Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Iterative Estimation of Rigid Body Transformations Application to robust object tracking and Iterative Closest Point

Micha Hersch · Aude Billard · Sven Bergmann

Received: date / Accepted: date

Abstract Closed-form solutions are traditionally used in computer vision for estimating rigid body transformations. Here we suggest an iterative solution for estimating rigid body transformations and prove its convergence. We show that for a number of applications involving repeated estimations of rigid body transformations, an iterative scheme is preferable to a closed-form solution. We illustrate this experimentally on two applications, 3D object tracking and image registration with Iterative Closest Point. Our results show that for those problems using an iterative and continuous estimation process is more robust than using many independant closed-form estimations.

Keywords Pose Estimation · Iterative Closest Point · Image registration · Rotation estimation

1 Introduction

Rigid body transformations in 3D relate different positions of a rigid object, or two different views of an object. They are thus widely used in computer vision, and the estimation of such transformations plays a major role in applications involving object localization. It is well known that three points and their image are sufficient to compute a closed-form expression for the transform. When more points (and their transform) are available, several methods have been presented in the eighties and nineties to find the best transform according to different criteria [13,14,3,27,28]. Those methods have been analyzed in [18] and compared in [9] and basic rigid-body transformation estimation is generally considered a closed research topic.

In this paper, however, we consider this problem anew and propose an iterative scheme for estimating rigid body transformation. Our point is that for some applications an iterative estimation process is advantageous over a closed-closed form solution. This is especially the case for iterative applications that require the estimation of many and putatively similar transformations, like tracking a moving object. In this case, an iterative estimation procedure can be "distributed" across the iterations of the application. Instead of having an independent estimation at each iteration of the application, we have a single iterative estimation process that spans across those iterations, ensuring a better use of the available information and a higher robustness to noise.

Iterative methods for estimating rigid body transformations have been suggested before [22,11,21]. Those methods rely on Kalman filtering, which, as pointed out in [17] does not guarantee to converge and may have some stability issues. The method we suggest is simpler and does not have such problems, as we prove that it globally converges to the least square solution. This property derives from a more adequate choice of parametrization of rotations, the so-called Rodrigues parametrization. Moreover, we show that this iterative procedure can replace closed-form solutions in image registration, thus providing a more efficient and precise algorithm.

In Section 2, we describe a novel iterative algorithm for rigid body transformation estimation called Itera-

M. Hersch, S. Bergmann
Department of Medical Genetics, University of Lausanne, CH - 1005 Lausanne and Swiss Institute of Bioinformatics, Switzerland
E-mail: {micha.hersch, sven.bergmann}@unil.ch

A. Billard
LASA Laboratory, School of Engineering, EPFL, CH - 1015 Lausanne, Switzerland
E-mail: aude.billard@epfl.ch

tive Estimator of Rigid Body Transformations (IERBT) and prove its global convergence. We then compare its use with a standard closed-form estimator on two applications, 3D object tracking and image registration (Sections 3 and 4 repectively). We conclude with a brief discussion.

2 Iterative Rigid Body Transformation Estimation

2.1 Parametrization

Any rigid body transformation can be represented as a rotation around an axis going through the origin followed by a translation. In this paper, the translation is trivially parametrized by a translation vector $\mathbf{t}$. For the rotation, a non-redundant vectorial parametrization [4] is adopted, as described in [12, pp. 277-305], also called the Rodrigues vector. Basically, the rotation is described by a vector $\mathbf{b}$ given by the first three components of its quaternion:

$$\mathbf{b} = \sin\frac{\phi}{2}\mathbf{a}, \tag{1}$$

where $\mathbf{a}$ is a unit-norm vector colinear to the rotation axis and ϕ is the rotation angle. Note that the fourth component of the corresponding quaternion can be easily retrieved as

$$\cos\frac{\phi}{2} = \sqrt{1-\|\mathbf{b}\|^2}. \tag{2}$$

Using this parametrization, a rotation $\mathbf{R_b}$ parametrized by $\mathbf{b}$ transforms a vector $\mathbf{x}$ into a vector $\mathbf{u}$ given by

$$\mathbf{u} = \mathbf{R_b}(\mathbf{x}) \tag{3}$$

$$= (1-2\mathbf{b}^T\mathbf{b})\mathbf{x} + 2\sqrt{(1-\mathbf{b}^T\mathbf{b})}\mathbf{b}\times\mathbf{x} + 2(\mathbf{b}^T\mathbf{x})\mathbf{b} \tag{4}$$

$$= (1-2\sin^2\frac{\phi}{2})\mathbf{x} + 2\cos\frac{\phi}{2}\sin\frac{\phi}{2}\mathbf{a}\times\mathbf{x} + 2\sin^2\frac{\phi}{2}(\mathbf{a}^T\mathbf{x})\mathbf{a}$$

$$= \cos(\phi)\mathbf{x} + \sin(\phi)\mathbf{a}\times\mathbf{x} + \big(1-\cos(\phi)\big)(\mathbf{a}^T\mathbf{x})\mathbf{a}, \tag{5}$$

where the last expression is the Rodrigues formula.

The vector $\mathbf{b}$ belongs to the closed ball of radius one, where antipodal point are merged, since rotations of $-\pi$ are equivalent to rotations of angle π (see [2] for details).

