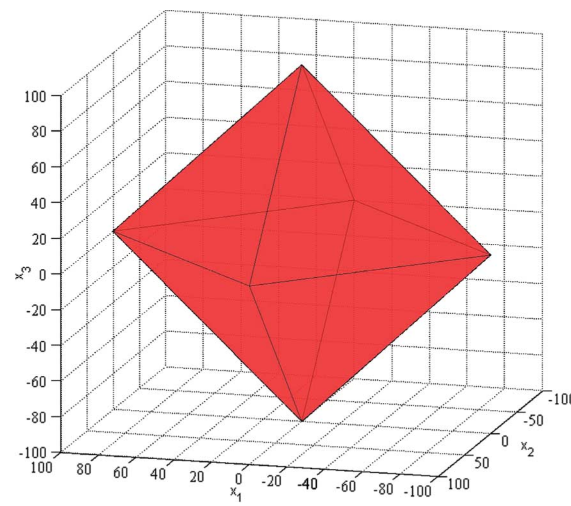

Fig. 2. Graphical representation of the set $\{x \in \mathbb{R}^3 : \|x\|_1 = 100\}$ used in Example 2.

yielding an ℓ_∞ matching error that is equal to 0.1. By using the proposed algorithm based on the MAE criterion, we were hence able to reduce the maximum matching distance across all point pairs by a 25.7% factor.

B. Example 2

In this second example, points in the template set A are generated uniformly at random on the surface of the set $\{x \in \mathbb{R}^3 : \|x\|_1 = 100\}$ shown in Fig. 2 (see [4] for a description of techniques for uniform generation in norm-bounded sets).

The point set B is constructed by adding a random perturbation to each entry of A. To simulate the presence of outliers, this perturbation is selected with a probability of 0.9 from a uniform distribution on $[-1, 1]$ and with a probability of 0.1 from a uniform distribution on $[-10, 10]$. Then, the perturbed points are rotated by 45° around the x-axis and translated by vector $[10\ 10\ 10]^\top$.

The SSE, MAE, and SAE optimal displacement problems are then solved for the pairs (A, B), with algorithm parameters set to $\gamma = 0.0524$ (corresponding to approximately 3° for corrective rotations), $\eta = 10^{-5}$.

The results obtained for various values of n (the number of points) are summarized in Table I, which also reports the number of iterations required to reach convergence, the running times, and the actual number n_{outl} of outliers present in each data instance. The figures in the table, which are marked with "no outl.," refer to cost values obtained on the cleaned point set, that is, on the point set with outliers removed.

Notice that, on average, over the five test cases, the e_∞ matching error obtained by the solution of the $\mathcal{P}_2$ problem is 28% higher than the minimum achieved by solving the specialized $\mathcal{P}_\infty$ problem. From the point of view of outlier resilience, looking at the e_1 criterion values on the cleaned point sets and comparing the figures obtained from the $\mathcal{P}_2$ and $\mathcal{P}_1$ problems, we remark a significant improvement in the fitting error. Notice also that solutions in the iterative algorithms are always obtained in less than five iterations, with running times that grow gracefully with the problem dimension.

TABLE I
MATCHING ERRORS ATTAINED ON FIVE RANDOMLY GENERATED SETS OF POINTS OF INCREASING CARDINALITY BY THE OPTIMAL SOLUTIONS OF PROBLEMS $\mathcal{P}_2, \mathcal{P}_\infty, \mathcal{P}_1$ (DATA AS IN EXAMPLE 2)

