3D-3D Rigid Registration: A Comparative Analysis Study on Femoral Bone Scans

Perrine Solt, Adlane Habed, Antoine Bautin, Pierre Maillet, and Michel de Mathelin

Abstract 3D point cloud registration is an important step in a variety of computer-assisted surgeries and particularly critical for their success. Such registration is traditionally carried out using the Iterative Closest Point (ICP) algorithm although more recent and promising algorithms have been reported in the literature. In this paper, we provide a comparative analysis of several rigid registration algorithms for 3D point clouds including point-to-point ICP, point-to-plane ICP, Go-ICP and Super4PCS, in the difficult context of bone registration. In particular, we study the case in which a point cloud of femoral condyles is to be registered at the distal extremity of a human femoral bone. The condyles can typically be acquired during surgery using 3D sensors (e.g. intraoperative CT-scans, time-of-flight cameras, structured-light scanners) while the bone can be scanned preoperatively. The algorithms have been tested for speed, accuracy and robustness using both real and simulated data.

Keywords Point cloud registration · Femoral bone · ICP · Go-ICP · Super4PCS

P. Solt (✉) · A. Habed · M. de Mathelin
CNRS, University of Strasbourg, Paris, France
e-mail: Perrine.solt@etu.unistra.fr

A. Habed
e-mail: habed@unistra.fr

M. de Mathelin
e-mail: demathelin@unistra.fr

P. Solt · A. Bautin · P. Maillet
Zimmer Biomet Robotics, Montpellier, France
e-mail: antoine.bautin1@zimmerbiomet.com

P. Maillet
e-mail: pierre.maillet@zimmerbiomet.com

R. Su et al. (eds.), *Medical Imaging and Computer-Aided Diagnosis*, Lecture Notes in Electrical Engineering 810, https://doi.org/10.1007/978-981-16-6775-6_21

1 Introduction

In computer-assisted surgery, imaging modalities (e.g. CT-scans, MRI, X-Ray) and computer technologies provide the surgeon with a 3D representation of the surgical region of interest. This not only allows for a precise navigation during surgery, but also for the preoperative planning of the surgical procedure. However, for planning to be any useful intraoperatively, proper registration of the patient and the preoperative surgical data in a single reference frame is required. In essence, such registration consists in aligning a partial and noisy intraoperative 3D source point cloud on the complete and precise preoperative 3D model point cloud.

When the navigation system or the surgical robotic device is rigidly attached to the patient, registration may only be needed once, as an initial step. This is generally achieved using bone fiducials or inserts [6]. In this case the registration time, though important to reduce the overall duration of surgery, is not critical. However, when the patient moves during surgery, registration is to be renewed, in real time, of the order of a few tens of milliseconds, as to keep the patient and the planning (hence the preoperative 3D model) reference frames aligned at all times.

Most available commercial robotic and navigation systems use optical trackers that are rigidly fixed (e.g. with pins or screws) to the patient's bones so as to track their movements using an optical device [3, 19, 20, 22, 23]. This makes the tracking problem quite straightforward. Unfortunately, attaching the trackers to the bones is quite invasive and may lead to infections and/or fractures in addition to longer surgical procedures and healing time. RGB-D cameras offer a promising noninvasive alternative to the existing optical tracking systems. Indeed, such cameras are inexpensive and provide a relatively accurate 3D shape that may facilitate tracking and avoid using markers altogether. As a result, several works in progress investigate the use of RGB-D cameras for tracking [7–10, 17] in order to overcome the drawbacks of existing systems. Tracking then boils down to registering the 3D intraoperatively scanned bone portions on a preoperative 3D bone model. However, this requires the registration process to be fast as to allow real-time tracking. Furthermore, registration accuracy, robustness to noise as well as to different amounts of bone motion are also required for a safe and precise surgical act. The context of femoral bone surgery, which is the subject of the current study, is particularly difficult and challenging for such registration because only a small portion of the bone is visible during surgery [7].

