Numer Algor (2017) 74:755–779
DOI 10.1007/s11075-016-0170-3

ORIGINAL PAPER

Robust registration of surfaces using a refined iterative closest point algorithm with a trust region approach

Per Bergström[1] · **Ove Edlund**[1]

Received: 8 September 2015 / Accepted: 22 June 2016 / Published online: 7 July 2016

Abstract The problem of finding a rigid body transformation, which aligns a set of data points with a given surface, using a robust M-estimation technique is considered. A refined iterative closest point (ICP) algorithm is described where a minimization problem of point-to-plane distances with a proposed constraint is solved in each iteration to find an updating transformation. The constraint is derived from a sum of weighted squared point-to-point distances and forms a natural trust region, which ensures convergence. Only a minor number of additional computations are required to use it. Two alternative trust regions are introduced and analyzed. Finally, numerical results for some test problems are presented. It is obvious from these results that there is a significant advantage, with respect to convergence rate of accuracy, to use the proposed trust region approach in comparison with using point-to-point distance minimization as well as using point-to-plane distance minimization and a Newton-type update without any step size control.

Keywords ICP · M-estimation · Registration · Robust · Surface · Trust region

Mathematics Subject Classification (2010) 65D19 · 65D18 · 49M15

✉ Per Bergström
per.bergstrom@ltu.se

Ove Edlund
ove.edlund@ltu.se

[1] Division of Mathematical Sciences, Department of Engineering Sciences and Mathematics, Luleå University of Technology, SE-971 87 Luleå, Sweden

1 Introduction

1.1 Survey of surface registration

Registration of surfaces that we consider is the problem of finding a rigid body transformation, consisting of a rotation and translation, such that a set of data points fits to a given surface under the transformation. Shape inspection is a common application where a surface registration problem of this kind needs to be solved, see, e.g., [8, 9, 29, 53]. A set of data points representing a measured shape of an arbitrarily oriented object is to be aligned with a computer-aided design (CAD) model, which defines the nominal shape of the measured object. This makes it possible to identify deviations between the measured shape and the nominal shape. Examples of shape measurement methods for this kind of applications are found in, e.g., [11, 28, 31, 42, 43, 50]. The nominal shape defined by the CAD model is geometrically a composite surface made up of finite simple surface-patches.

In order to find a best fit between a set of data points and a given surface, an initial alignment is required to approximately establish a correspondence. Once an initial alignment is determined, a refined alignment can be found using a local registration technique. For determining the initial alignment, a global registration technique is required to avoid ending up in a local minimum different from the global one. Global registration techniques are typically using evaluations of various trial alignments. Methods for reducing the number of trial alignments are discussed in, e.g., [3, 14, 18, 35]. However, we will here focus on local registration.

An algorithm that can be used to solve the local surface registration problem is the iterative closest point (ICP) algorithm [7, 12, 41]. In general, there are two common types of ICP-algorithms. These are based on point-to-point distance minimization [7] and point-to-plane distance minimization [12]. The planes are the tangent planes at the closest surface points to the data points.

The convergence for some different registration algorithms are discussed in [37]. Minimization of point-to-point distances gives an algorithm that converges to a local minimum, and the sequence of values of the objective function over the iterations is monotonically decreasing. Unfortunately, the asymptotic convergence is linear and in general very slow. Minimization of point-to-plane distances using a Newton-type update allows flat regions to slide along each other and gives quadratic asymptotic convergence. Unfortunately, monotone convergence cannot be ensured without any step size control procedure, where a number of evaluations of the objective function are performed to find an appropriate step. If evaluations of the objective function are computationally expensive, which is the case in surface registration problems, step size control procedures will then also be expensive.

1.2 Robust estimation

Shape measurements using optical full-field methods from a single set of synchronously captured images with short exposure are prone to have measurement errors. The advantage of using these kinds of shape measurement methods is that the measurement can be done very quickly, which enables shape measurements of moving objects. The disadvantage is that errors will be much more prominent. In addition, when using optical full-field shape measurement methods the background will be measured as well. Different filters can be used in order to more or less remove the background from the representation of the measured shape. However, it cannot be expected that used filters will remove all background. When processing data of a measured shape from an optical full-field method, it is essential to use a robust method, as considered in, e.g., [36, 41, 48, 54]. Least squares estimates are highly sensitive to gross errors. The registration runs the risk of being spoiled by aberrant and incorrect data if robust methods are not used.

Weighting of point-pairs is a robust registration method discussed in [19, 41]. The method is based on assigning different weights to different point-pairs for solving a weighted least squares problem. Lower weights are assigned to point-pairs that seem to be outliers giving them less influence on the estimation. The weights are typically defined from the distances between the points or some quality values of the data points obtained from their source.

Rejection of point-pairs is a robust registration method, which is also discussed in [41]. The rejection method is similar to weighting of point-pairs but the weights are either one or zero. The rejected point-pairs can be point-pairs with an inner distance greater than a given distance apart [49] or the n % of point-pairs with greatest distance [38]. In the first case, a threshold distance must be determined and in the second case an assumption about the amount of outliers must be done. Rejection of point-pairs with a surface point on the boundary to the model, see [49], gives a geometry dependent rejection method. When using a rejection method, "bad" data might be used as "good" data and "good" data might be rejected as "bad" data.

Sparse ICP, see [10, 34], is yet another variant of a robust registration method worth to be mentioned. It uses a sparsity inducing ℓ_p-norm formulation of the registration problem, where $p \in [0, 1]$. It is adapted for establish correspondences between point sets representing partially overlapping scans with various degrees of noise and outliers, and solving for the optimal transformation that brings these correspondences into alignment.

A method using the Levenberg-Marquardt algorithm, LM-ICP, is presented in [15]. It is applied directly on the squared sum of shortest distances, so the Levenberg-Marquardt step is used for the update of the rigid body transformation. What is required though is the partial derivatives of the residuals for all data points, which either must be approximated at considerable computational cost, or estimated from a precomputed customized data structure. A good point of the LM-ICP algorithm is that it makes it is easy to introduce robust estimators.

M-estimation, which originates from robust statistics [23, 26, 40], is a technique commonly used in robust regression but is also used in robust registration [27, 30, 47]. It is a generalization of least squares minimization where the squares are replaced with a robust criterion function giving less influence of strongly deviating data. In that way, it is very similar to the weighting of the point-pairs method but it gives better control of the estimation since an objective function to be minimized is defined.

1.3 Trust region methods

Trust region is a term used in mathematical optimization to denote a subset of the domain of an objective function, on which the objective function is approximated by a model function, see, e.g., [2, 13, 17, 20, 45, 51]. Commonly, the model function is a second-order approximation of the objective function. The main idea of trust region methods originates from a modification of Newton's method with a line search procedure. Unlike line search procedures, where a search is carried out along a given direction in the domain, trust region methods obtain the new iterate point by searching in the trust region, which is normally a neighborhood of the current iterate point.

In trust region methods, a trial step is to be found where a maximal step length is given. It is found by minimizing the model function that approximates the objective function near the current iteration point so the step length restriction is satisfied. That gives a trial step and by comparing the value of the objective function with its approximation at that point, it is determined if the computed step can be accepted or if a new trial step has to be found by changing the maximal step length.

The model function is computed at the current iteration point so it is assumed that it approximates the objective function quite well near that point. Far away from the current iteration point, there is no guarantee that the approximation is good enough and that is why the trial step has to be judged so that convergence is ensured. The problem of finding an appropriate step length is discussed in, e.g., [1, 25, 39, 52]. It is desired to determine an appropriate step length with as few evaluations of the objective function as possible.

Applied to optimization problems, trust region methods perform often better than Newton's method with a line search procedure. The reason is because trust region methods takes full advantage of the approximating model function, whereas Newton's method with line search utilizes only the minimizer of a second-order approximation and a directional derivative. Trust region methods do not require a positive definite Hessian, which is an essential property to achieve fast convergence using Newton's method. However, if the number of decision variables is large, resolving the trust region problem can be costly, but this is not an issue in our approach.

1.4 Proposed approach

A refined approach for updating the rigid body transformation, which combines the advantages of the different convergence behaviors for both the point-to-point distance minimization and point-to-plane distance minimization ICP-algorithms using a robust criterion, is proposed. The approach is a trust region method combined with an M-estimation technique. We are describing how a better update of the transformation can be obtained than using point-to-point distance minimization but still have the value of the objective function under control and ensure convergence.

The proposed trust region approach is to constraining the updating transformation in its parameter space over a well-defined region, which is derived from an expression of a sum of weighted squared point-to-point distances. A second-order approximation to a function of a sum of point-to-plane distances is minimized over the trust region for obtaining the updating transformation. A distinct difference with most of the existing trust region methods, see, e.g., [13, 51], is that the proposed trust region is defined from a function of the step instead of a norm of the step. The minimizer of the sum of weighted squared point-to-point distances can be considered as a "center point" of the proposed trust region. Convergence of the objective function is guaranteed without any additional function evaluations. Therefore, the sum of weighted squared point-to-point distances forms a natural basis for defining a trust region in registration problems.

Two different ways of bounding the proposed trust region are discussed, and that is to use only information from the current iteration, and to use information from both the current and the previous iteration. The significant difference with the methods discussed in [1, 25, 39, 52] is that no additional evaluations of the objective function is performed in order to determine the size of the trust region. Only a minor number of additional computations are required to use it.

The robust surface registration method presented in this paper is a development of the method presented in [5], where robust registration of arbitrary point sets is considered without any requirement of availability of surface normals. The new approach originates from a similar but substantially improved idea where more information about the surface is used. From an optimization point of view, much is different since in this paper we are solving a constrained point-to-plane distance minimization problem to obtain the updating transformation, whereas in [5] a weighted least squares problem is solved.