	$n = 5$ ($n_{\text{outl}} = 1$)	$n = 10$ ($n_{\text{outl}} = 1$)	$n = 50$ ($n_{\text{outl}} = 5$)	$n = 100$ ($n_{\text{outl}} = 11$)	$n = 1000$ ($n_{\text{outl}} = 89$)
$\mathcal{P}_2$	$e_2 = 5.3359$ $e_\infty = 10.5016$ $e_1 = 4.2741$ $e_2 = 2.8317$ no outl. $e_\infty = 3.9582$ no outl. $e_1 = 2.7172$ no outl.	$e_2 = 1.992$ $e_\infty = 5.395$ $e_1 = 1.4623$ $e_2 = 1.0841$ no outl. $e_\infty = 1.5733$ no outl. $e_1 = 1.0254$ no outl.	$e_2 = 2.9548$ $e_\infty = 10.6117$ $e_1 = 1.8127$ $e_2 = 1.2225$ no outl. $e_\infty = 2.026$ no outl. $e_1 = 1.1416$ no outl.	$e_2 = 3.7316$ $e_\infty = 13.5229$ $e_1 = 2.0928$ $e_2 = 1.0847$ no outl. $e_\infty = 1.8919$ no outl. $e_1 = 1.0293$ no outl.	$e_2 = 3.1436$ $e_\infty = 16.1937$ $e_1 = 1.7415$ $e_2 = 1.0151$ no outl. $e_\infty = 1.6382$ no outl. $e_1 = 0.97251$ no outl.
$\mathcal{P}_\infty$	iters $= 4$ time $= 0.43$ s $e_2 = 6.501$ $e_\infty = 6.9171$ $e_1 = 6.4623$ $e_2 = 6.3928$ no outl. $e_\infty = 6.9171$ no outl. $e_1 = 6.3486$ no outl.	iters $= 4$ time $= 0.50$ s $e_2 = 2.6487$ $e_\infty = 3.3009$ $e_1 = 2.5085$ $e_2 = 2.5661$ no outl. $e_\infty = 3.3009$ no outl. $e_1 = 2.4205$ no outl.	iters $= 5$ time $= 1.14$ s $e_2 = 6.3623$ $e_\infty = 7.5891$ $e_1 = 6.2972$ $e_2 = 6.3609$ no outl. $e_\infty = 7.5891$ no outl. $e_1 = 6.3134$ no outl.	iters $= 5$ time $= 1.72$ s $e_2 = 7.7376$ $e_\infty = 11.8345$ $e_1 = 7.2016$ $e_2 = 7.359$ no outl. $e_\infty = 11.8345$ no outl. $e_1 = 6.8319$ no outl.	iters $= 4$ time $= 11.74$ s $e_2 = 3.5327$ $e_\infty = 14.2295$ $e_1 = 2.4679$ $e_2 = 1.8958$ no outl. $e_\infty = 3.6108$ no outl. $e_1 = 1.7546$ no outl.
$\mathcal{P}_1$	iters $= 4$ time $= 0.87$ s $e_2 = 5.9095$ $e_\infty = 13.1199$ $e_1 = 3.1669$ $e_2 = 0.78723$ no outl. $e_\infty = 1.0147$ no outl. $e_1 = 0.6787$ no outl.	iters $= 3$ time $= 0.83$ s $e_2 = 2.1837$ $e_\infty = 6.5901$ $e_1 = 1.227$ $e_2 = 0.6876$ no outl. $e_\infty = 1.0053$ no outl. $e_1 = 0.63113$ no outl.	iters $= 3$ time $= 1.70$ s $e_2 = 3.0207$ $e_\infty = 11.0723$ $e_1 = 1.6955$ $e_2 = 1.0143$ no outl. $e_\infty = 1.6878$ no outl. $e_1 = 0.96726$ no outl.	iters $= 3$ time $= 2.58$ s $e_2 = 3.7518$ $e_\infty = 13.7081$ $e_1 = 2.0371$ $e_2 = 0.99446$ no outl. $e_\infty = 1.5617$ no outl. $e_1 = 0.94955$ no outl.	iters $= 3$ time $= 20.02$ s $e_2 = 3.1461$ $e_\infty = 16.2055$ $e_1 = 1.7359$ $e_2 = 1.0059$ no outl. $e_\infty = 1.6206$ no outl. $e_1 = 0.96476$ no outl.

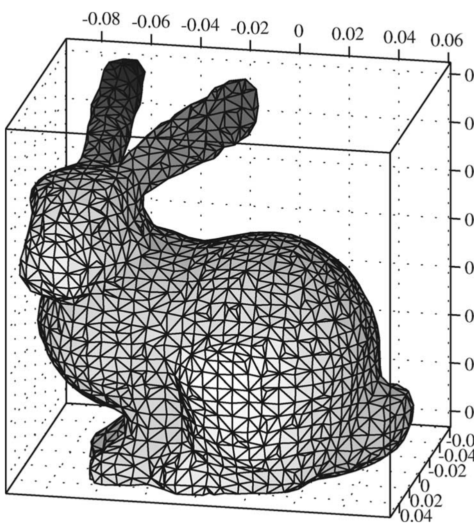

Fig. 3. "Stanford bunny" 3-D model.

C. Example 3

In the final example, we considered as template points $n = 453$ vertices of a 3-D model known as the "Stanford bunny," available at http://graphics.stanford.edu/software/scanview/models/bunny.html (see Fig. 3).

The point set B is constructed by adding to each point of A a random perturbation which, with a probability of 0.9, is extracted from a zero-mean normal with covariance $10^{-6}I_3$, and with a probability of 0.1 from a zero-mean normal with covariance $10^{-4}I_3$. Then, the perturbed points are rotated by 45° around the $[1\ 1\ 1]^\top$ axis and translated by $[1\ 1\ 1]^\top$.