This paper provides a comparative analysis study of 3D-3D rigid registration algorithms of bones in the context of knee surgery. In particular, we focus on the application described by Liu and Baena in [10] and in which femoral condyles are scanned with a RGB-D camera and require intraoperative registration on a preoperatively scanned femoral bone point cloud. This is a challenging registration procedure with real-time requirement and in which only a portion of the bone is visible during surgery. In this regard, we compare the viability and efficiency of the well-known and widely used Iterative Closest Point Algorithm (ICP) in its point-to-point [2] and point-to-plane [11] variants against two more recent and promising algorithms:

Super 4-Points Congruent Sets (Super4PCS) [12] and Globally Optimized ICP (Go-ICP) [21]. Super4PCS is based on the 4PCS procedure: it computes the best alignment in the least squares sense by finding coplanar bases of 4 congruents pointsets within a RANSAC (RAndom SAmple Consensus) procedure. In contrast, Go-ICP is based on a branch-and-bound search using ICP as a subroutine. To test these algorithms in the context of the chosen application, we propose a workflow to simulate bone movement and evaluate the registration accuracy, while applying increasing perturbations to the scanned femoral condyles.

Our paper is organized as follows: the targeted registration algorithms are presented in Sect. 2. Section 3 describes the material and methods used to compare the registration algorithms. The results of our experiments along with our analysis are given in Sect. 4. Section 5 concludes this work and provides future works.

2 Registration Algorithms

This section provides a review of the four 3D-3D rigid point cloud registration algorithms considered in our comparative analysis study: namely, point-to-point ICP, point-to-plane ICP, Super4PCS and Go-ICP.

Point-to-Point ICP: The ICP algorithm from Besl and McKay [2] is the most commonly used algorithm for solving the 3D-3D registration problem and has many variants [16]. The algorithm attempts to register the two point clouds by iteratively considering the closest points (in the Euclidean sense) in the source and model point clouds as corresponding ones. For S a source point cloud containing N points with $\mathbf{s}_i = (s_{ix}, s_{iy}, s_{iz}, 1)^T$ a source point, and $\mathbf{m}_i = (m_{ix}, m_{iy}, m_{iz}, 1)^T$ the closest model point, at each iteration, the ICP algorithm estimates the optimal 4 × 4 rigid transformation matrix $\mathbf{M}_{opt}$ by solving

$$\mathbf{M}_{opt} = \arg\min_{M} \sum_{i=0}^{N-1} \|\mathbf{m}_i - \mathbf{M}\mathbf{s}_i\|^2. \tag{1}$$

The estimated transformation is then applied to the source points and the process of solving (1) is repeated.

Point-to-Plane ICP: While the point-to-plane variant of the ICP proceeds in the same iterative way as its point-to-point counterpart, it differs from it in that it considers, as a minimization metric, the distance between each source point and the tangent plane at its closest corresponding model point [11]. This allows one to take advantage of the surface normal information and was observed to generally lead to more robust and accurate registration results. Using the same notation as for the point-to-point ICP, and with $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz}, 0)^T$ the unit normal vector at point $\mathbf{m}_i$ from the model, the optimal transformation is obtained by solving:

$$\mathbf{M}_{opt} = \arg\min_{M} \sum_{i=0}^{N-1} ((\mathbf{m}_i - \mathbf{M}\mathbf{s}_i)^T \mathbf{n}_i)^2. \tag{2}$$

Super4PCS: The 4PCS procedure consists in three steps [1]: creating some wide 4-points coplanar base, searching for all congruent bases and finding the most appropriate one. First, to create a coplanar base, three points, say p_1, p_2 and p_3, are randomly selected and a fourth point, p_4, is selected on the plane defined by the first three points. The size of this wide base is conditioned by the *overlap* value, set by the user. This value defines the proportion of common points in the point clouds. Then, congruent base points are extracted. For a rigid alignment, two distances are computed from the base as invariants. As it is always possible to find two intersecting lines between the four coplanar points, let set p_1p_2 and p_3p_4 the lines intersecting in a point p_5. The two invariants are defined by the ratios:

$$r_1 = \| p_1 - p_5\| / \| p_1 - p_2\| \tag{3a}$$

$$r_2 = \| p_3 - p_5\| / \| p_3 - p_4\| \tag{3b}$$

All the bases having the same invariants, up to a user-defined approximation level δ, are selected. Finally, the best aligning transformation is sought within a RANSAC procedure. The chosen base is the one having the largest number of points within δ distance from model points. In order to deal with the quadratic time complexity, Super4PCS removes the redundant 4-points candidates by using a rasterization approach.