2 Robust registration of surfaces

2.1 Problem

The considered registration problem is to find a rigid body transformation, consisting of a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$, i.e. $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = +1$, and a translation vector $\mathbf{t} \in \mathbb{R}^3$, such that a set of N data points, $\{\mathbf{p}_i\}_{i=1}^{N}$, fits to a set of surface points, S, under the transformation. An M-estimate of the rigid body transformation using a robust criterion function $\varrho : \mathbb{R} \to [0, \infty)$ is obtained by solving

$$\min_{\mathbf{R},\mathbf{t}} f(\mathbf{R}, \mathbf{t}) , \tag{1}$$

where

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \varrho(\mathrm{d}(\mathbf{R}\mathbf{p}_i + \mathbf{t}, S)) , \tag{2}$$

and d is a distance operator defined by

$$\mathrm{d}(\mathbf{p}, S) = \min_{\mathbf{y}\in S} \|\mathbf{p} - \mathbf{y}\|_2 .$$

An essential property of a robust criterion function ϱ is that it reduces undesired influence of data with gross errors on the estimation. We make a formal definition of a set of criterion functions $\mathcal{Q}$ that we are using. The derivative of ϱ is denoted by

ψ, i.e., $\psi(r) = \varrho'(r)$, which is the standard notation in robust statistics. A criterion function ϱ belongs to the set $\mathcal{Q}$ if and only if all of the following conditions are fulfilled:

1. $\varrho(r)$ is an even function and $\boldsymbol{C}^1$-continuous on $\mathbb{R}$, and $\varrho(0) = 0$.
2. $\varrho(r)$ is monotonically increasing on $[0, \infty)$.
3. $\psi(r)/r$ is monotonically decreasing and bounded above on $(0, \infty)$.

Since the function ϱ is even and continuously differentiable, its derivative at zero satisfies $\psi(0) = 0$ and the second derivative $\varrho''(0) = \psi'(0)$ exists for all $\varrho \in \mathcal{Q}$. Criterion functions in $\mathcal{Q}$ are monotonically increasing on $[0, \infty)$ so that $\psi(r) \geq 0$ for all $r \geq 0$.

A weight function w is defined as

$$w(r) = \begin{cases} \dfrac{\psi(r)}{r} & \text{if } r \neq 0\,, \\ \lim\limits_{r \to 0} \dfrac{\psi(r)}{r} = \psi'(0) & \text{if } r = 0\,. \end{cases} \tag{3}$$

From the definition of $\mathcal{Q}$, it follows that the weight function w associated to $\varrho \in \mathcal{Q}$ is an even and non-negative function, which is bounded above and monotonically decreasing on $[0, \infty)$. A property of the function w is that $w(r)r = \psi(r)$ holds for all $r \in \mathbb{R}$.

Cauchy's function ϱ_{Ca} and Welsch's function ϱ_{We} are examples of robust criterion functions in $\mathcal{Q}$. These functions, their derivatives, and corresponding weight functions are as follows:

- Cauchy's function

$$\varrho_{\mathrm{Ca}}(r) = \tfrac{\kappa_{\mathrm{Ca}}^2}{2} \ln\left(1 + \left(\tfrac{r}{\kappa_{\mathrm{Ca}}}\right)^2\right)$$

$$\psi_{\mathrm{Ca}}(r) = \frac{r}{1+\left(\frac{r}{\kappa_{\mathrm{Ca}}}\right)^2}$$

$$w_{\mathrm{Ca}}(r) = \frac{1}{1+\left(\frac{r}{\kappa_{\mathrm{Ca}}}\right)^2}$$

- Welsch's function

$$\varrho_{\mathrm{We}}(r) = \tfrac{\kappa_{\mathrm{We}}^2}{2} \left(1 - \exp\left(-\tfrac{r^2}{\kappa_{\mathrm{We}}^2}\right)\right)$$

$$\psi_{\mathrm{We}}(r) = r \exp\left(-\tfrac{r^2}{\kappa_{\mathrm{We}}^2}\right)$$

$$w_{\mathrm{We}}(r) = \exp\left(-\tfrac{r^2}{\kappa_{\mathrm{We}}^2}\right)$$

The graphs of these criterion functions are shown in Fig. 1.

The two criterion functions have different robust properties in parameter estimation with respect to handling data of occasional gross errors. Welsch's function has the strongest protection of high influence caused by this kind of undesired data, which

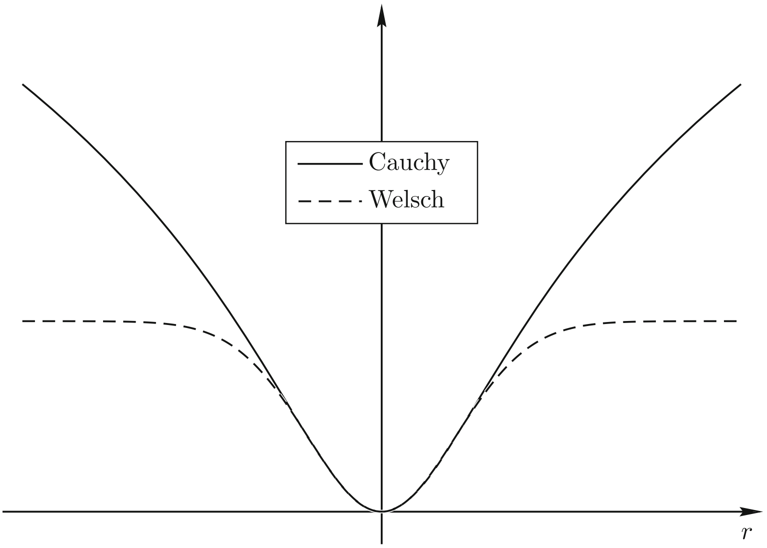

Fig. 1 Two robust criterion functions, Cauchy's function and Welsch's function.

is almost excluded in the estimation. However, the protection against influence from outliers using Cauchy's criterion function is also strong.

To make the robust criterion functions comparable, the parameter κ is chosen in such a way that the asymptotic variance

$$V(\psi, F) = \frac{\int \psi^2 dF}{\left(\int \psi' dF\right)^2},$$

(see, e.g., [22, 26, 32]) is close to one, given an assumption of normal distribution $\mathcal{N}(0, 1)$ for the residuals. Here, F is the probability density function of the normal distribution $\mathcal{N}(0, 1)$. The asymptotic variance $V(\psi, F)$ is always greater than one for robust criterion functions. By choosing it to be 1.01, we get the following values of the parameters:

$$\begin{aligned} \kappa_{\text{Ca}} &= 4.3040\,, \\ \kappa_{\text{We}} &= 4.7536\,. \end{aligned} \tag{4}$$

A scaling of the residuals is required to conform with the scale of the used criterion function. That is to enable an accurate robust estimation.

2.2 Algorithm

Problem (1) using criterion functions in $\mathcal{Q}$ can be solved with the ICP-algorithm, which is presented by Algorithm 1. It operates in two principal stages in each iteration and that is to find the closest surface points to a set of data points, and to estimate a rigid body transformation. The problem of finding the closest surface points is

Algorithm 1 An ICP-algorithm using a robust criterion

```
Require: Data points $\{\mathbf{p}_i\}_{i=1}^N$, surface points $S$ and surface normals, $\varrho$, $\psi$, $\psi'$, $w$
Require: $\mathbf{R}^{(0)} = \mathbf{I}$, $\mathbf{t}^{(0)} = \mathbf{0}$ (or a more appropriate initial rigid body transformation)
  /* Operations are performed for $i = 1, \ldots, N$ */
  $k = 0,\ \mathbf{p}_i^{(0)} = \mathbf{R}^{(0)}\mathbf{p}_i + \mathbf{t}^{(0)}$
  repeat
    $k = k + 1$
    $\mathbf{y}_i^{(k-1)} = \mathcal{C}(\mathbf{p}_i^{(k-1)}, S)$
    $w_i = w\big(\|\mathbf{p}_i^{(k-1)} - \mathbf{y}_i^{(k-1)}\|_2\big)$
  if any $w_i > 0$ then
    Find an appropriate update $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$ of the rigid body transformation
    $\mathbf{p}_i^{(k)} = \mathbf{R}^\dagger \mathbf{p}_i^{(k-1)} + \mathbf{t}^\dagger$
    $\mathbf{R}^{(k)} = \mathbf{R}^\dagger \mathbf{R}^{(k-1)}$
    $\mathbf{t}^{(k)} = \mathbf{R}^\dagger \mathbf{t}^{(k-1)} + \mathbf{t}^\dagger$
  else /* all $w_i = 0$ */
    $\mathbf{p}_i^{(k)} = \mathbf{p}_i^{(k-1)}$
    $\mathbf{R}^{(k)} = \mathbf{R}^{(k-1)}$
    $\mathbf{t}^{(k)} = \mathbf{t}^{(k-1)}$
    Terminate iterations
  end if
until convergence
return $\mathbf{R}^{(k)}$, $\mathbf{t}^{(k)}$, $\{\mathbf{p}_i^{(k)}\}_{i=1}^N$
```

considered in, e.g., [6, 21, 46], and the estimation of the updating rigid body transformation $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$ is the main content in this paper. An initial rigid body transformation, $(\mathbf{R}^{(0)}, \mathbf{t}^{(0)})$, is required. In the testing phase, we use $\mathbf{R}^{(0)} = \mathbf{I}$ and $\mathbf{t}^{(0)} = \mathbf{0}$, but if there is any prior knowledge that can be utilized, the algorithm benefits greatly from it. In Algorithm 1, we use the notation $\mathcal{C}$ to denote a closest point operator. That is $\mathbf{y}_i^{(k-1)} = \mathcal{C}(\mathbf{p}_i^{(k-1)}, S)$ is the closest surface point in S to a transformed data point, $\mathbf{p}_i^{(k-1)} = \mathbf{R}^{(k-1)}\mathbf{p}_i + \mathbf{t}^{(k-1)}$. If the closest surface point is not unique, $\mathbf{y}_i^{(k-1)}$ can be arbitrarily chosen from the set of closest surface points. The choice of termination criterion, that is the determination when the solution is assumed to have converged, can be done in many different ways. It depends on what is considered as a sufficiently good solution. In our numerical experiments, see Section 5, we terminate the iterations dependent of the transformation update $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$. When it is close enough to the identity transformation $(\mathbf{I}, \mathbf{0})$, the current transformation $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ and the transformed data points are returned.