2.2 Estimation

The estimation problem is the following. Suppose that we have a set of n pairs of points $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^n$ such that

$$\mathbf{y}_i = \mathbf{T}^*(\mathbf{x}_i) + \epsilon_i, \tag{6}$$

where $\mathbf{T}^*$ is an unkown rigid-body transform, the aim is to estimate $\mathbf{T}^*$ so as to minimize the errors ϵ_i (in the least mean squares sense).

The iterative estimation scheme suggested here starts from an initial estimate $\mathbf{T}$ of the transform and iteratively randomly picks a pair of point $(\mathbf{x}_i,\mathbf{y}_i)$ and performs a simple gradient descent step on the corresponding residual $\|\mathbf{y}_i - \mathbf{T}(\mathbf{x}_i)\|^2$. In other words, if $\mathbf{T}(\mathbf{x}) = \mathbf{R_b}(\mathbf{x}) + \mathbf{t}$ is parametrized by vectors $\mathbf{t}$ and $\mathbf{b}$ (as described in Section 2.1), those parameters are iteratively updated by

$$\Delta\mathbf{t} = -\eta_t \cdot \mathrm{grad}_{\mathbf{t}}\frac{1}{2}\|\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\|^2 \tag{7}$$

$$\Delta\mathbf{b} = -\eta_b \cdot \mathrm{grad}_{\mathbf{b}}\frac{1}{2}\|\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\|^2, \tag{8}$$

where η_t and η_b are small learning steps. The gradient can then be computed as follows:

$$\frac{\partial}{\partial\mathbf{t}}\frac{1}{2}\|\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\|^2 = \big(\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\big)^T \tag{9}$$

$$\frac{\partial}{\partial\mathbf{b}}\frac{1}{2}\|\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\|^2 = \big(\mathbf{y}_i - (\mathbf{R_b}(\mathbf{x}_i)+\mathbf{t})\big)^T \frac{\partial}{\partial\mathbf{b}}\mathbf{R_b}(\mathbf{x}_i)$$

where $\frac{\partial}{\partial\mathbf{b}}\mathbf{R_b}(\mathbf{x}_i)$ is obtained by deriving (4) with respect to $\mathbf{b}$:

$$\frac{\partial}{\partial\mathbf{b}}\mathbf{R_b}(\mathbf{x}_i) = -2\mathbf{x}_i\mathbf{b}^T - \frac{1}{\sqrt{(1-\mathbf{b}^T\mathbf{b})}}(\mathbf{b}\times\mathbf{x}_i)\mathbf{b}^T \tag{10}$$

$$+2\sqrt{(1-\mathbf{b}^T\mathbf{b})}\mathbf{x}_i\uparrow + \mathbf{b}\mathbf{x}_i^T + (\mathbf{b}^T\mathbf{x}_i)\mathbf{I}. \tag{11}$$

In this equation, $\mathbf{I}$ is the 3×3 identity matrix and the unary operator $\uparrow$ is defined as

$$\mathbf{x}\uparrow \doteq \frac{\partial}{\partial\mathbf{b}}(\mathbf{b}\times\mathbf{x}) = \begin{pmatrix} 0 & x^{(3)} & -x^{(2)} \\ -x^{(3)} & 0 & x^{(1)} \\ x^{(2)} & -x^{(1)} & 0 \end{pmatrix}, \tag{12}$$

with $\mathbf{x} = [x^{(1)}\ x^{(2)}\ x^{(3)}]^T$. Because of the topology of the domain of $\mathbf{b}$, one must ensure that when the gradient descent takes $\mathbf{b}$ out of the ball, then $\mathbf{b}$ enters the ball from the opposite side. This can be done by multiplying $\mathbf{b}$ by $(1-2/\|\mathbf{b}\|)$ when $\|\mathbf{b}\| > 1$.

Thus the IERBT algorithm simply consists in iteratively updating an initial estimate of parameters $\mathbf{b}$ and $\mathbf{t}$, using (7) and (8) with a random pair of points $(\mathbf{x}_i,\mathbf{y}_i)$. The initial estimate may depend on the application, and short of a better guess the identity can be used. To ensure that $\Delta\mathbf{b}$ remains small, it is advised to multiply $\Delta\mathbf{b}$ by $\sqrt{(1-\mathbf{b}^T\mathbf{b})}$ and to pick η_b such that $\|\Delta\mathbf{b}\| \leq 0.01$. As for η_t, it can typically be set to 0.01.

2.3 Convergence

Proposition 1 *For a fixed set of three or more non-aligned data points $\{(\mathbf{x}_i,\mathbf{y}_i)\}$, the IERBT algorithm described in Section 2 converges to an optimal estimate in the least mean square sense.*

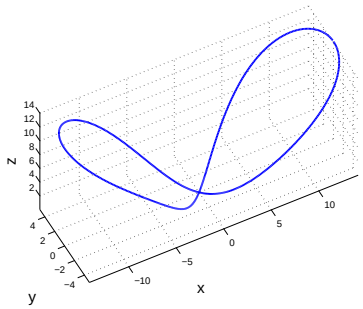

Fig. 1 The 3D trajectory used for simulating the data.