Go-ICP: Go-ICP uses a nested branch-and-bound (BnB) structure together with the point-to-point ICP minimization problem (1). The BnB structure consists in splitting the search intervals using a tree structure, and evaluating candidate solutions by comparison with lower and upper estimated bounds. In this case, the outer BnB loop explores the rotation space, whereas the inner one explores the translational component of the rigid transformation. The algorithm is based on defining a progressively tight underestimator of the globally optimal registration error within a parameter space interval. While this corresponds to the most optimistic registration cost, the most pessimistic one is provided by the traditional local point-to-point ICP. Clearly, when an optimistic cost is worse than the pessimistic one, its corresponding parameter interval may be safely dropped. This makes the algorithm globally convergent and guarantees global optimality (up to a predefined optimality threshold).

Three main parameters of Go-ICP can be set by the user: the MSE threshold defining the convergence threshold based on the mean of squared errors, the trimming factor used to manage outliers, and the size of the distance transform used to compute the closest distances for bound evaluation.

3 Material and Methods

3.1 *Material and Preprocessing*

For this study, we use one right femur 3D model extracted from the database provided by Nolte et al. [14]. A preprocessing of this 3D bone model is applied. It consists in placing the bone in a well-defined femoral coordinate system, in defining a region of interest (ROI) and in upsampling the point cloud.

First, some studies propose methods to determine the anatomical axis of the femur and the tibia [4, 13]. For our purpose, we use the following definition (see Fig. 1):

- The X axis is defined as the epicondylar axis, going through both epicondylar points and oriented towards the right side of the patient, i.e. laterally for the right femur and medially for the left femur.
- The Y axis is placed on the antero-posterior axis, also named Whiteside line. It is oriented from the posterior to the anterior side.
- The Z axis is oriented from the intersection between X and Y axis towards the center of the femoral head.

Then, in order to decrease the computation time, a ROI is defined by keeping only the 10 cm femoral distal part of the point cloud. This is realistic since the part of the bone that is scanned during surgery is on the incision site.

Finally, point clouds are upsampled to increase and uniformize the point cloud density of the model. This is necessary because the fitness score computed by the

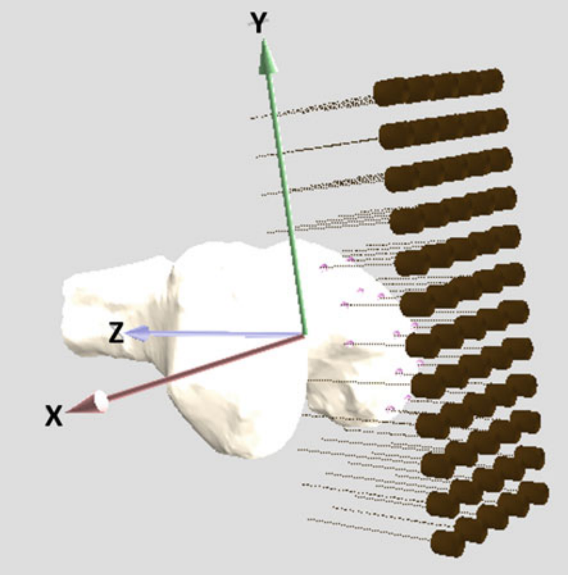

Fig. 1 Femoral coordinate system and femoral condyles point cloud generation process. Only a half matrix of transducers with low density (d = 10 mm) is presented for visualization ease

ICP algorithms depends on the point cloud density. For this purpose we use the Poisson-disk sampling algorithm presented by Corsini et al. [5] and implemented in Meshlab to create a Poisson density with a mean distance of 0.4 mm. After this preprocessing, the model point cloud contains 63,641 points.