For a given criterion function ϱ and a positive scale parameter σ_*, let $\tilde{\varrho}(r) = \varrho(r/\sigma_*)$ be a re-scaled criterion function. If the criterion function ϱ is in the set $\mathcal{Q}$, then $\tilde{\varrho}$ will also be in the set $\mathcal{Q}$, and therefore, the problem

$$\min_{\mathbf{R},\mathbf{t}} \tilde{f}(\mathbf{R}, \mathbf{t}) \, , \tag{5}$$

Algorithm 2 Extension of Algorithm 1 using re-scaled residuals

```
repeat
    ...
    Estimate $\sigma_{k-1}$
    $w_i = w\big(\|\mathbf{p}_i^{(k-1)} - \mathbf{y}_i^{(k-1)}\|_2/\sigma_{k-1}\big)$
    ...
until convergence
```

where

$$\tilde{f}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \varrho\left(\frac{\mathrm{d}(\mathbf{R}\mathbf{p}_i + \mathbf{t}, S)}{\sigma_*}\right), \tag{6}$$

can be solved using the ICP-algorithm given by Algorithm 1.

If the value of σ_* is small and ψ is close or equal to zero for large residuals, the criterion function $\tilde{\varrho}$ will have a narrow descend at zero. It might result in a slow convergence or convergence to a stationary point far from the global minimum, which could even be a maximum. The remedy of this problem is to use a quite large value of a scaling parameter σ_{k-1}, greater than σ_*, in the first iterations and then gradually decrease its value to the desired value σ_*. An extended version of Algorithm 1 is given by Algorithm 2, where a re-scaling of the residuals is used. In Algorithm 2, most of the steps are the same as in Algorithm 1 and are therefore removed to make it shorter.

In [5], it is shown that the values $\tilde{f}_k$ of the objective function $\tilde{f}$ in (6) from iteration k of Algorithm 2 satisfy

$$\sum_{i=1}^{N} \varrho\left(\frac{r_i}{\sigma_k}\right) \le \tilde{f}_k \le \frac{\sigma_k^2}{\sigma_*^2} \sum_{i=1}^{N} \varrho\left(\frac{r_i}{\sigma_k}\right), \tag{7}$$

if $\sigma_k \ge \sigma_* > 0$ and $\varrho \in \mathcal{Q}$. When using re-scaled residuals, we cannot expect that Algorithm 2 will have the same convergence properties as Algorithm 1, but if the sequence of σ_k approaches to σ_* Algorithm 2 will eventually behave like Algorithm 1, which follows from (7).

After a scaling using an appropriate value of σ_k, the residuals of the "good" data, which is the data that represents the surface quite well, that are entered into the criterion function can be assumed to have approximately the same distribution in each iteration k. For simplicity, we will not use the scaling parameter σ_k in the following text until the numerical results section.

2.3 Parametrization

A rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ can be parameterized as

$$\mathbf{R}(\mathbf{z}) = \begin{bmatrix} \mathrm{c}_2\mathrm{c}_3 & \mathrm{s}_1\mathrm{s}_2\mathrm{c}_3 - \mathrm{c}_1\mathrm{s}_3 & \mathrm{c}_1\mathrm{s}_2\mathrm{c}_3 + \mathrm{s}_1\mathrm{s}_3 \\ \mathrm{c}_2\mathrm{s}_3 & \mathrm{s}_1\mathrm{s}_2\mathrm{s}_3 + \mathrm{c}_1\mathrm{c}_3 & \mathrm{c}_1\mathrm{s}_2\mathrm{s}_3 - \mathrm{s}_1\mathrm{c}_3 \\ -\mathrm{s}_2 & \mathrm{s}_1\mathrm{c}_2 & \mathrm{c}_1\mathrm{c}_2 \end{bmatrix}, \tag{8}$$

where $\mathbf{z} = [z_1, z_2, z_3, z_4, z_5, z_6]^{\mathrm{T}}$, and s and c with index 1,2,3 are abbreviations for sin and cos of angles z_1, z_2, and z_3, respectively. This parametrization of $\mathbf{R}$ rotates points in $\mathbb{R}^3$ through the angle z_1 about the x-axis, the angle z_2 about the y-axis, and the angle z_3 about the z-axis, in that order. All rotation matrices in $\mathbb{R}^{3\times 3}$ can be represented by $-\pi < z_1, z_3 \leq \pi$, and $-\pi/2 \leq z_2 \leq \pi/2$. A translation vector $\mathbf{t} \in \mathbb{R}^3$ can be parameterized as

$$\mathbf{t}(\mathbf{z}) = [z_4, z_5, z_6]^{\mathrm{T}}, \tag{9}$$

where z_4, z_5, and z_6 are the translations in x, y, and z directions. Therefore, an arbitrary rigid body transformation can be parameterized by $\mathbf{z} \in \mathbb{R}^6$, where $-\pi < z_1, z_3 \leq \pi$ and $-\pi/2 \leq z_2 \leq \pi/2$.

Let φ be an operator for obtaining the rigid body transformation $(\mathbf{R}, \mathbf{t})$ from a given point $\mathbf{z}$, which is written as $[\mathbf{R}, \mathbf{t}] = \varphi(\mathbf{z})$. We also introduce an operator ϑ for obtaining the point $\mathbf{z}$ from a given rigid body transformation $(\mathbf{R}, \mathbf{t})$ such that

$$\mathbf{z} = \vartheta(\mathbf{R}, \mathbf{t}) \;\Leftrightarrow\; [\mathbf{R}, \mathbf{t}] = \varphi(\mathbf{z})\,, \tag{10}$$

if $-\pi < z_1, z_3 \leq \pi$ and $-\pi/2 < z_2 < \pi/2$.

3 Transformation update

3.1 Point-to-point distance minimization

If point-to-point distance minimization is used in Algorithm 1, see [5], no surface normals are required and the update of the transformation $(\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger}) = (\mathbf{R}^{*}, \mathbf{t}^{*})$ is obtained from

$$\left[\mathbf{R}^{*}, \mathbf{t}^{*}\right] = \underset{\mathbf{R},\mathbf{t}}{\arg\min}\, g(\mathbf{R}, \mathbf{t})\,, \tag{11}$$

where

$$g(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} w_i \|\mathbf{R}\mathbf{p}_i^{(k-1)} + \mathbf{t} - \mathbf{y}_i^{(k-1)}\|_2^2\,. \tag{12}$$

The minimization problem in (11) is a weighted rigid body transformation problem considered in, e.g., [33]. There always exists a solution set for this problem since all weights w_i are non-negative. The global minimizer of g, $(\mathbf{R}^{*}, \mathbf{t}^{*})$, is usually unique. A non-unique minimizer corresponds to a rotation about a line or a point.

Let

$$\hat{w} = \sum_{i=1}^{N} w_i\,,\;\; \bar{\mathbf{p}} = \frac{1}{\hat{w}} \sum_{i=1}^{N} w_i \mathbf{p}_i^{(k-1)}\,,\;\; \bar{\mathbf{y}} = \frac{1}{\hat{w}} \sum_{i=1}^{N} w_i \mathbf{y}_i^{(k-1)}\,,$$

denote the sum of the weights and weighted mean values of the data points $\mathbf{p}_i^{(k-1)}$ and the closest surface points $\mathbf{y}_i^{(k-1)}$. In the expression of g in (12), we replace $\mathbf{t}$ with a new independent vector $\mathbf{u}$ using the change of variables $\mathbf{t} = -\mathbf{R}\bar{\mathbf{p}} + \bar{\mathbf{y}} + \mathbf{u}$. The function g expressed in terms of $\mathbf{R}$ and $\mathbf{u}$ is then

$$\begin{aligned} & g(\mathbf{R}, -\mathbf{R}\bar{\mathbf{p}} + \bar{\mathbf{y}} + \mathbf{u}) = \\ & \quad \sum\nolimits_{i=1}^{N} w_i \|\mathbf{R}\mathbf{p}_i^{(k-1)} - \mathbf{R}\bar{\mathbf{p}} + \bar{\mathbf{y}} + \mathbf{u} - \mathbf{y}_i^{(k-1)}\|_2^2 = \\ & \quad \sum\nolimits_{i=1}^{N} w_i \|\mathbf{p}_i^{(k-1)} - \bar{\mathbf{p}}\|_2^2 + \sum\nolimits_{i=1}^{N} w_i \|\mathbf{y}_i^{(k-1)} - \bar{\mathbf{y}}\|_2^2 + \hat{w}\|\mathbf{u}\|_2^2 - 2\,\mathrm{trace}(\mathbf{R}\mathbf{C}) , \end{aligned} \tag{13}$$

where

$$\mathbf{C} = \sum_{i=1}^{N} [\, w_i \, \mathbf{p}_i^{(k-1)} \big(\mathbf{y}_i^{(k-1)}\big)^{\mathrm{T}} \,] - \hat{w}\bar{\mathbf{p}}\bar{\mathbf{y}}^{\mathrm{T}} \in \mathbb{R}^{3\times 3} .$$

Obviously, the expression of $g(\mathbf{R}, -\mathbf{R}\bar{\mathbf{p}} + \bar{\mathbf{y}} + \mathbf{u})$ is minimized by $\mathbf{u}^* = \mathbf{0}$ and by searching for a rotation matrix $\mathbf{R}$ that maximizes trace($\mathbf{RC}$). Finding this $\mathbf{R}^*$ is analogous to the un-weighted rigid body transformation problem, see, e.g., [4, 24, 44]. The rotation matrix $\mathbf{R}^*$ can be obtained from a singular value decomposition of $\mathbf{C}$, which is $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$, as $\mathbf{R}^* = \mathbf{V}\mathbf{U}^{\mathrm{T}}$. In some rare degenerate cases, this may however produce a reflection matrix, i.e. $\det(\mathbf{V}\mathbf{U}^{\mathrm{T}}) = -1$, which is corrected by reevaluating the matrix as $\mathbf{R}^* = \mathbf{V}\,\mathrm{diag}(1, 1, -1)\,\mathbf{U}^{\mathrm{T}}$. From the calculated $\mathbf{R}^*$ and $\mathbf{u}^* = \mathbf{0}$, in the change of variables expression, we obtain the translation vector $\mathbf{t}^* = \bar{\mathbf{y}} - \mathbf{R}^*\bar{\mathbf{p}}$.

The convergence of Algorithm 1 using point-to-point distance minimization is ensured according to the proof of convergence in [5]. Unfortunately, the asymptotic convergence is linear and in general very slow [37]. That is why another method for finding an appropriate update of the rigid body transformation is needed.