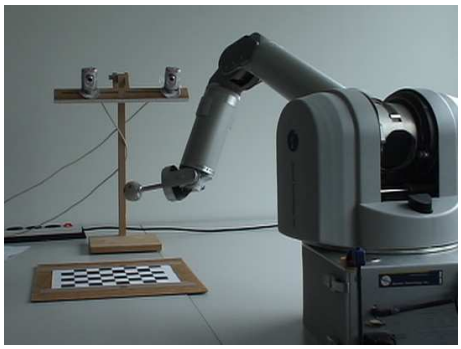

Fig. 2 The experimental setup for the tracking experiment. Two cameras track the end-effector of a WAM robot. The chessboard is used for camera calibration.

The proof is given in Appendix A.

3 Object tracking

3.1 Description

The first application of IERBT is to track a moving object, using a set of identifiable markers. We consider the simple case where marker positions are already given as 3D coordinates, for example using range images or stereovision. We are thus in the "3D to 3D correspondence" case, according to the classification given in [15]. This problem can of course also be solved with a closed-form estimator, but there are a number of reasons why the iterative procedure may be advantageous. First, the iterative scheme takes advantage of the continuity of the object trajectory as the estimate is updated by a small amount each time a marker is detected. This is likely to make IERBT less sensitive to noise than a "memoryless" closed-form solution, especially as there are only few markers. Second, and maybe more importantly, the closed-form solution assumes that all points are concomittant, i.e. all markers are tracked at the same time. If at a given time only one or two markers are localized, the closed-form cannot be applied, as opposed to IERBT. This is especially relevant in the case of occlusions. Of course, both estimation procedures can be combined, as done below.

3.2 Experiment

3.2.1 Data generation

The use of IERBT for object tracking was first investigated on simulated data. We simulated three markers on an object following a 3D lemniscate-like trajectory (drawn in Fig. 1). We considered two noise levels, a zero noise level and a Gaussian noise with $\sigma^2 = 5$. In both cases, we randomly removed 50% of the data points to simulate missing data.
To evaluate the algorithm in a real setting, we also generated data using a simple experimental setup consisting of a WAM robot and two USB cameras, as depicted in Fig 2. Three color markers were sticked to the robot end-effector which was moved within the field of view of the cameras. The marker positions were tracked using an OpenCV-based stereovision software, runing at about 5 fps. The joint trajectories of the robot were also recorded. Since the robot encoders sensing the joint positions are very precise, they were used to compute the true position and orientation of the end-effector, using the robot forward kinematics. About thirty minutes of data were recorded, amounting to more than ten thousands frames. As truth value, the trajectory of the end-effector was computed using the positions recorded by the robot.

For the simulated as well as the real data, three methods were then used to locate the end-effector from the marker positions obtained by stereovision system. The closed-form method [13] (CF), the iterative scheme (IERBT), and a combined scheme (CF+IERBT), which uses the iterative scheme only if less than three markers were tracked at a given time.

3.2.2 Results

To estimate the accuracy of the tracking, two measures were used to quantify respectively the error on the translation and on the rotation. For the translation, the mean Euclidean distance between the true and estimated translation was used. For the rotation, we use the mean Riemannian distance (in the group of rotations) between the true and the estimated rotations $\widehat{\mathbf{R}}$

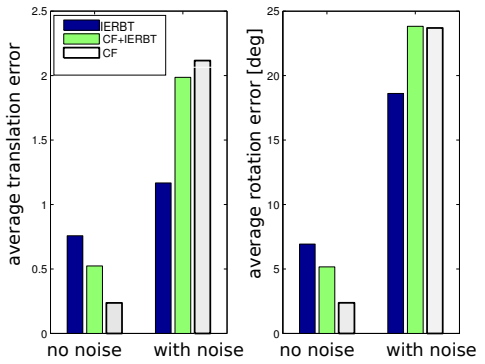

Fig. 3 The tracking accuracy on simulated data. For the estimation of the translation as well as the rotation, the closed-form method performs better without noise, but the iterative method is more accurate in the presence of noise.

method	average translation error [mm] (original data)	average rotation error [deg]
IERBT	18.89	55.84
CF	21.58	52.71
CF+IERBT	18.24	52.98

Table 1 The overall results of the real tracking experiment. For the translation, the closed-form method performs worse than the combined scheme and the iterative method. For the rotation, the iterative method performs worse than the closed-form and the combined method. Overall, the combined scheme is the most precise.

and $\mathbf{R}^*$, given by the angle of the rotation $\widehat{\mathbf{R}}^{-1}\mathbf{R}^*$ [20].

The results for the simulated data are shown in Fig. 3. One sees that for noiseless data, the closed-form solution yields more accurate estimations of translation as well as rotations. However, for noisy data, the iterative method is more accurate, indicating that this method is more robust to noise. In both cases the combined method yields intermediate results. Similar results were obtained without simulating missing data (data not shown).