3.2 Simulation Workflow

We propose an implementation of a simulation workflow. It aims at quantifying the registration error of the point cloud of the condyles on the preprocessed bone model point cloud during bone movement in real time. It consists of 4 steps: leg movement simulation, femoral condyles point cloud generation, registration and error quantification (see Fig. 2).

Leg Movement Simulation: The movement of the leg (Fig. 2a) is simulated through random transformations of the preprocessed bone model. These random transformations are generated with a Gaussian distribution of μ_r mean Euler angle with σ_r standard deviation, and μ_t mean translation with σ_t standard deviation. We denote by $M_{applied}$ the resulting 4×4 rigid transformation matrix. Different leg movement speeds are simulated by varying the mean value of these transformations. Note that, at each trial, the algorithms are fed the same data set obtained by applying a generated transformation.

Femoral Condyles Point cloud Generation: The point cloud of the femoral condyles is modeled by the following method (see Fig. 1). Each sensor is represented by a matrix of $n \times m$ transducers. A raycast is generated from each transducer. All the intersections between the raycasts and the bone model mesh are collected and account for the condyle point cloud. Each condyle point belongs to the bone mesh, but is not necessarily included in the bone point set.

We propose to simulate different point cloud densities to account for various resolutions. This is done by varying the distance d (in mm) between two transducers (e.g. between two rays). We also add perturbation to this point cloud through Gaussian noise of mean value μ_{noise} and standard deviation σ_{noise}. After this step, the source

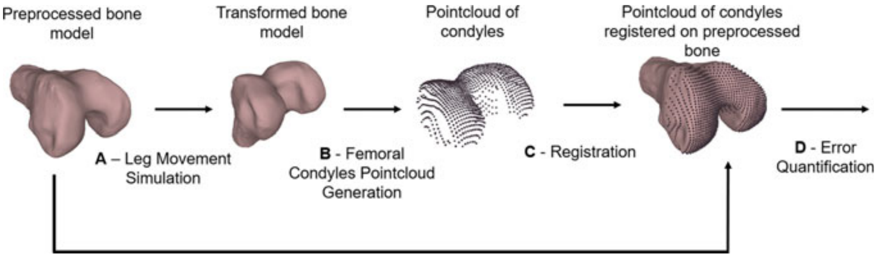
Preprocessed bone model
Transformed bone model
Pointcloud of condyles
Pointcloud of condyles registered on preprocessed bone
A – Leg Movement Simulation
B - Femoral Condyles Pointcloud Generation
C - Registration
D - Error Quantification

Fig. 2 Simulation workflow to compare registration error for different registration algorithms

point cloud is composed of 691 points for $d = 2\,\text{mm}$, 110 points for $d = 5\,\text{mm}$ and 2747 points for $d = 1\,\text{mm}$.

Registration: The generated femoral condyle point cloud is registered (Fig. 2c) on the bone model point cloud using one of the registration algorithms described in Sect. 2.

Error Quantification: In order to quantify the registration error (Fig. 2d), we use a RMSE with the following definition.

For S a source point cloud containing N points, with $\mathbf{s}_i = (s_{ix}, s_{iy}, s_{iz}, 1)^T$ a source point and $\mathbf{M}_{est}$ the 4×4 rigid transformation matrix estimated by the registration algorithm as the inverse of $\mathbf{M}_{applied}$ to align the transformed source on the model, the RMSE is obtained by:

$$\mathbf{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} \|s_i - \mathbf{M}_{est}\mathbf{M}_{applied}s_i\|^2} \tag{4}$$

This value differs from the standard RMSE error internally used by the registration algorithms and also named fitness score [2]. In fact, the fitness score compares the distance from each point of the registered source to its nearest neighbor in the model. The RMSE instead compares the position of each point of the registered source (after a transformation has been applied) $\mathbf{M}_{est}\mathbf{M}_{applied}s_i$ to the same point s_i from the initially well-aligned source.

This metric has two advantages: it is not influenced by the quality of the matching between both model and registered source point clouds and it does not depend on the model point cloud density. But the metric can only be used if the initially applied transformation is known, which is the case in our experiments.