3.2 Point-to-plane distance minimization

An alternative to point-to-point distance minimization in Algorithm 1 is to use point-to-plane distance minimization. The considered planes, with unit normal $\mathbf{n}_i$, are the tangent planes of the surface at the surface points $\mathbf{y}_i$. We assume that our surface is G^1-continuous, i.e., tangentially continuous, so a surface normal exists for each point in S. Let

$$v(\mathbf{z}) = \sum_{i=1}^{N} \varrho\Big(\mathbf{n}_i^{\mathrm{T}}\big(\mathbf{R}(\mathbf{z})\mathbf{p}_i + \mathbf{t}(\mathbf{z}) - \mathbf{y}_i\big)\Big) , \tag{14}$$

be a function of point-to-plane distances. A second-order approximation to the function v about $\mathbf{z} = \mathbf{0}$, with the constant term excluded, is

$$q(\mathbf{z}) = \mathbf{a}^{\mathrm{T}}\mathbf{z} + \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} , \tag{15}$$

where $\mathbf{a}$ is the gradient and $\mathbf{H}$ is the Hessian of the function $v(\mathbf{z})$ at $\mathbf{z} = \mathbf{0}$. These quantities are given by

$$\mathbf{a} = \begin{bmatrix} \mathbf{A}^{\mathrm{T}}\mathbf{b} \\ \mathbf{B}^{\mathrm{T}}\mathbf{b} \end{bmatrix} \in \mathbb{R}^6, \ \ \mathbf{H} = \begin{bmatrix} \mathbf{A}^{\mathrm{T}}\mathbf{D}\mathbf{A} + \mathbf{E} & \mathbf{A}^{\mathrm{T}}\mathbf{D}\mathbf{B} \\ \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{A} & \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{6\times 6},$$

where

$$\mathbf{A} = \left[\mathbf{n}_i^{\mathrm{T}} \frac{\partial \mathbf{R}}{\partial z_j} \mathbf{p}_i\right]_{i,j} \in \mathbb{R}^{N\times 3}, \quad i = 1, \ldots, N, \quad j = 1, 2, 3\,,$$

$$\mathbf{b} = \begin{bmatrix} \psi(r_1) \\ \vdots \\ \psi(r_N) \end{bmatrix} \in \mathbb{R}^N, \; \mathbf{B} = \begin{bmatrix} \mathbf{n}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{n}_N^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{N\times 3}, \; \mathbf{D} = \operatorname{diag}\big(\psi'(r_1), \ldots, \psi'(r_N)\big) \in \mathbb{R}^{N\times N},$$

$$\mathbf{E} = \sum_{i=1}^{N} \psi(r_i) \begin{bmatrix} \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_1^2} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_1 \partial z_2} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_1 \partial z_3} \mathbf{p}_i \\ \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_2 \partial z_1} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_2^2} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_2 \partial z_3} \mathbf{p}_i \\ \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_3 \partial z_1} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_3 \partial z_2} \mathbf{p}_i & \mathbf{n}_i^{\mathrm{T}} \frac{\partial^2 \mathbf{R}}{\partial z_3^2} \mathbf{p}_i \end{bmatrix} \in \mathbb{R}^{3\times 3}.$$

Computationally, the gradient and the Hessian are acquired from these expressions after substantially simplifications. The submatrices above are not explicitly created.

The update of the rigid body transformation using point-to-plane distance minimization is obtained from the minimum of the second-order approximation q, so this approach is a Newton-type method. The minimizer of q, i.e., a full Newton step, is denoted by $\mathbf{z}^{\mathrm{N}}$. The asymptotic convergence using point-to-plane distance minimization is quadratic [37]. Unfortunately, the ICP-algorithm using point-to-plane distance minimization is not guaranteed to converge without line search or some other procedure of step size control, which is computational expensive for registration problems.

4 Trust region

4.1 Derivation

The trust region will be derived from the sum of weighted squared point-to-point distances. Let

$$\hat{p} = \sum_{i=1}^{N} w_i \|\mathbf{p}_i^{(k-1)} - \bar{\mathbf{p}}\|_2^2\,, \quad \hat{y} = \sum_{i=1}^{N} w_i \|\mathbf{y}_i^{(k-1)} - \bar{\mathbf{y}}\|_2^2\,,$$

and from our change of variables in Subsection 3.1 we have that $\mathbf{t} + \mathbf{R}\bar{\mathbf{p}} - \bar{\mathbf{y}} = \mathbf{u}$. From the latter expression in (13), it follows that the function g can be written as

$$g(\mathbf{R}, \mathbf{t}) = \hat{p} + \hat{y} + \hat{w}\,\|\mathbf{t} + \mathbf{R}\bar{\mathbf{p}} - \bar{\mathbf{y}}\|_2^2 - 2\operatorname{trace}(\mathbf{RC})\,. \tag{16}$$

For given $\hat{w}$, $\bar{\mathbf{p}}$, $\bar{\mathbf{y}}$, and $\mathbf{C}$, a function $h = h(\mathbf{z})$ is defined as

$$h(\mathbf{z}) = \hat{w}\,\|\mathbf{t}(\mathbf{z}) + \mathbf{R}(\mathbf{z})\bar{\mathbf{p}} - \bar{\mathbf{y}}\|_2^2 - 2\operatorname{trace}\big(\mathbf{R}(\mathbf{z})\mathbf{C}\big)\,. \tag{17}$$

Note that the function h consists of the terms in (16) that depends on $\mathbf{R}$ and $\mathbf{t}$, which are here parameterized by $\mathbf{z}$. The rigid body transformation $(\mathbf{R}^*, \mathbf{t}^*)$ gives the minimum value of h, written as $h^* = h(\mathbf{z}^*)$, where $\mathbf{z}^* = \vartheta(\mathbf{R}^*, \mathbf{t}^*)$. The value of h evaluated at the identity transformation $(\mathbf{I}, \mathbf{0})$ is written as $h^\circ = h(\mathbf{0})$.

We define a region $\Omega_\alpha \subset \mathbb{R}^6$ such that

$$\mathbf{z} \in \Omega_\alpha \Leftrightarrow h(\mathbf{z}) \leq (1-\alpha)h^* + \alpha h^\circ, \tag{18}$$

where α satisfies $0 \leq \alpha < 1$. If $\alpha = 0$ we have that $\Omega_\alpha = \{\mathbf{z}^*\}$. The boundary of Ω_α is a level surface in $\mathbb{R}^6$ where $h(\mathbf{z}) = (1-\alpha)h^* + \alpha h^\circ$. If $0 < \alpha < 1$ and $h^* < h^\circ$, the normal to the boundary of Ω_α is given by the gradient $\nabla h(\mathbf{z})$. Further, if $\alpha'' \leq \alpha'$, we have that $\Omega_{\alpha''} \subseteq \Omega_{\alpha'}$.

4.2 Convergence analysis

We will here explain how convergence of Algorithm 1, with respect to the value of the objective function f, is ensured using the region Ω_α. First, we need to establish some properties of the ingoing parts.

A quadratic approximation to $\varrho(s)$ about $s = r$ is

$$\eta(s, r) = \begin{cases} \varrho(r) - \frac{\psi(r)}{2} r + \frac{\psi(r)}{2r} s^2 & \text{if } r \neq 0\,, \\ \frac{\psi'(0)}{2} s^2 & \text{if } r = 0\,, \end{cases} \tag{19}$$

which satisfies $\eta(r, r) = \varrho(r)$ and $\frac{\partial}{\partial s}\eta(s, r) = \psi(r)$ at $s = r$. It follows from (3) and (19) that the quadratic approximation η can be written as

$$\eta(s, r) = \varrho(r) - \frac{\psi(r)}{2} r + \frac{w(r)}{2} s^2. \tag{20}$$

In [5], it is proved that for all criterion functions $\varrho \in \mathcal{Q}$, the associated quadratic approximation $\eta(s, r)$ fulfills

$$\eta(s, r) \geq \varrho(s) \;\; \forall s, r \in \mathbb{R}\,, \tag{21}$$

which holds with equality if $s = r$.

Let $s_i(\mathbf{z}) = \|\mathbf{R}(\mathbf{z})\mathbf{p}_i^{(k-1)} + \mathbf{t}(\mathbf{z}) - \mathbf{y}_i^{(k-1)}\|_2$ be a function of $\mathbf{z}$ quantifying the residual between a transformed data point and its associated closest surface point, and let $r_i = s_i(\mathbf{0})$ here be a fixed value of s_i. It follows from (20) that the minimization problems

$$\min_{\mathbf{z}} \sum_{i=1}^{N} \eta(s_i(\mathbf{z}), r_i)\,, \tag{22}$$

and

$$\min_{\mathbf{z}} \sum_{i=1}^{N} w(r_i) s_i^2(\mathbf{z})\,, \tag{23}$$

must share the same solution set and have the same properties since the objective functions differ only with a constant factor of one half and a sum, which is independent of $\mathbf{z}$. There always exist a solution set for these minimization problems since the weight function w is non-negative. The minimization problem in (23) is the same as the minimization problem in (11) if $\mathbf{R}$ and $\mathbf{t}$ are parameterized by $\mathbf{z}$. Since the difference between h and g is constant, g and η are linear related if $s = s_i$ and $r = r_i$, and $\eta(r, r) = \varrho(r)$, the following equivalences must hold

$$\begin{aligned} & h(\mathbf{z}) \leq h(\mathbf{0}) \Leftrightarrow g(\mathbf{R}, \mathbf{t}) \leq g(\mathbf{I}, \mathbf{0}) \\ & \Leftrightarrow \textstyle\sum_{i=1}^{N} \eta(s_i(\mathbf{z}), r_i) \leq \sum_{i=1}^{N} \eta(r_i, r_i) = \sum_{i=1}^{N} \varrho(r_i)\,. \end{aligned} \tag{24}$$

In a minimum of the function f, let the transformed data points and associated closest surface points be given. Further, let $r_i = r_i(\mathbf{z})$ be the point-to-point distance between a transformed data point, dependent of $\mathbf{z}$, and its associated closest surface point. We consider the derivatives of r_i with respect to z_j, $j = 1, \ldots, 6$, as zero if $r_i = 0$. The function g in (12), in terms of the point-to-point distances r_i, can be written as $g = g(\mathbf{z})$. The optimality condition for g is

$$\nabla g = 2 \sum_{i=1}^{N} w_i r_i \nabla r_i = \mathbf{0} . \tag{25}$$

Assume now that the set of surface points S is a finite countable set of points. In the neighborhood of the minimum of f, the point-to-surface distances can be considered as the point-to-point distances $r_i(\mathbf{z})$, and the function f can locally be written as a function of $\mathbf{z}$ in terms of r_i. The optimality condition for f is

$$\nabla f = \sum_{i=1}^{N} \psi(r_i) \nabla r_i = \mathbf{0} . \tag{26}$$

At a stationary point of f, these two optimality conditions are equivalent, which follows from the identity $w(r_i) r_i = \psi(r_i)$, thus validating our choice of weight function. Also, note that after applying (11), the condition $\nabla g = \mathbf{0}$ will be fulfilled.