For the real experiment, the data were obtained by sampling the true trajectory at 20 Hz and for each position, and compare it to the last estimated position given by the tracking algorithm. The results are summarized in Table 1. One sees that with respect to the translation, the closed-form solution (CF) does not perform as well as the iterative and the combined schemes, whereas for estimating the rotation, the iterative method (IERBT) performs worse than the two others. So in this experiment, overall, the combined method is the most precise. As illustrated in Fig. 4 at time 800-810 (solid arrow), the iterative and the combined methods can produce

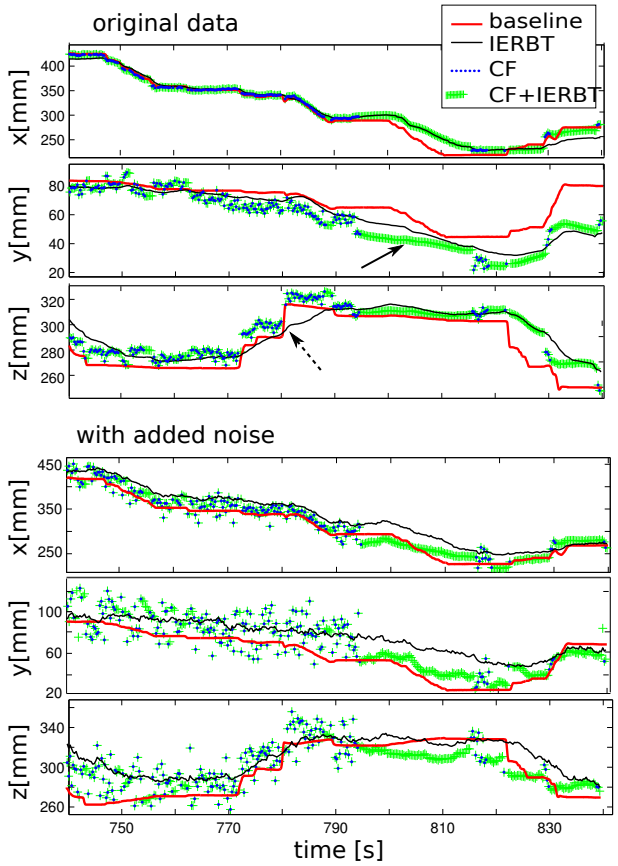

Fig. 4 Top: tracking results for a short sample of the data, showing the true end-effector position (thick line), the iterative (IERBT) estimation (thin line), the closed-form (CF) estimation (dots) and the combined (CF+IERBT) estimation (crosses). Bottom: results for the same sample, but with added Gaussian noise ($\sigma^2 = 400[mm^2]$).

meaningful estimates of the position for a given frame, even if one or two marker positions are missing, which the closed-form solution cannot. The main handicap of the iterative solution is that it can sometimes not keep up with a rapid displacement, as can be seen on Fig. 4 at time 780 (dotted arrow).
The results of Table 1 show that the errors on the estimates of the rotations, unlike the translations, are quite large for all methods, probably indicating that the markers were placed somewhat too close from one another.

3.3 Discussion

Those results show that the IERBT is a useful tool in the case of visual 3D object tracking. Its main advantages is that it does not require the simultaneous localization of three points and that it filters out the noise. Its drawback is that, like any filter, it has some "inertia"

and one must adapt the learning step to the expected speed of the object. But we have shown that using a simple combination of the iterative and closed-form estimation scheme, it is possible to significantly improve the tracking accuracy. More sophisticated combinations would probably further improve the method. In order to make the tracking robust to noise, Kalman filtering is often used with closed-form solutions. However, this assumes linear trajectories and requires the tuning of the process and measurement noises. Extended or Unscented Kalman filtering methods have also been suggested [11,21] to deal with missing data, but they also assume a known measurement noise and an appropriate motion model. Our method does not make such assumptions and is much simpler to implement and use.

4 Iterative Closest Point

4.1 Description

The second application deals with the image registration algorithm called *Iterative Closest Point* (ICP). It adresses a problem very similar to the tracking problem described above, but here the correspondance of each point across the two sets is not known and must be inferred from the data. More formally, one has two sets of points $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$ related by a rigid-body transformation. Typically, those points are obtained by sampling two surfaces related by a rigid-body transformation. The aim is to recover this transformation from the two sets of points.
The original algorithm suggested by [6] is the following:

1. Start with an estimate $\mathbf{T}$ of the transform.
2. Pair each point $\mathbf{x}_i$ with the point $\mathbf{y}_j$ that minimizes the Euclidian distance $\|\mathbf{y}_j - \mathbf{T}\mathbf{x}_i\|$. This results in a mapping $j = J(i)$.
3. Estimate $\mathbf{T}$ using a closed form solution that minimizes the sum of squared error $\sum_i \|\mathbf{y}_{J(i)} - \mathbf{T}\mathbf{x}_i\|^2$.
4. If $\mathbf{T}$ has changed, go back to 2, otherwise keep $\mathbf{T}$ as the final estimate $\widehat{\mathbf{T}}$.

Since then, many variants and improvements of this algorithms have been suggested (see [24,23]). For example, to speed up the pairing of the points, it was suggested to subsample the points [19] and to use a dedicated data structure, the *k-d tree* [5]. In [26], other metrics than the Euclidean distance were used to pair the points, to match points with similar features (e.g. curvature). The original algorithm cannot handle case of poorly overlapping data sets, i. e., if only a subset of the $\{\mathbf{x}_i\}$ has its image in the $\{\mathbf{y}_j\}$. To deal with this, a Least Median Square [19] and a Trimmed Least Squares [7] approaches were suggested, where only the best matching pairs are considered. In [10], the precision of the estimates was improved by estimating the (possibly anisotropic) noise in the data and in [1] a lookup matrix was intoduced to ensure that J is bijective. A probabilistic matching was used in [16], and more recently a version of ICP for the affine case was also presented in [8].
However, all the suggested improvements used a closed-form solution for updating the estimate $\mathbf{T}$ of the transform. The modification we suggest here is, at each iteration to pick only one point $\mathbf{x}_i$ find its counterpart $\mathbf{y}_j$ and update $\mathbf{T}$ using that single pair of points, using the iterative update rule (9,10). This way, the updates of $\mathbf{T}$ proceed in a continuous manner as it only slightly varies between each iteration. This is in contrast to standard ICP, where $\mathbf{T}$ can vary discontinuously. In the rest of this paper, we thus refer to this new version of ICP as continous ICP. Continuous ICP can thus be summarized as follows:

1. Start with an estimate $\mathbf{T}$ of the transform.
2. Randomly pick one point $\mathbf{x}_i$ and pair it with the point $\mathbf{y}_j$ that minimizes the Euclidian distance $\|\mathbf{y}_j - \mathbf{T}\mathbf{x}_i\|$.
3. Update $\mathbf{T}$ with points $\mathbf{x}_i$ and $\mathbf{y}_j$ using the iterative estimation scheme (9,10).
4. If $\mathbf{T}$ is not stationnary go back to 2, otherwise keep $\mathbf{T}$ as the final estimate $\widehat{\mathbf{T}}$.

To estimate the stationnarity of $\mathbf{T}$, the last values of $\mathbf{T}$ are kept in a buffer. It is clear that the improvements mentioned above made to standard ICP can also be applied to continuous ICP, be it for speeding up the pairing process, using more informed metrics, or trimming the data set. Therefore, in the next section we only compare standard ICP to continuous ICP, assuming that further improvements will equally affect both algorithms.

4.2 Experiments

4.2.1 Data generation

In order to estimate the possible advantages of continuous ICP, the alogrithms were compared on generated data. For each comparison, a polynomial surface of degree 4 was randomly generated and 10000 points sampled from this surface, yielding a set of points $\{\mathbf{x}_i\}$. A reference rigid body tranformation $\mathbf{T}^*$ was randomly generated and the image set $\{\mathbf{y}_i\}$ was computed as $\mathbf{y}_i = \mathbf{T}^*\mathbf{x}_i + \epsilon_i$, where ϵ_i was generated from a centered Gaussian distribution with variance σ^2. An initial estimate $\mathbf{T}$ was then also randomly generated and the algorithm was run once using standard ICP (using a

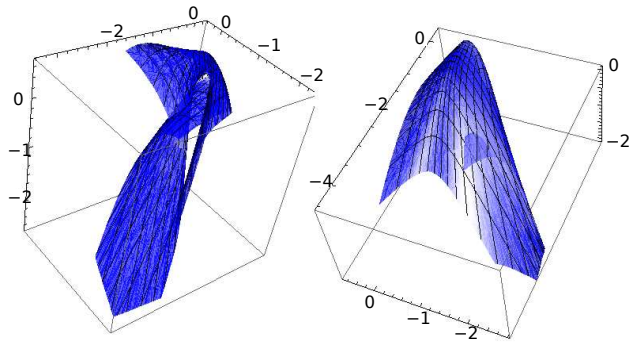

Fig. 5 Two examples of randomly generated polynomial surfaces. The coefficients of the polynomials as well as the two polynomial variables are comprised between -1 and 1.

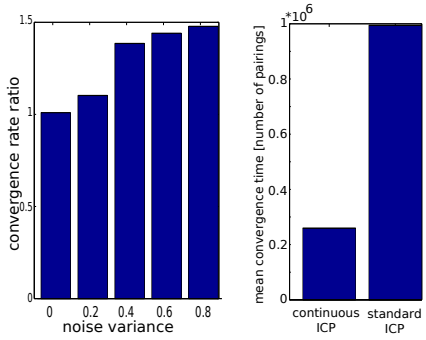

Fig. 6 Left: the ratio of the overall convergence rates for the continuous ICP over standard ICP as a function of noise level. A ratio higher than 1 indicates a better convergence rate of the continuous ICP over standard ICP. The higher the noise, the bigger is the improvement brought by continuous ICP over standard ICP. Right: The mean number of pairings needed before reaching convergence. Continuous ICP is about four times faster than standard ICP.

subsample of 6000 points at each iteration), and once using continuous ICP, yielding two estimates $\widehat{\mathbf{T}}$ of the transformation. A large number (18000 for $\sigma^2 = 0.2$ and 3000 for $\sigma^2 \in \{0, 0.4, 0.6, 0.8\}$) of such comparisons were made and statistics were gathered.

4.2.2 Results

Standard and continuous ICP were compared with respect to convergence rate and convergence time. Convergence of the algorithm was assessed by looking again at the angle of the rotation $\mathbf{R}^d = \widehat{\mathbf{R}}^{-1}\mathbf{R}^*$, where $\widehat{\mathbf{R}}$ is the final estimate of the rotation matrix (the Riemannian distance in group of rotations [20]). So if this distance as well as the distance between the estimated and true translations are below a threshold, the algorithm is assumed to have converged.