The SSE, MAE, and SAE optimal displacement problems are then solved for the (A, B) pair, setting the algorithm parameters to $\gamma = 0.0524$, $\eta = 10^{-5}$. The results are reported in Table II. We notice that both $\mathcal{P}_2$ and $\mathcal{P}_1$ yield an ℓ_∞ error about 23% higher than the one achieved by the $\mathcal{P}_\infty$ problem. We also remark that the cost values attained by the $\mathcal{P}_2$ and the $\mathcal{P}_1$ problems tend to become numerically similar, as the number n of data point increases. This effect is mainly due to the fact that variations in the residuals in the e_2 or e_1 costs are weighted by

TABLE II
MATCHING ERRORS ATTAINED BY THE OPTIMAL SOLUTIONS OF PROBLEMS $\mathcal{P}_2, \mathcal{P}_\infty, \mathcal{P}_1$ ON THE "STANFORD BUNNY" EXAMPLE

$\mathcal{P}_2$	$\mathcal{P}_\infty$	$\mathcal{P}_1$
	iters $= 5$	iters $= 3$
	time $= 9.1866$ s	time $= 12.964$ s
$e_2 = 0.0062237$	$e_2 = 0.0104$	$e_2 = 0.0062324$
$e_\infty = 0.036445$	$e_\infty = 0.02953$	$e_\infty = 0.036353$
$e_1 = 0.0033589$	$e_1 = 0.0094834$	$e_1 = 0.0033334$

a $1/\sqrt{n}$ or $1/n$ factor, respectively. Therefore, changes in the optimal displacement parameters may result in small variations of the cost when n is large.

VII. CONCLUSION

The classical cost criterion based on the sum of squared distances between matching points might be unsuitable in problems where the data are affected by outliers or nonnormal noise or where the actual maximum matching error across all the points needs to be kept under control. The SAE and MAE criteria, respectively, are preferable matching measures in these cases.

In this paper, we showed that suboptimal solutions for the point matching problems under the MAE and the SAE cost criteria can be obtained by exploiting an iterative "convexification" technique that requires a solution at each iteration of an SOCP. Such problems can be solved efficiently in polynomial time by means of interior-point methods for convex programming. The overall numerical cost, although admittedly higher than that of an SVD factorization (which is required in the classical approach based on the sum-of-squared-distances cost criterion), appears to be acceptable for those applications that can tolerate computation times on the order of a few seconds.

The technique discussed in this paper is currently limited to the case where point correspondences are given *a priori*. An important open problem that we leave for future research involves a generalization of the method to tackle the problem of simultaneous point labeling and displacement optimization under the considered nonstandard cost criteria.

APPENDIX

Proof of Theorem 1: Notice first that the optimal arguments of problem (1) remain the same if we minimize the squared objective $e_2^2(R,t)$, instead of $e_2(R,t)$. Thus, we solve $\min_{R\in \mathrm{so}(3), t\in\mathbb{R}^3} e_2^2(R,t)$, which is rewritten in matrix form as

$$\begin{aligned} e_{2,\mathrm{opt}}^2 &= \frac{1}{n} \min_{R\in\mathrm{so}(3), t\in\mathbb{R}^3} \|A - RB - tu^\top\|_F^2 \\ &= \frac{1}{n} \min_{R\in\mathrm{so}(3)} \min_{t\in\mathbb{R}^3} \|A - RB - tu^\top\|_F^2. \end{aligned} \tag{10}$$

For any fixed R, the inner minimization problem in (10) is written as

$$\begin{aligned} &\min_{t\in\mathbb{R}^3} \|A - RB - tu^\top\|_F^2 \\ &\quad = \min_{t\in\mathbb{R}^3} \left\{ \|A - RB\|_F^2 - 2\mathrm{Tr}(A - RB)ut^\top + nt^\top t \right\}. \end{aligned}$$

This minimum is easily computed by setting the gradient with respect to t to zero

$$\nabla_t = -2u^\top (A - RB)^\top + 2nt^\top = 0$$

which yields the optimal translation vector as a function of R

$$t = \frac{1}{n}(A - RB)u.$$

Substituting t back into (10), with the notation in (2), we obtain

$$ne_{2,\mathrm{opt}}^2 = \min_{R\in\mathrm{so}(3)} \|\tilde{A} - R\tilde{B}\|_F^2. \tag{11}$$