3.3 Preprocessing

A point cloud preprocessing is applied before registration. First the model and the source point clouds are centered. For both point-to-point and point-to plane ICP, the parameters are computed so that the model is centered at the origin, and the same parameters are applied to the source point cloud. For Super4PCS and Go-ICP, we use an independent centralization: both source and model point clouds are centered at their respective origins. In all cases, a normalization is then applied such that the model lies within the unit-radius sphere. Both model and source point clouds are scaled with the same scale factor. Finally, the point indices in the model and source point clouds are randomized.

4 Experiments and Results

The simulations were carried out on a laptop with an Intel i7-11850H 2.50GHz CPU, Ubuntu 20.04, Oracle VM VirtualBox.

4.1 Parametrization of the Registration Algorithms

Point-to-Point and Point-to-Plane ICP: Point-to-Point and Point-to-plane ICP were tested using their respective Point Cloud Library's (PCL v1.12 [18]) implementations. These are based on solving (1) and (2) using Singular Value Decomposition.

Parametrization of Super4PCS: Super4PCS was tested using its OpenGR library implementation [15]. Particular attention was paid to the setting of following parameters:

- The $overlap$ parameter defines the overlap ratio between the source and the model point clouds, with respect to surface area of the smallest point cloud. The number of trials of different 4-point sets bases is directly linked to this value: the larger $overlap$, the less trials. The femoral condyles being totally included in the bone point cloud, the real $overlap$ value is 1. In our experiments, the $overlap$ was set to 0.9 because in practice some points may not necessarily belong to the bone. As a consequence, the number of RANSAC iterations naturally increases.
- The parameter δ has an influence on the process of extraction of the pairs of points and in the search of congruent bases. With a larger δ, more pairs of points are considered to have the same invariants and more congruent bases are evaluated. For δ too small, the number of retrieved potential correspondences may be insufficient and the algorithm fails to converge. With a much bigger δ than it ought to be, many more possibilities are evaluated and the algorithm may take prohibitively longer to terminate. In our experiments δ was set to 1 mm and then normalized with the same scale factor used during the normalization process described in Sect. 3.3.
- Super4PCS relies on a sparse matching of points across point clouds. It is hence essential, as it is also recommended in the OpenGR library documentation, to use a reasonable sample size limited to a few thousands of points. We used a sample size of 3000 points.
- We set 60 s as a time limit as it has been proposed for some test data provided by Mellado et al. [12] along with their software.

As suggested by the authors, the registration output of Super4PCS is fed into an ICP refinement.

Parametrization of Go-ICP: Some parameters require proper setting by the user for the Go-ICP algorithm.

- The $trimming$ factor is meant to manage outliers by excluding extreme values. We set it to 0.

- Go-ICP's runtime depends on the convergence threshold (global optimality gap). A compromise is hence needed between time and accuracy. We set the optimality threshold to 0.001 mm. A smaller value slows the algorithm down but increases the registration accuracy.
- The number of nodes per dimension of the distance transform is used to compute the closest distances for fast bound evaluation. We set this value to 30.

Description of the experiments: We conducted four experiments. For each of them, we were interested in the mean registration time (Fig. 3), in the convergence (Fig. 4) and in the mean of the RMSE value (Fig. 6) over 100 trials.

Both point clouds were initially aligned. For the two first experiments, we were interested in the ability of the algorithms to converge while increasing the bone movement amplitude for a fixed time difference. This movement is defined as a transformation composed of a rotation and a translation. Except for the third experiment, the matrix of transducers used to generate the femoral condyle points is of size of 100 × 120 boxes, with $d = 2$ mm. All the distances, for applied translations and distance between transducers, are defined in mm before being normalized with the same scale factor used during the normalization process described in Sect. 3.3.