The results are presented in Fig. 6. One sees that continuous ICP has an overall better convergence rate than standard ICP. This improvement is very small if there

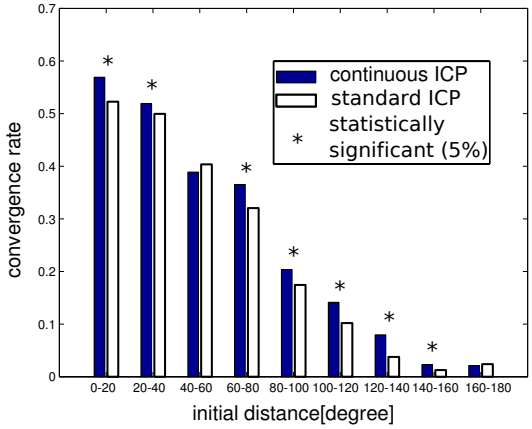

Fig. 7 The convergence rate as a function of the distance between the initial estimate of the true tranformation for a given noise level ($\sigma^2 = 0.2$). Continuous ICP performs better for good as well as for poor initial estimates.

is no noise, but it gets more and more important, as the noise increases. Moreover, continuous ICP converges about four times faster than standard ICP as it requires less pairings before reaching convergence (pairing is by far the slowest step in ICP).

Focusing on $\sigma^2 = 0.2$, our results indicate that the continuous ICP yields an increase in performance for good as well as poor initial guesses, as shown in Fig. 7. This improvement was shown to be statistically significant using a t-test ($\alpha = 5\%$).

The precision of the estimation of the rotation and translation are shown in Fig. 8. For all noise levels, continuous ICP provides, on average, a more precise estimation than standard ICP.

4.3 Discussion

The above results indicate that continuous ICP is advantageous over standard ICP in terms of precision, convergence rate and convergence time. In standard ICP, at each iteration the present estimate of the transformation is only used indirectly, through the pairings of the points, thus requiring an extensive use of those pairings to refine the estimate. Contrastingly continuous ICP makes a direct use of the current estimate, and thus the estimation can be refined using a single point. Hence no choice need to be made about the proper number of points needed for the estimation, which may explain the faster convergence.

It is beyond the scope of this paper to compare our continuous to the standard discrete estimation approach for all the many variants of ICP. In principle they could all be applied to continuous ICP as well. For example,

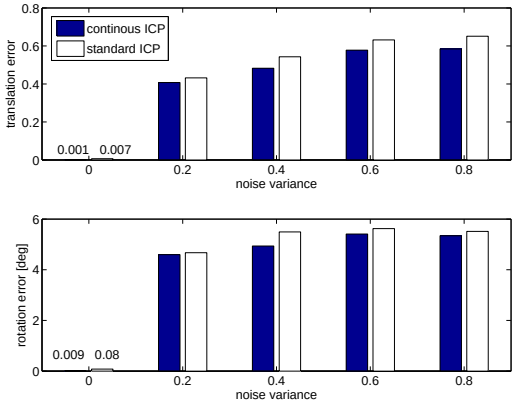

Fig. 8 The mean error of the estimates of the translation (top) and the rotation (bottom) using continuous and standard ICP. For all tested noise levels, continuous ICP converges to a more precise estimation than standard ICP, for the translation as well as for the rotation. Means where computed over all of samples where both methods converged to a correct estimate.

Trimmed ICP [7] that only uses the closest pairs for the estimation could be implemented by only performing the update step if the current pairs is among the closer of the last n visited pairs. *A priori*, there is no reason why the advantages of the continuous approach would not carry over to other variants of ICP.

5 Conclusion

Closed-form solutions are generally considered preferable to iterative solutions as they are usually more precise and require less computations. Focusing on rigid-body transformation estimation, we have shown that in some cases an iterative estimation scheme is advantageous over a closed-form estimation method. When considering iterative algorithms including repeated estimations of a rigid-body transformation, IERBT, unlike a closed-form estimate can take advantage of estimates of previous algorithm iterations. It is thus likely to produce better results. This was experimentally verified on two applications relying heavliy on the estimation of rigid-body tranformation, 3D object tracking and 3D registration using real and simulated data. Those results could only be achieved because IERBT is guaranteed to converge to the optimal estimate, as we have formally proven.
This proof also shows that for the above parametrization of rotations, the extrema in the total squared Euclidean distance between the true and the estimated transformed set of points $\sum_i(\mathbf{T}^*(\mathbf{x}_i) - \mathbf{T}(\mathbf{x}_i))^2$ correspond to the extrema of the squared Euclidean distance between the representations of the true and estimated rotations $(\mathbf{b}^* - \mathbf{b})$. This correspondance, that is not necessarily true for other representations of rotations such as the Euler angles, provides an additional reason to use the Rodrigues parametrization when using the Euclidean distance for rotation estimation as in [21,25]. It is thus probably possible to improve existing algorithms simply by changing to a Rodrigues representation of rotations.