The derivatives of r_i with respect to z_j, $j = 1, \ldots, 6$, do not exist if $r_i = 0$ but it does not cause any trouble though, since ψ is continuous and $\psi(0) = 0$. Thus, the terms of ∇g and ∇f for the zero residuals vanish. An assumption about a finite countable set of surface points is done. If for example the set of surface points S consists of points on a continuous surface, this assumption is not fulfilled. However, points on a continuous surface can be considered as a finite countable set of points, where the points are evenly distributed over the surface and the number of points are arbitrary large.

Theorem 1 *The ICP-algorithm, given by Algorithm 1, converges globally to a stationary point of the objective function f in (2) and the sequence of function values f_k over the iterations $k = 1, 2, \ldots$ is monotonically decreasing to the value at the stationary point for all criterion functions $\varrho \in \mathcal{Q}$ if the update of the transformation $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$ satisfies $\vartheta(\mathbf{R}^\dagger, \mathbf{t}^\dagger) \in \Omega_\alpha$, where $0 \le \alpha < 1$.*

Proof The convergence is ensured for $\alpha = 0$ according to the proof of convergence in [5] since in that case $(\mathbf{R}^\dagger, \mathbf{t}^\dagger) = (\mathbf{R}^*, \mathbf{t}^*)$. It remains to proof convergence for $0 < \alpha < 1$. For all quantities with index i, we let $i = 1, \ldots, N$ without writing it explicitly. The parameter values of the update of the transformation is written as $\mathbf{z}^\dagger$, i.e., $\mathbf{z}^\dagger = \vartheta(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$.

In iteration k, the set of data points $\{\mathbf{p}_i^{(k-1)}\}$ from iteration $k - 1$ is given and the set of corresponding closest surface points $\mathbf{y}_i^{(k-1)} = \mathcal{C}(\mathbf{p}_i^{(k-1)}, S)$ is then obtained.

Residuals $r_i^{(k-1)} = \|\mathbf{p}_i^{(k-1)} - \mathbf{y}_i^{(k-1)}\|_2$ and weights $w_i = w(r_i^{(k-1)})$ are calculated, and the value f_{k-1} of the objective function f is

$$f_{k-1} = \sum_{i=1}^{N} \varrho(r_i^{(k-1)}) .$$

Since $\eta(r_i^{(k-1)}, r_i^{(k-1)}) = \varrho(r_i^{(k-1)})$ for all residuals, it follows that

$$\sum_{i=1}^{N} \eta(r_i^{(k-1)}, r_i^{(k-1)}) = f_{k-1} .$$

If all weights w_i are zero, a maximum is reached and we will not obtain any better solution using the ICP-algorithm as given by Algorithm 1. The iterations are then terminated and the current transformation is returned. If any of the weights w_i are greater than zero, the updating rigid body transformation $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$ satisfying $\vartheta(\mathbf{R}^\dagger, \mathbf{t}^\dagger) \in \Omega_\alpha$ is applied to $\mathbf{p}_i^{(k-1)}$ giving the update of the data points $\mathbf{p}_i^{(k)} = \mathbf{R}^\dagger \mathbf{p}_i^{(k-1)} + \mathbf{t}^\dagger$ and new residuals $s_i^{(k)} = \|\mathbf{p}_i^{(k)} - \mathbf{y}_i^{(k-1)}\|_2$ are calculated. Since $0 < \alpha < 1$, we have that $h(\mathbf{z}^\dagger) \leq (1-\alpha)h^* + \alpha h^\circ \leq h^\circ = h(\mathbf{0})$ and it follows from (24) that

$$\sum_{i=1}^{N} \eta(s_i^{(k)}, r_i^{(k-1)}) \leq \sum_{i=1}^{N} \eta(r_i^{(k-1)}, r_i^{(k-1)}) ,$$

is satisfied. Let

$$e_k = \sum_{i=1}^{N} \varrho(s_i^{(k)}) ,$$

and according to (21), see [5], we have that $\eta(s_i^{(k)}, r_i^{(k-1)}) \geq \varrho(s_i^{(k)})$ for all i and thus

$$e_k \leq \sum_{i=1}^{N} \eta(s_i^{(k)}, r_i^{(k-1)}) ,$$

which gives that $e_k \leq f_{k-1}$.

In the next iteration, i.e., iteration $k+1$, the closest surface points $\mathbf{y}_i^{(k)} = \mathcal{C}(\mathbf{p}_i^{(k)}, S)$ are found, and the residuals $r_i^{(k)} = \|\mathbf{p}_i^{(k)} - \mathbf{y}_i^{(k)}\|_2$ and the new weights $w_i = w(r_i^{(k)})$ are calculated, which gives the value f_k of f as

$$f_k = \sum_{i=1}^{N} \varrho(r_i^{(k)}) .$$

The points $\mathbf{y}_i^{(k)}$ are the closest surface points to $\mathbf{p}_i^{(k)}$ so it is clear that

$$\|\mathbf{p}_i^{(k)} - \mathbf{y}_i^{(k)}\|_2 \leq \|\mathbf{p}_i^{(k)} - \mathbf{y}_i^{(k-1)}\|_2 ,$$

i.e., $r_i^{(k)} \leq s_i^{(k)}$, holds for all i. Since ϱ is a monotone increasing function for positive residuals, we have that $f_k \leq e_k$. The equality is satisfied if and only if $\|\mathbf{p}_i^{(k)} - \mathbf{y}_i^{(k-1)}\|_2 = \mathrm{d}(\mathbf{p}_i^{(k)}, S)$ for all i, i.e., if and only if all points $\mathbf{y}_i^{(k-1)}$ are closest surface points to the data points $\mathbf{p}_i^{(k)}$.

To summarize, we can conclude that

$$0 \leq f_k \leq e_k \leq f_{k-1} \,,$$

is fulfilled. The lower bound is zero since a sum of non-negative values cannot be negative. It gives that the sequences of f_k and e_k are monotonically decreasing and bounded below. Thus, the sequences will converge to a real value, which is the same for both sequences.

Since $0 \leq \alpha < 1$, the function g will eventually be minimized and the updating transformation will satisfy $(\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger}) = (\mathbf{R}^{*}, \mathbf{t}^{*})$. There are just two possibilities for the transformation $(\mathbf{R}^{*}, \mathbf{t}^{*})$ that do not change the values of f_k and e_k to their updates in the subsequent iteration. One of these possibilities is that the transformation is equal to the identity transformation, $(\mathbf{R}^{*}, \mathbf{t}^{*}) = (\mathbf{I}, \mathbf{0})$, and that occurs when the function g has a unique minimum. For this transformation, all residuals are preserved so that $s_i = r_i$ for all i. The other possibility is that the transformation $(\mathbf{R}^{*}, \mathbf{t}^{*})$ rotates the data points around a line or a point and that occurs when the function g does not have a unique minimum. The residuals associated with positive weights are preserved also for this transformation. However, residuals corresponding to weights that are equal to zero can change but all terms of both ∇g and ∇f corresponding to the zero-valued weights are zero and unaffected by these changes.

The function g is eventually minimized so $\nabla g = \mathbf{0}$, the residuals associated with positive weights are preserved, and the identity $w(r_i)r_i = \psi(r_i)$ holds. Thus, from the optimality conditions (25) and (26), the gradient of f satisfies $\nabla f = \frac{1}{2}\nabla g = \mathbf{0}$ and the condition of a stationary point for f is fulfilled. Suppose that the sequence of f_k will converge to a value that is larger than the value of the objective function f at the stationary point. Then, this would contradict the operation of the closest point operator $\mathcal{C}$. We can do the conclusion that the ICP-algorithm, given by Algorithm 1, converges globally to a stationary point of the objective function f in (2). □

Theorem 1 is similar to the convergence theorem given in [5] but it improves the possibilities to find good updates of the transformation in comparison to the robust ICP-algorithm based on point-to-point distance minimization. From Theorem 1, we know that the ICP-algorithm, given by Algorithm 1, converges to a stationary point for all criterion functions in $\mathcal{Q}$ if the update of the transformation $(\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger})$ satisfies $\vartheta(\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger}) \in \Omega_{\alpha}$, where $0 \leq \alpha < 1$. This is an essential property of an optimization algorithm. There is a possibility that the ICP-algorithm will return a transformation $(\mathbf{R}, \mathbf{t})$ that gives a maximum of f. It can happen if the used criterion function ϱ has a constant value for inputs of large magnitude, and all initial residuals $\|\mathbf{p}_i^{(0)} - \mathbf{y}_i^{(0)}\|_2$ are of such significant size. The value of the criterion function ϱ cannot be larger than its value at these residuals and therefore, the objective function f attains a maximum. It can also happen, but it is extremely unlikely, that the ICP-algorithm will end up in a saddle point after some iterations. We cannot exclude that possibility, but it will only happen in some very rare cases.

In general, we can expect that the ICP-algorithm will reach a local minimum. If for example the initial orientation of the data points is turned the wrong way, being upside-down, or having some other apparent misalignment in comparison with the surface points in S, the ICP-algorithm will probably not end up in the global minimum. The same thing can also happen if the surface has many similar repeating geometrical features. However, if the initial rigid body transformation, $(\mathbf{R}^{(0)}, \mathbf{t}^{(0)})$, is carefully chosen, we can expect that the achieved stationary point is the global minimum.