- First, we increased the translation while not applying any rotation. μ_t was taken in the range between -55 and 55 mm with a step of 10 mm along the three axes X, Y and Z, with a fixed $\delta_t = 5$ mm.
- We then assessed the robustness of the algorithms while increasing the rotation with a small realistic Gaussian translation ($\mu_t = 5$ mm, $\sigma_t = 5$ mm).
- In the third experiment, we focused on the influence of the dimensions of the matrix of transducers on the registration convergence. We increased d to change the density of the source point cloud. Gaussian rotations ($\mu_r = 10°$, $\sigma_r = 5°$) and translations ($\mu_t = 10$ mm, $\sigma_t = 5$ mm) were applied.
- Finally, we assessed the impact of noise on each registration algorithm. For this purpose, we increased σ_{noise} while keeping $\mu_{noise} = 0$ mm. Gaussian rotations ($\mu_r = 10°$, $\sigma_r = 5°$) and translations ($\mu_t = 10$ mm, $\sigma_t = 5$ mm) were applied.

4.2 Registration Time

Figure 3a shows that Go-ICP and Super4PCS were quite slow (between 15 and 25 s) in comparison to the local algorithms point-to-point and point-to-plane ICP (less than 4 s) in case of variation of translations. For translation values between -15 mm and 15 mm (resp. -5 mm and 5 mm), point-to-point ICP converged on average in 0.26 s (resp. 0.18 s), and point-to-plane ICP converged on average in 0.13 s (resp. 0.08 s).

It can be seen in Fig. 3b that, by reducing the distance d between the center of each transducer (i.e. by increasing the source point cloud density), the registration time increased exponentially for Go-ICP, while it did not increase much for both local ICP variants.

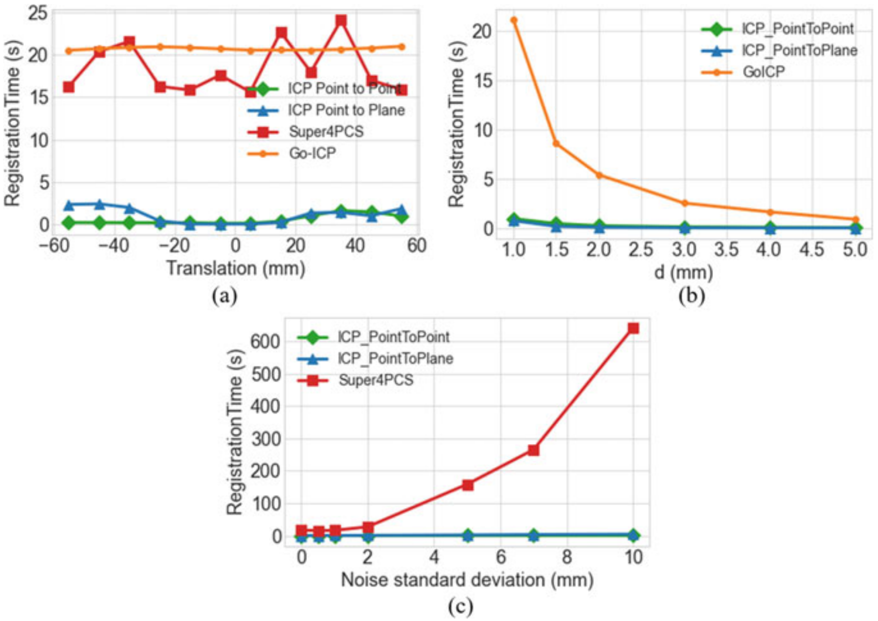

Fig. 3 Registration time for: **a** a variation of the translation applied to the source point cloud, **b** a variation of the distance between each transducer, **c** a variation in the applied noise standard deviation

While applying increasing noise to the source point cloud (see Fig. 3c), the registration time did not increased much for point-to-point ICP: typically 0.24 s of mean time in the absence of noise and 0.58 s in the presence of noise with a standard deviation of 10 mm. Point-to-plane ICP registration time increased from 0.17 s with no noise to 4.61 s for noise with 10 mm standard deviation. Super4PCS was a lot slower with a mean registration time of 17.2 s without noise and 642 s for noise with 10 mm standard deviation. The simulations with Go-ICP were not carried out to the end because, for each trial, the time exceeded 95 s for a noise standard deviation of 2 mm and to more than 2 h for noise with 5 mm standard deviation.

4.3 Convergence

We are interested in the convergence of the different algorithms. We consider that a run converges if the RMSE value is less than 5 mm. This value has been chosen by considering the histograms from Fig. 4, with a bin size of 5 mm. It separates the cluster of points with a RMSE close to 0 from the rest of the clusters of points with high RMSE values, representing point clouds that have been wrongly registered.