A Proof of Convergence of IERBT

Proof Let $\mathbf{T}^*$ be the true rigid body transformation mapping a finite set of points $\{\mathbf{x}_i\} = \mathcal{V}$ into their corresponding image. If $\mathcal{V}$ contains at least three unaligned points, there is only one such transformation. Let $\mathbf{T} \neq \mathbf{T}^*$ be the current estimate of this transformation.
We then define the following function $\mathbf{E}(\mathbf{T})$

$$\mathbf{E}(\mathbf{T}) = \sum_{i=1}^{n} \mathbf{E}_i(\mathbf{T}), \text{ with } \quad \mathbf{E}_i(\mathbf{T}) = \frac{1}{2}\|\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2. \tag{13}$$

Here and in the following, the parentheses around $\mathbf{x}_i$ are omitted to lighten the notation. We first notice that for sufficiantly small η the algorithm performs a gradient descent on $\mathbf{E}$, because $\sum_i \text{grad}\mathbf{E}_i = \text{grad}\sum_i \mathbf{E}_i = \text{grad}\mathbf{E}$. As $\mathbf{E}$ is bounded, the algorithm converges to a solution.
In order to show that the algorithm converges to the right solution $\mathbf{T}^*$, we have to show that it is the only minimum of $\mathbf{E}$. So we show that for any $\mathbf{T}$, $\mathbf{T}^*$, $\mathcal{V}$, such that $\mathbf{T} \neq \mathbf{T}^*$ there is a transformation $\mathbf{T}^\dagger$ belonging to a neighborhood of $\mathbf{T}$ such that $\mathbf{E}(\mathbf{T}^\dagger) < \mathbf{E}(\mathbf{T})$. This amounts to saying that there is no local minimum for $\mathbf{E}(\mathbf{T})$, only a global one.

We assume, without loss of generality, that the $\mathbf{x}_i$ are centered and distributed with a covariance matrix $\mathbf{C}$ of rank two or three, i.e., they are not aligned.
Let the transformation $\mathbf{T}$ be defined by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$. We now consider the transformation $\mathbf{T}^\dagger$ in the neighbourhood of $\mathbf{T}$, defined by translation vector $\mathbf{t}^\dagger$ and rotation $\mathbf{R}^\dagger$ respectively in the neigbourhoods of $\mathbf{t}$ and $\mathbf{R}$.

$$\mathbf{t}^\dagger = \mathbf{t} + \epsilon(\mathbf{t}^+) \quad \mathbf{R}^\dagger = \epsilon\mathbf{R}^+ \circ \mathbf{R} \quad \text{with} \quad \epsilon > 0, \tag{14}$$

where $\epsilon\mathbf{R}^+$ is an infinitesimal rotation of unit rotation axis given by $\mathbf{b}^+$. If ϵ is small enough, we have, see [2, p.80],

$$\mathbf{R}^\dagger\mathbf{x} = \mathbf{R}\mathbf{x} + \epsilon(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}), \tag{15}$$

and we can thus define

$$\epsilon\mathbf{T}^+\mathbf{x} \doteq \mathbf{T}^\dagger\mathbf{x} - \mathbf{T}\mathbf{x} = \epsilon(\mathbf{t}^+ + \mathbf{b}^+ \times \mathbf{R}\mathbf{x}) \tag{16}$$

The fixed points of the algorithm are given by $\mathbf{T}$ for which $\mathbf{E}(\mathbf{T}^\dagger) = \mathbf{E}(\mathbf{T})$ for any $\mathbf{b}^+$. The variation in $\mathbf{E}$ when moving from $\mathbf{T}$ to $\mathbf{T}^\dagger$ is given by

$$\begin{aligned} \Delta\mathbf{E} &= \mathbf{E}(\mathbf{T}^\dagger) - \mathbf{E}(\mathbf{T}) = \sum_i \|\mathbf{T}^\dagger\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 - \sum_i \|\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 \\ &= \sum_i \|\epsilon\mathbf{T}^+\mathbf{x}_i + \mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 - \|\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 \qquad (17) \\ &= \sum_i 2\epsilon(\mathbf{T}^+\mathbf{x}_i)^T(\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i) + \epsilon^2\|\mathbf{T}^+\mathbf{x}_i\|^2 \qquad (18) \end{aligned}$$

If ϵ is small enough, we can discard terms in $\mathcal{O}(\epsilon^2)$.

$$\begin{aligned}\Delta\mathbf{E} &\simeq \sum_i 2\epsilon(\mathbf{t}^+ + \mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i + \mathbf{t} - \mathbf{t}^*) \quad (19)\\ &= 2\epsilon\Big(- n(\mathbf{t}^+)^T(\mathbf{t}^* - \mathbf{t}) + (\mathbf{t}^+)^T(\sum_i(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i))\\ &\quad + \sum_i(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i)\Big) + \sum_i(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{t}^* - \mathbf{t})\\ &= -n(\mathbf{t}^+)^T(\mathbf{t}^* - \mathbf{t}) + 2\epsilon\Big(\sum_i(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i)\Big), \quad (20)\end{aligned}$$

since the $\{\mathbf{x}_i\}$ are centered. The translation and rotation components can thus be separated and $\Delta\mathbf{E} = 0$ for all $\mathbf{t}^+$ only if $\mathbf{t} = \mathbf{t}^*$. Assuming this is the case, defining $\mathbf{B} \doteq \mathbf{b}^+\uparrow$, see (12), and using matrix representations, we can now consider the rotation component

$$\begin{aligned}\Delta\mathbf{E} &= -2\epsilon\sum_i(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T\mathbf{R}^*\mathbf{x}_i = -2\epsilon\sum_i(\mathbf{B}\mathbf{R}\mathbf{x}_i)^T\mathbf{R}^*\mathbf{x}_i \quad (21)\\ &= 2\epsilon\sum_i \mathbf{x}_i^T\mathbf{R}^T\mathbf{B}\mathbf{R}^*\mathbf{x}_i = 2\epsilon\mathrm{Tr}(\sum_i \mathbf{x}_i^T\mathbf{R}^T\mathbf{B}\mathbf{R}^*\mathbf{x}_i) \quad (22)\\ &= 2\epsilon\mathrm{Tr}(\sum_i \mathbf{B}\mathbf{R}^*\mathbf{x}_i\mathbf{x}_i^T\mathbf{R}^T) = 2n\epsilon\mathrm{Tr}(\mathbf{B}\mathbf{R}^*\mathbf{C}\mathbf{R}^T), \quad (23)\end{aligned}$$