4.3 Adaptive extension

The advantage of using the region Ω_α is that convergence can always be ensured for all updating transformations $(\mathbf{R}^\dagger, \mathbf{t}^\dagger)$ satisfying $\vartheta(\mathbf{R}^\dagger, \mathbf{t}^\dagger) \in \Omega_\alpha$ without any unnecessary evaluations of the objective function f. Unfortunately, there is a risk that after some iterations the region Ω_α will be limited to a small neighborhood about $\vartheta(\mathbf{R}^*, \mathbf{t}^*)$. Thus, using an updating transformation satisfying $\vartheta(\mathbf{R}^\dagger, \mathbf{t}^\dagger) \in \Omega_\alpha$ might be quite conservative. The full Newton step $\mathbf{z}^{\mathrm{N}}$ would be far outside of the region, which severely impacts the asymptotic convergence rate.

To remedy this, introduce for $k \geq 2$

$$\beta = \frac{\|\mathbf{z}_{k-1}^\dagger\|_2}{\|\mathbf{z}_{k-1}^\dagger\|_2 + \|\mathbf{z}_{k-1}^{\mathrm{N}} - \mathbf{z}_{k-1}^\dagger - \mathbf{z}_k^{\mathrm{N}}\|_2}, \tag{27}$$

where $\mathbf{z}_k^{\mathrm{N}}$ and $\mathbf{z}_{k-1}^{\mathrm{N}}$ are the full Newton steps in iteration k and $k-1$, and $\mathbf{z}_{k-1}^\dagger$ is the used parameter update in iteration $k-1$. The value of β is describing how much a second-order approximation q to f has changed in an update from iteration $k-1$ to iteration k. If $\|\mathbf{z}_{k-1}^\dagger\|_2 > 0$, we have $\beta \in (0, 1]$. A value of β that is equal to one indicates that q is unchanged from the previous iteration, and for a significant change of q we get a value of β close to zero. In the iterations of the ICP-algorithm far from an optimum of f, we can expect significantly changes of q giving values of β close to zero. In the iterations close to an optimum of f, just minor changes of q are expected, which give values of β close to one. Using β, we define a region $\Phi \subset \mathbb{R}^6$ in iteration $k \geq 2$ such that

$$\mathbf{z} \in \Phi \Leftrightarrow h(\mathbf{z}) \leq \max\left\{h\big(\mathbf{z}_k^* + \beta(\mathbf{z}_k^{\mathrm{N}} - \mathbf{z}_k^*)\big), h\big(\mathbf{z}_k^* - \beta(\mathbf{z}_{k-1}^\dagger + \mathbf{z}_k^*)\big)\right\}. \tag{28}$$

In the first iteration, i.e., $k = 1$, we let $\Phi = \Omega_\alpha$, where $\alpha = 1$.

Schematic illustrations of the regions that constrain a rigid body transformation $(\mathbf{R}, \mathbf{t})$ in its parameter space are given in Fig. 2. The advantage of using the region Φ is that it allows the algorithm to use full Newton steps close to the solution where the second-order approximation $q(\mathbf{z})$ will not change so much. This however invalidates the required properties for the proof of convergence (see Theorem 1). The approach has been seen to work very well in practice, though.

A value of β close to one indicates that the ICP-algorithm is near the optimum, and a full Newton step will probably be very useful. If the value of β is close to zero, it gives an indication that the ICP-algorithm is far from the optimum and the full Newton step will probably not be reliable so it must be damped. However, β will be one if and only if the optimum of the second-order approximation $q(\mathbf{z})$ is unchanged between two iterations, which is not likely. That is why a full Newton step will not

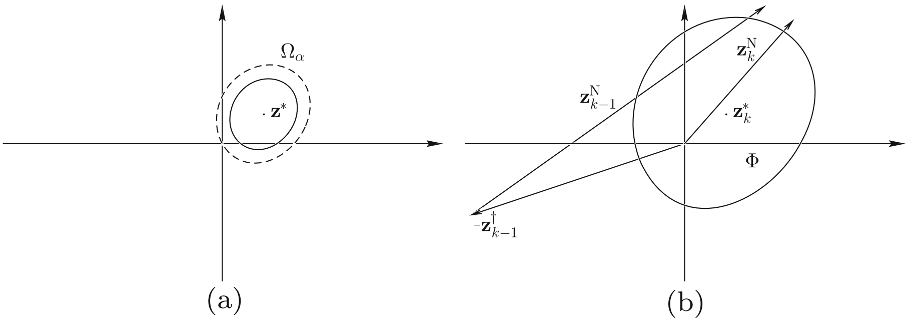
Ω_α
$\cdot\,\mathbf{z}^*$
(a)
$\mathbf{z}_k^{\mathrm{N}}$
$\mathbf{z}_{k-1}^{\mathrm{N}}$
$\cdot\,\mathbf{z}_k^*$
Φ
$\mathbf{z}_{k-1}^{\dagger}$
(b)

Fig. 2 Schematic illustrations of different regions in $\mathbb{R}^6$ where $\mathbf{z}^* = \vartheta(\mathbf{R}^*, \mathbf{t}^*)$. The index k and $k-1$ in **b** represents the iteration number. **b** Ω_α. **a** Φ

be utilized if only $h\big(\mathbf{z}_k^* + \beta(\mathbf{z}_k^{\mathrm{N}} - \mathbf{z}_k^*)\big)$ is used. The way the region Φ is defined makes a full Newton step possible under good conditions.

4.4 Application

The way we propose to update the transformation in the ICP-algorithm is to solve the constrained minimization problem

$$\begin{aligned} \mathbf{z}^{\dagger} &= \underset{\mathbf{z}\,\in\,\Omega_\alpha \text{ or } \Phi}{\arg\min}\ \mathbf{a}^{\mathrm{T}}\mathbf{z} + \tfrac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z}\,, \\ [\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger}] &= \varphi(\mathbf{z}^{\dagger})\,, \end{aligned} \tag{29}$$

and evaluate the rigid body transformation $(\mathbf{R}^{\dagger}, \mathbf{t}^{\dagger})$ from the obtained solution $\mathbf{z}^{\dagger}$. The choice of constraint, Ω_α or Φ, depends on which region that is used and forms a natural trust region for registration problems. The trust region subproblem in (29), with $q(\mathbf{z})$ as a model function, is a modification of the point-to-plane distance minimization and the Newton-type update described in Subsection 3.2. It combines the best properties of point-to-point distance minimization and point-to-plane distance minimization.

The way we are solving the constrained minimization problem in (29) is by using a reciprocal barrier function of the constraint. A feasible point in both Ω_α and Φ is $\mathbf{z}^*$, and this point is used as an initial choice of $\mathbf{z}$. The model function $q(\mathbf{z})$ together with the barrier function is minimized using Newton's method, see, e.g., [16] for more information about possible optimization methods.

The constrained minimization problem in (29) does not have to be solved with any high accuracy, since it will just give an update of the parameters in an iterative process. Computations of Ω_α or Φ can be done at a very low cost when the required quantities are given. The acquirement of these quantities is the principal additional computational cost compared to the cost for obtaining a Newton-type update without any step size control, as described in Subsection 3.2. The computational cost to obtain these required quantities is small in comparison to the cost of finding closest surface points and obtaining the Hessian and the gradient. When solving the minimization problem in (29) having all necessary quantities, there is no need to use the N data points nor the N corresponding closest surface points and surface normals. Therefore, only a minor number of additional computations are required to use the trust regions Ω_α or Φ.

5 Numerical results

5.1 Test problems

The robust criterion functions ϱ_{Ca} and ϱ_{We}, and the non-robust least squares criterion function $\varrho_{\mathrm{LS}}(r) = r^2$, are used to estimate rigid body transformations in different surface registration problems. Our surface registration problems can be formulated as (5), where S is a set of surface points and most of the data points represent the shape of the surface quite well.

Six different $\boldsymbol{G}^1$-continuous surfaces, see Fig. 3, are used in the numerical experiment. The first four surfaces might seem to have sharp edges and corners but they are actually rounded. From each of these six surfaces, we have sampled 1,000,000 points. A local second-order surface approximation is used at each of these points to represent the surface as a set of surface points S. From the surfaces, we are also sampling a set of $N_{\mathrm{g}} = 1{,}000$ (“good”) non-disturbed data points $\{\mathbf{p}_i\}_{i=1}^{N_{\mathrm{g}}}$, and $N_{\mathrm{b}} = 500$ (“bad”) disturbed data points $\{\mathbf{p}_i\}_{i=N_{\mathrm{g}}+1}^{N_{\mathrm{g}}+N_{\mathrm{b}}}$, where the disturbance are randomly generated from a normal distribution in the 3D space. All these $N = N_{\mathrm{g}} + N_{\mathrm{b}}$ data points are transformed by a rigid body transformation to their initial positions. The data points in their initial positions are also shown in Fig. 3.

5.2 Experiment

The ICP-algorithm with re-scaled residuals, given by Algorithm 2, and the trust regions Ω_α and Φ are used to solve the introduced test problems. Point-to-point distance minimization (11), as considered in [5], and point-to-plane distance minimization using a full Newton step, without any step size control, are also used in this experiment as a comparison.

Theorem 1 tells that the convergence of Algorithm 1 is ensured for Ω_α, where $0 \le \alpha < 1$. If $\alpha = 1$, convergence cannot be ensured for Ω_α but by using $\alpha = 1$ in combination with point-to-plane distance minimization (29) it is very likely that convergence is achieved. In order to make a clear distinction between the result when using (29) and the trust region Ω_α, and the result when using point-to-point distance minimization, we are here using $\alpha = 1$, and the region Ω_α is simply written as Ω_1.