An optimal solution to (11) can be determined as follows. Recall that for orthogonal R, it holds that $\|R\tilde{B}\|_F = \|\tilde{B}\|_F$, and let $U\Sigma V^\top$ be the singular value factorization of $\tilde{B}\tilde{A}^\top$. Then, we write

$$\begin{aligned} \|\tilde{A} - R\tilde{B}\|_F^2 &= \|\tilde{A}\|_F^2 + \|R\tilde{B}\|_F^2 - 2\mathrm{Tr}\, R\tilde{B}\tilde{A}^\top \\ &= \|\tilde{A}\|_F^2 + \|\tilde{B}\|_F^2 - 2\mathrm{Tr}\, T\Sigma \\ &= \|\tilde{A}\|_F^2 + \|\tilde{B}\|_F^2 - 2\sum_{i=1}^3 T_{ii}\sigma_i \end{aligned} \tag{12}$$

where $T \doteq V^\top RU$ is an orthogonal matrix. Clearly, (12) is minimized if $\sum_{i=1}^3 T_{ii}\sigma_i$ is maximized. Suppose first that $\det \tilde{B}\tilde{A}^\top \geq 0$ (or that $\det U \det V = +1$), then—since orthogonality of T imposes $|T_{ii}| \leq 1$—the maximum is achieved by choosing $T_{11} = T_{22} = T_{33} = 1$, i.e., $T = I_3$, which results in a rotation matrix $R = VU^\top$.

Notice that, in the "degenerate" case when $\det U \det V = -1$, the previous choice would make R a reflection, instead of a rotation matrix. To correct this issue, we choose in this case $T_{11} = 1, T_{22} = 1, T_{33} = -1$, which yields the optimal rotation $R = V\mathrm{diag}(1,1,-1)U^\top$ (see [20] for further details on the degenerate case). ■

REFERENCES

[1] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D points sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.

[2] Y. Q. Bai, G. Q. Wang, and C. Roos. (2004). "A new primal-dual interior-point algorithm for second-order cone optimization," *Optimization Online*. [Online]. Available: www.optimization-online.org/DB_HTML/2004/11/1001.html

[3] G. Calafiore and B. Bona, "Constrained optimal fitting of three-dimensional vector patterns," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 838–844, Oct. 1998.

[4] G. Calafiore, F. Dabbene, and R. Tempo, "Radial and uniform distributions in vector and matrix spaces for probabilistic robustness," in *Topics in Control and Its Applications*, D. E. Miller and L. Qiu, Eds. New York: Springer-Verlag, 1999, pp. 17–31.

[5] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Proc. Int. Conf. Pattern Recog.*, Québec City, QC, Canada, 2002, pp. 545–548.

[6] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometric matching between primitive surfaces," in *Proc. Int. Joint Conf. Artif. Intell.*, Karlshrue, Germany, 1983, vol. 8, pp. 996–1002.

[7] J. M. Fitzpatrick, J. B. West, and C. R. Maurer, "Predicting error in rigid-body point-based registration," *IEEE Trans. Med. Imag.*, vol. 17, no. 5, pp. 694–702, Oct. 1999.

[8] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[9] Y. L. Gu, "An exploration of orientation representation by Lie algebra for robotic applications," *IEEE Trans. Syst., Man Cybern.*, vol. 20, no. 1, pp. 243–248, Jan./Feb. 1990.

[10] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.

[11] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[12] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[13] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Lin. Alg. Appl.*, vol. 284, no. 1, pp. 193–228, Nov. 1998.

[14] J. Löfberg. (2004). "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, Taipei, Taiwan, R.O.C. pp. 284–289. [Online]. Available: control.ee.ethz.ch/~joloef/yalmip.php

[15] R. J. Micheals and T. E. Boult. (2000). *On the Robustness of Absolute Orientation*. [Online]. Available: citeseer.ifi.unizh.ch/533357.html

[16] C. F. Olson, "Maximum-likelihood image matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 853–857, Jun. 2002.

[17] N. Sebe and M. S. Lew, "Robust shape matching," in *Proc. Int. CIVR*, London, U.K., Jul. 2002, pp. 17–28.

[18] N. Sebe, M. S. Lew, and D. P. Huijsmans, "Toward improved ranking metrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1132–1141, Oct. 2000.

[19] J. P. P. Starink and E. Backer, "Finding point correspondences using simulated annealing," *Pattern Recognit.*, vol. 28, no. 2, pp. 231–240, Feb. 1995.

[20] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.

[21] R. C. Veltkamp, "Shape matching: similarity measures and algorithms," in *Proc. Int. Conf. SMI*, Genova, Italy, 2001, pp. 188–197.

Giuseppe C. Calafiore received the Laurea degree in electrical engineering and the Doctorate degree in information and system theory from the Politecnico di Torino, Torino, Italy, in 1993 and 1997, respectively.

He held visiting positions with the Information Systems Laboratory, Stanford University, Stanford, CA, in 1995, with the Ecole Nationale Supérieure de Techniques Avenceés, Paris, France, in 1998, and with the University of California, Berkeley, in 1999, 2003 and 2007. He is currently an Associate Professor with the Dipartimento di Automatica e Informatica, Politecnico di Torino. His research interests are in the field of analysis and control of uncertain systems, convex optimization, probabilistic algorithms, and risk management. He is an author or coauthor of six books and over 100 publications on international journals and conferences.

Dr. Calafiore is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.