In the above equations the "trace trick" is used, and the fact that $\mathrm{Tr}(XYZ) = \mathrm{Tr}(YZX)$.
Since the matrix $\mathbf{B}$ is skew-symmetric with a zero diagonal, the trace in (23) is zero for all $\mathbf{B}$ if and only if $\mathbf{R}^*\mathbf{C}\mathbf{R}^T$ is symmetric. Hence, as $\mathbf{C}$ has at least two non-zero eigenvalues,

$$\begin{aligned}\Delta\mathbf{E} = 0 &\Leftrightarrow \mathbf{R}^*\mathbf{C}\mathbf{R}^T = \mathbf{R}\mathbf{C}(\mathbf{R}^*)^T \Leftrightarrow \mathbf{C} = (\mathbf{R}^*)^T\mathbf{R}\mathbf{C}(\mathbf{R}^*)^T\mathbf{R} \quad (24)\\ &\Leftrightarrow (\mathbf{R}^*)^T\mathbf{R} = \pm\mathbf{I} \quad (25)\end{aligned}$$

If $(\mathbf{R}^*)^T\mathbf{R} = \mathbf{I}$, $\mathbf{R} = \mathbf{R}^*$, which corresponds to the minimum of $\mathbf{E}(\mathbf{T})$. If $(\mathbf{R}^*)^T\mathbf{R} = -\mathbf{I}$, $\mathbf{R} = -\mathbf{R}^*$, which corresponds to the maximum of $\mathbf{E}(\mathbf{T})$. Thus, we have shown that $\mathbf{E}(\mathbf{T})$ has a single minimum, which proves the convergence of our algorithm as it performs a gradient descent on $\mathbf{E}(\mathbf{T})$. □

References

1. A. Almhdie, C. Léger, M. Deriche, and R. Lédée. 3D registration using a new implementation of the ICP algorithm based on a comprehensive lookup matrix: Application to medical imaging. *Pattern Recognition Letters*, 28(12):1523–1533, 2007.
2. S.L. Altmann. *Rotations, Quaternions and Double Groups*. Oxford University Press, 1986.
3. K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987.
4. O.A. Bauchau and L. Trainelli. The vectorial parameterization of rotation. *Nonlinear Dynamics*, 32:71–92, 2003.
5. J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, pages 509–517, 1975.
6. P.J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.
7. D. Chetverikov, D. Stepanov, and P. Krsek. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.
8. S. Du, N. Zheng, S. Ying, and J. Liu. Affine iterative closest point algorithm for point set registration. *Pattern Recognition Letters*, 31(9), 2010.
9. D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformation: a comparison of four major algorithms. *Machine Vision and Applications*, 1997.
10. R.S.J. Estepar, A. Brun, and C.F. Westin. Robust generalized total least squares iterative closest point registration. *Lecture Notes in Computer Science*, pages 234–241, 2004.
11. K. Halvorsen, T. Söderström, V. Stokes, and H. Lanshammar. Using an extended kalman filter for rigid body pose estimation. *Journal of biomechanical engineering*, 127:475, 2005.
12. D. Hestenes. *New Foundations for Classical Mechanics*. Fundamental Theories of Physics. Kluwer Academic Publishers, 2 edition, 1999.
13. B.K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–641, 1987.
14. B.K. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, pages 1127–1135, 1988.
15. T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.
16. H. Hufnagel, X. Pennec, J. Ehrhardt, N. Ayache, and H. Handels. Generation of a statistical shape model with probabilistic point correspondences and the expectation maximization-iterative closest point algorithm. *International Journal of Computer Assisted Radiology and Surgery*, 2(5):265–273, 2008.
17. SH Joseph. Optimal Pose Estimation in Two and Three Dimensions* 1. *Computer Vision and Image Understanding*, 73(2):215–231, 1999.
18. K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–549, 1994.
19. T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-D model construction. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 1, 1996.
20. M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002.
21. M. Hedjazi Moghari and P. Abolmaesumi. Point-based rigid-body registration using an unscented kalman filter. *IEEE Transactions on Medical Imaging*, 26(12):1708–1728, 2007.
22. J. Porrill, SB Pollard, and JEW Mayhew. Optimal combination of multiple sensors including stereo vision. *Image and vision computing*, 5(2):174–180, 1987.
23. S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of 3D Imaging and Modeling*, pages 145–152, 2001.
24. J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.
25. R. Sandhu, S. Dambreville, and A. Tannenbaum. Point Set Registration Via Particle Filtering and Stochastic Dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
26. G.C. Sharp, S.W. Lee, and D.K. Wehe. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 90–102, 2002.
27. M.W. Walker, L. Shao, and R.A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 1991.
28. Z. Wang and A. Jepson. A new closed-form solution for absolute orientation. In *Proceedings of of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 129–134, 1994.