If the magnitude of any of the updating rotational parameters exceeds $\pi/4$, the whole vector $\mathbf{z}^\dagger$ is re-scaled such that this condition is fulfilled. It gives an obvious improvement of the comparative point-to-plane distance minimization without any step size control. The search for the closest point in S to an arbitrary data point, written as the operator $\mathcal{C}$, is done by first finding the closest point among all the 1,000,000 sampled surface points in S. The corresponding local second-order surface approximation is then used to find an even better approximation of the closest point on the

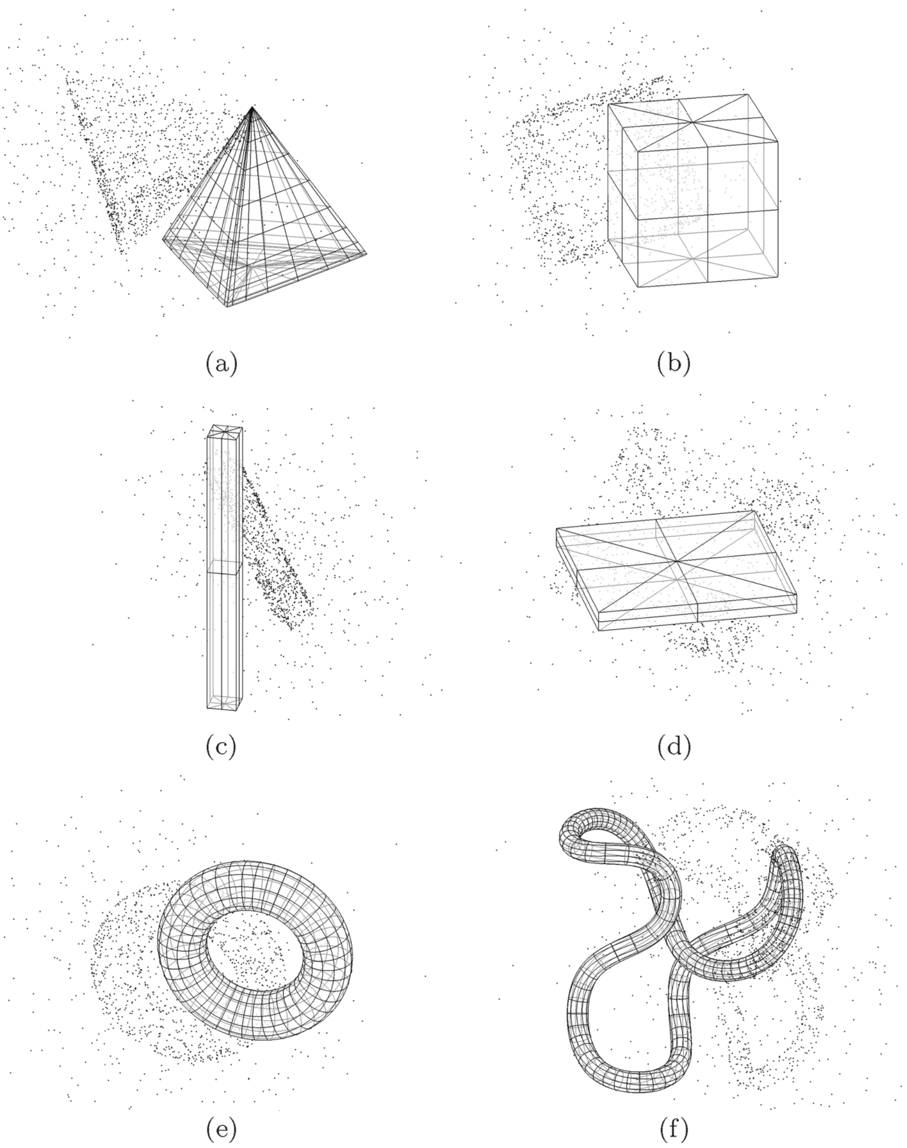

Fig. 3 Surfaces and data points. **a** Rounded tetrahedron. **b** Rounded cube. **c** Rounded cuboid. **d** Rounded cuboid. **e** Torus. **f** Swept

surface from which the required surface normal is computed. When using an update obtained from the trust region subproblem (29), a minimization problem has to be solved. As a starting point in this iterative process, we are using $\mathbf{z}^* = \vartheta(\mathbf{R}^*, \mathbf{t}^*)$ which is known to be feasible. The constrained minimization problem in (29) is solved approximately by using a barrier method with a reciprocal barrier function.

The computational time to solve this problem is about five microseconds, which is far less than the computational time for the other computations in each iteration.

The scaling σ_k should be chosen so that the algorithm converges fast to a minimum without getting stuck at a maximum outside the region of convergence. During the iterations, the value of σ_k should approach to σ_* from above. However, the value of σ_* might not be known. The way we chose the scaling σ_k is to let

$$\sigma_k = 1.90 \, \text{median} \left(\{r_i\}_{i=1}^{N} \right) , \tag{30}$$

which adapts the scaling to the current distribution of the residuals. If the residuals r_i are positive observations from a normal distribution $\mathcal{N}(0, 1)$, about 80 % of the residuals will be less than σ_k. Using the median in the estimation of σ_k provides less vulnerability to outliers than if the scaling were based on the actual upper residuals. We can also be sure that not all weights w_i are equal to zero so that the algorithm will not get stuck at a maximum.

The aim of the experiment is to test the performance of the proposed methods and to make a comparison with the other methods. In iteration k, we are using

$$\ell_k = \frac{1}{N_{\text{g}}} \sum_{i=1}^{N_{\text{g}}} r_i^2 , \tag{31}$$

as a measure of accuracy, which is the mean squared residual between the ("good") non-disturbed data points and the surface. It is non-negative and should eventually approach to a value close to zero. Since the search for the closest surface point is just approximate by the usage of the 1,000,000 sampled surface points and the corresponding local second-order surface approximation, we cannot expect that the value of ℓ_k will be exactly zero. The values of the parameters κ in (4) are used to make the criterion functions comparative. The iterations are terminated when $\|\mathbf{z}_k^\dagger\|_2 < 10^{-4}$. Figure 4 shows the result from the numerical experiment.

5.3 Conclusions

In Fig. 4, we can see that the robust methods using the proposed trust regions manage to find good solutions with respect to the measure of accuracy ℓ_k given in (31) despite the large residuals in the "bad" data. Welsch's criterion function ϱ_{We} gives the best result but Cauchy's criterion function ϱ_{Ca} is not far behind. We can also see that all least squares methods fail.

As expected, the point-to-point distance minimization by updating the transformation using (11) gives a slow convergence. Point-to-plane distance minimization and a trust region, Ω_α or Φ, give much faster convergence. We can also see in Fig. 4 that the comparative method using point-to-plane distance minimization and a Newton type update without any step size control gives an extremely unreliable output. It fails in test problems 1, 4, 5, and 6 for at least one of Cauchy's and Welsch's criterion functions.

The convergence of ℓ_k for surface 3 and 4, see Fig. 4c, d, are much slower compared to the other ones. Surface 3 is drawn-out in one direction, which gives slow

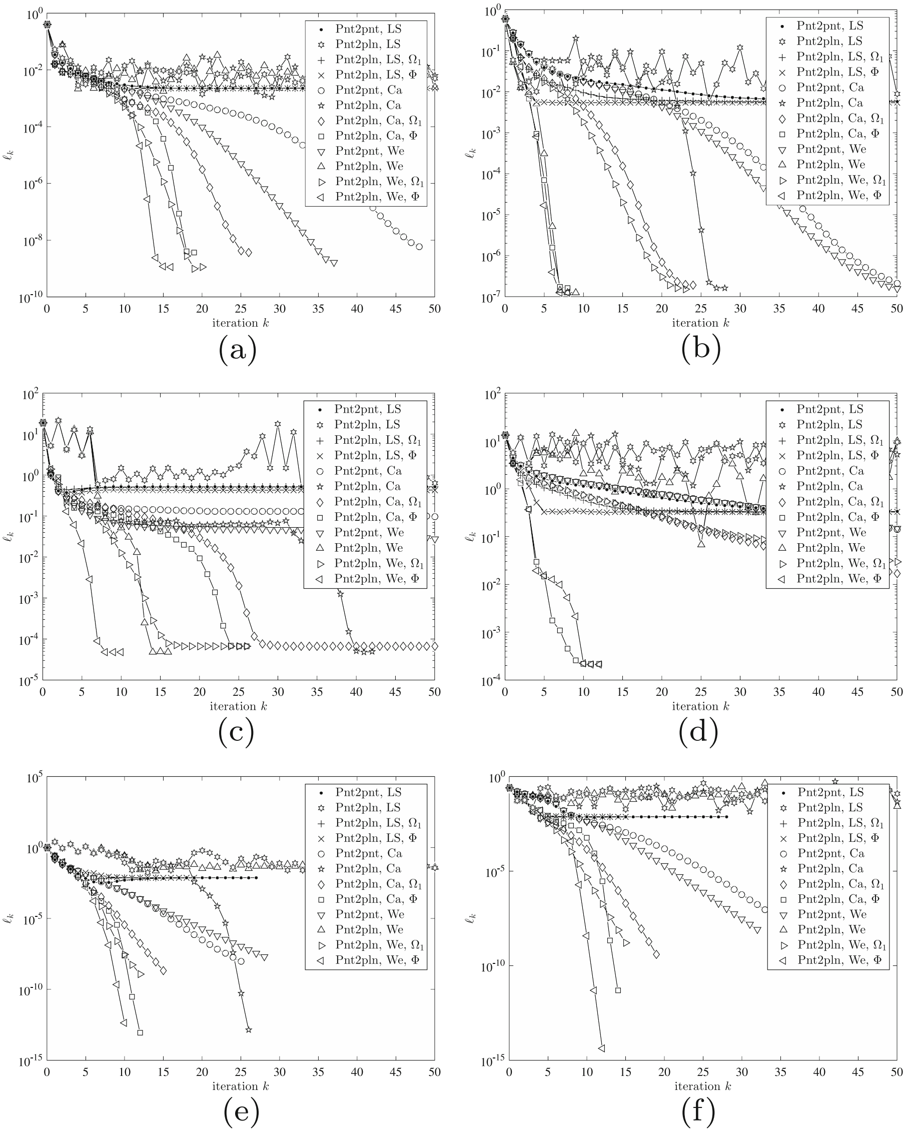

Fig. 4 Observed values of ℓ_k in iteration k, given by (31), for the **a** rounded tetrahedron, **b** rounded cube, **c** rounded cuboid, **d** rounded cuboid, **e** torus, and **f** swept. The values marked "Pnt2pnt, XX" are obtained with point-to-point distance minimization (11). The values marked "Pnt2pln, XX" (and nothing else) are obtained with point-to-plane distance minimization and a full Newton step. The values marked "Pnt2pln, XX, Ω_1/Φ" are obtained with (29). The robust criterion functions ϱ_{Ca} and ϱ_{We} and the non-robust least squares criterion function are used in the tests

convergence in that direction. Surface 4 is flattened; it is almost a plane giving the same difficulties of convergence as points on a plane, which is slow convergence in two directions and one rotation. The other surfaces have more distinct shapes resulting in faster convergence. It is obvious that the shape of the surface has a significant impact on the convergence behavior for the ICP-algorithm.