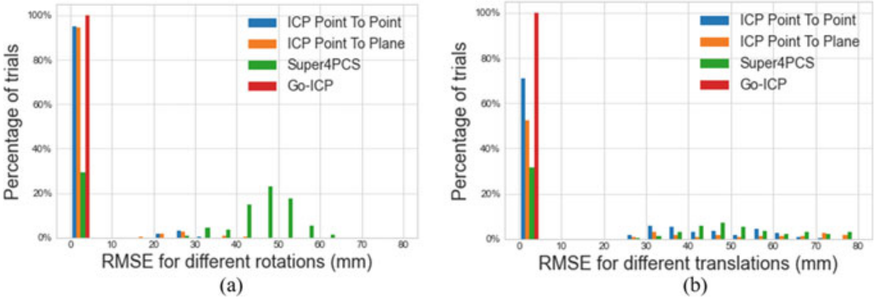

Fig. 4 Histogram of RMSE for: **a** a variation in the rotation, **b** a variation in the translation

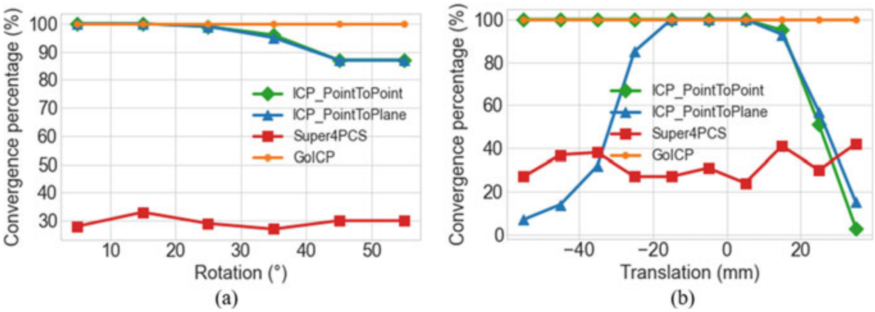

Fig. 5 Convergence percentage for: **a** a variation in the rotation, **b** a variation in the translation

In both cases of rotation variation (Fig. 5a) and translation variation (Fig. 5b), Super4PCS converged only in about 30% of the cases with our parametrization. Go-ICP always converged. Both local ICP algorithms increasingly failed to converge for rotations of more than 20° and for translations exceeding than 10 mm. Such failures are likely due to a premature convergence to a local optimum.

4.4 Registration Accuracy

Considering only the cases where the algorithms converged, Super4PCS exhibited the same behavior as that of Go-ICP and point-to-plane ICP when increasing the rotations (see Fig. 6a), with a RMSE error above 0.3 mm, while point-to-plane ICP outperformed all algorithms with a RMS error of 3.3E−3 mm. For translations (see Fig. 6b), Super4PCS performed better than point-to-point ICP and Go-ICP when it converged. No values for translations more than 35 mm are shown because point-to-point ICP failed to converge all the time. To explain the shift in the convergence observed in Fig. 6b, we present in Fig. 6c the variation of the RMSE for the point-to-point ICP algorithm by decomposing the applied translations into each Euclidean axis. We suspect the shape of the bone point cloud to be the cause of the behavior observed on the Z axis, and to induce the previously mentioned shift. Figure 6d shows the mean RMS error for all the algorithms with non-converging cases included.