Using an extension of the trust region Ω_α, like for example Φ, to obtain a faster convergence seems to be a good idea since the trust region Ω_α will be quite small after some iterations. By setting the value of β according to (27), we get an adapted size of the trust region dependent on how much the second-order approximation q changes between two successive iterates. From our numerical experiment, we can conclude that using the robust criterion function, and the trust region Φ to constrain the updating transformation, gives the best observed convergence with respect to the measure of accuracy ℓ_k.

References

1. Ahookhosh, M., Amini, K.: A nonmonotone trust region method with adaptive radius for unconstrained optimization problems. Comput. Math. Appl. **60**(3), 411 – 422 (2010). doi:10.1016/j.camwa.2010.04.034
2. Ahookhosh, M., Ghaderi, S.: Two globally convergent nonmonotone trust-region methods for unconstrained optimization. J. Appl. Math. Comput. **50**(1), 529–555 (2016). doi:10.1007/s12190-015-0883-9
3. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graph. **27**(3), 85:1–85:10 (2008). doi:10.1145/1360612.1360684
4. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell. **9**(5), 698–700 (1987). doi:10.1109/tpami.1987.4767965
5. Bergström, P., Edlund, O.: Robust registration of point sets using iteratively reweighted least squares. Comput. Optim. Appl. **58**(3), 543–561 (2014). doi:10.1007/s10589-014-9643-2
6. Bergström, P., Edlund, O., Söderkvist, I.: Repeated surface registration for on-line use. Int. J. Adv. Manuf. Technol. **54**, 677–689 (2011). doi:10.1007/s00170-010-2950-6
7. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992). doi:10.1109/34.121791
8. Birdal, T., Bala, E., Eren, T., Ilic, S.: Online inspection of 3D parts via a locally overlapping camera network. In: Winter Conference on Applications of Computer Vision (WACV). IEEE Computer Society (2016). doi:10.1109/wacv.2016.7477630
9. Bispo, E.M., Fisher, R.B.: Free-form surface matching for surface inspection. Clarendon Press, NY, USA (1996)
10. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. Comput. Graph. Forum **32**(5), 113–123 (2013). doi:10.1111/cgf.12178
11. Chen, F., Brown, G.M., Song, M.: Overview of three-dimensional shape measurement using optical methods. Opt. Eng. **39**(1), 10–22 (2000). doi:10.1117/1.602438
12. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. Image Vis. Comput. **10**(3), 145–155 (1992). doi:10.1016/0262-8856(92)90066-C
13. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust region methods. Society for Industrial & Applied Mathematics (SIAM) (2000). doi:10.1137/1.9780898719857
14. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: Conference on Computer Vision and Pattern Recognition, pp. 998–1005. IEEE Computer Society (2010). doi:10.1109/cvpr.2010.5540108
15. Fitzgibbon, A.W.: Robust registration of 2D and 3D point sets. Image Vis. Comput. **21**(13–14), 1145–1153 (2003). doi:10.1016/j.imavis.2003.09.004
16. Fletcher, R. Practical methods of optimization, 2nd edn. NY, USA, Wiley-Interscience (1987)
17. Gay, D.M.: Computing optimal locally constrained steps. SIAM J. Sci. Stat. Comput. **2**(2), 186–197 (1981). doi:10.1137/0902016

18. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: Proceedings of the Third Eurographics Symposium on Geometry Processing, SGP '05. Eurographics Association, Switzerland (2005)
19. Godin, G., Rioux, M., Baribeau, R.: Three-dimensional registration using range and intensity information. In: El-Hakim, S.F. (ed.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 2350, pp. 279–290 (1994)
20. Gould, N.I.M., Sainvitu, C., Toint, P.L.: A filter-trust-region method for unconstrained optimization. SIAM J. Optim. **16**(2), 341–357 (2005). doi:10.1137/040603851
21. Greenspan, M., Yurick, M.: Approximate k-d tree search for efficient ICP. In: 2003. 3DIM 2003. Proceedings. Fourth International Conference on 3-D Digital Imaging and Modeling, pp. 442–448 (2003). doi:10.1109/IM.2003.1240280
22. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust statistics. Wiley, NY, USA (1986)
23. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust statistics: the approach based on influence functions. Wiley (2005). doi:10.1002/9781118186435
24. Hanson, R.J., Norris, M.J.: Analysis of measurements based on the singular value decomposition. SIAM J. Sci. Stat. Comput. **2**(3), 363–373 (1981). doi:10.1137/0902029
25. Hei, L.: A self-adaptive trust region algorithm. J. Comput. Math. **21**(2), 229–236 (2003)
26. Huber, P.J.: Robust statistics. Wiley, NY, USA (1981)
27. Kaneko, S., Kondo, T., Miyamoto, A., Igarashi, S.: Robust ICP registration algorithm extended by M-estimation. J. Jpn. Soc. Precis. Eng. **67**(8), 1276–1280 (2001)
28. Körner, K., Windecker, R.: Absolute macroscopic 3-D measurements with the innovative depth-scanning fringe projection technique (DSFP). Opt. - Int. J. Light Electron Opt. **112**(9), 433 – 441 (2001). doi:10.1078/0030-4026-00082
29. Li, Y., Gu, P.: Free-form surface inspection techniques state of the art review. Comput.-Aided Des. **36**(13), 1395–1417 (2004). doi:10.1016/j.cad.2004.02.009
30. Li, Z., Xu, Z., Cen, M., Ding, X.: Robust surface matching for automated detection of local deformations using least-median-of-squares estimator. Photogramm. Eng. Remote Sens. **67**(11), 1283–1292 (2001)
31. Luhmann, T.: Close range photogrammetry for industrial applications. ISPRS J. Photogramm. Remote Sens. **65**(6), 558–569 (2010). doi:10.1016/j.isprsjprs.2010.06.003
32. Maronna, R.A., Martin, D.R., Yohai, V.J.: Robust statistics: theory and methods. Wiley, NY, USA (2006)
33. Maurer C.R., J., Aboutanos, G., Dawant, B., Maciunas, R., Fitzpatrick, J.: Registration of 3-D images using weighted geometrical features. IEEE Trans. Med. Imaging **15**(6), 836–849 (1996). doi:10.1109/42.544501
34. Mavridis, P., Andreadis, A., Papaioannou, G.: Efficient sparse ICP. Comput. Aided Geom. Des. **35-36**, 16–26 (2015). doi:10.1016/j.cagd.2015.03.022
35. Mellado, N., Aiger, D., Mitra, N.J.: Super 4PCS fast global pointcloud registration via smart indexing. Comput. Graph. Forum **33**(5), 205–215 (2014). doi:10.1111/cgf.12446
36. Phillips, J.M., Liu, R., Tomasi, C.: Outlier robust ICP for minimizing fractional RMSD. In: Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM) (2007)
37. Pottmann, H., Huang, Q.X., Yang, Y.L., Hu, S.M.: Geometry and convergence analysis of algorithms for registration of 3D shapes. Int. J. Comput. Vis. **67**(3), 277–296 (2006). doi:10.1007/s11263-006-5167-2
38. Pulli, K.: Multiview registration for large data sets. In: International Conference on 3-D Digital Imaging and Modeling, pp. 160–168. IEEE Computer Society (1999). doi:10.1109/IM.1999.805346
39. Reza Peyghami, M., Ataee Tarzanagh, D.: A relaxed nonmonotone adaptive trust region method for solving unconstrained optimization problems. Comput. Optim. Appl. **61**(2), 321–341 (2015). doi:10.1007/s10589-015-9726-8
40. Rousseeuw, P.J., Leroy, A.M.: Robust regression & outlier detection. Wiley, NY, USA (1987)
41. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: IEEE Computer Society, pp. 145–152 (2001). doi:10.1109/im.2001.924423
42. Sansoni, G., Trebeschi, M., Docchio, F.: State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. Sensors **9**(1), 568–601 (2009). doi:10.3390/s90100568

43. Sjödahl, M., Synnergren, P.: Measurement of shape by using projected random patterns and temporal digital speckle photography. Appl. Opt. **38**(10), 1990–1997 (1999). doi:10.1364/ao.38.001990
44. Söderkvist, I.: Perturbation analysis of the orthogonal procrustes problem. BIT **33**(4), 687–694 (1993). doi:10.1007/bf01990543
45. Sorensen, D.C.: Newton's method with a model trust region modification. SIAM J. Numer. Anal. **19**(2), 409–426 (1982). doi:10.1137/0719026
46. Sproull, R.F.: Refinements to nearest-neighbor searching in k-dimensional trees. Algorithmica **6**(1), 579–589 (1991). doi:10.1007/BF01759061
47. Stewart, C., Tsai, C.L., Roysam, B.: The dual-bootstrap iterative closest point algorithm with application to retinal image registration. IEEE Trans. Med. Imaging **22**(11), 1379 –1394 (2003). doi:10.1109/TMI.2003.819276
48. Trucco, E., Fusiello, A., Roberto, V.: Robust motion and correspondence of noisy 3-D point sets with missing data. Pattern Recogn. Lett. **20**(9), 889–898 (1999). doi:10.1016/S0167-8655(99)00055-0
49. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 311–318. ACM Press, NY, USA (1994). doi:10.1145/192161.192241
50. Wang, Z., Du, H., Park, S., Xie, H.: Three-dimensional shape measurement with a fast and accurate approach. Appl. Opt. **48**(6), 1052–1061 (2009). doi:10.1364/ao.48.001052
51. Yuan, Y.X.: Recent advances in trust region algorithms. Math. Programm. **151**(1), 249–281 (2015). doi:10.1007/s10107-015-0893-2
52. Zhou, Q., Zhang, C.: An adaptive trust region method based on simple conic models, vol. 14 (2015). doi:10.1007/s10852-015-9279-y
53. Zhu, L., Barhak, J., Srivatsan, V., Katz, R.: Efficient registration for precision inspection of free-form surfaces. Int. J. Adv. Manuf. Technol. **32**(5-6), 505–515 (2006). doi:10.1007/s00170-005-0370-9
54. Zinßer, T., Schmidt, J., Niemann, H.: A refined ICP algorithm for robust 3-D correspondence estimation. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. II–695–698. IEEE Computer Society, Barcelona, Spain (2003). doi:10.1109/icip.2003.1246775