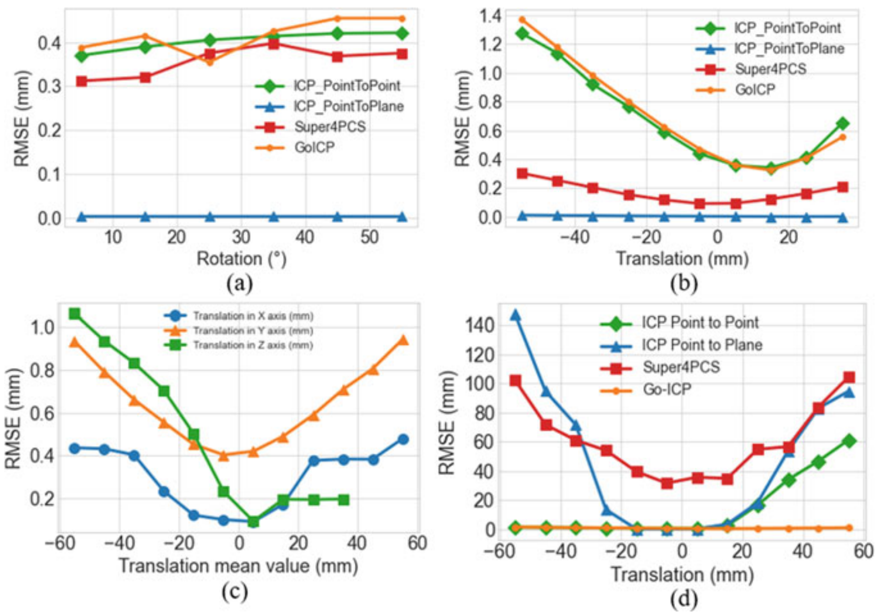

Fig. 6 Registration accuracy for: **a** a variation of the only rotation in the case of convergence, **b** a variation in only the translations in the case of convergence, **c** a variation of the translation in Euclidean axes only in case of convergence, **d** a variation in the translations for all cases (i.e. with and without convergence)

5 Conclusion and Future Work

We conducted a comparative analysis study of four 3D-3D rigid registration algorithms—point-to-point ICP, point-to-plane ICP, Super4PCS and Go-ICP—in the context of knee tracking with a 3D camera. We tested the registration robustness to the amplitude of the leg movement (for increasing transformations), to noise and to the density of the source point cloud. Our study has shown that point-to-plane ICP is the most adequate to guarantee convergence despite a variation in the source point cloud density and noise, while being fast. The condition for that is that the movement amplitude stays small, thus inducing small motions between consecutive point clouds. This is realistic because of the high frequency of data acquisition in real-time tracking. Nevertheless, our study shows that there is a need for registration algorithms that are more suitable for real-time applications. Dedicated solutions, such as local point cloud descriptors, specific to the bone local shape, should be investigated.

References

1. Aiger, D., Mitra, N., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. 35th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'08) 27 (08 2008)
2. Besl, P., McKay, H.: A method for registration of 3-D shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 14, 239–256 (1992)
3. Brainlab: Knee3 surgical technique p. 58 (2015), https://www.brainlab.com/wp-content/uploads/2016/12/Knee3-Surgical-Technique.pdf
4. Chen, H., Kluijtmans, L., Bakker, M., Dunning, H., Kang, Y., van de Groes, S., Sprengers, A.M.J., Verdonschot, N.: A robust and semi-automatic quantitative measurement of patellofemoral instability based on four dimensional computed tomography. Medical Engineering & Physics 78, 29–38 (2020)
5. Corsini, M., Cignoni, P., Scopigno, R.: Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. IEEE transactions on visualization and computer graphics 18, 914–24 (2012)
6. Fitzpatrick, J.M.: The Role of Registration in Accurate Surgical Guidance. Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine 224(5), 607–622 (2010)
7. He, G., Mustahsan, V.M., Bielski, M.R., Kao, I., Khan, F.A.: Report on a novel bone registration method: A rapid, accurate, and radiation-free technique for computer- and robotic-assisted orthopedic surgeries. Journal of Orthopaedics 23, 227–232 (2021)
8. Hu, X., Liu, H., Rodriguez y Baena, F.: Markerless Navigation System for Orthopaedic Knee Surgery: A Proof of Concept Study. IEEE Access 9 (Apr 2021)
9. Hu, X., Nguyen, A., Baena, F.R.y.: Occlusion-robust visual markerless bone tracking for computer-assisted orthopedic surgery. IEEE Transactions on Instrumentation and Measurement 71, 1–11 (2022)
10. Liu, H., Baena, F.R.Y.: Automatic Markerless Registration and Tracking of the Bone for Computer-Assisted Orthopaedic Surgery. IEEE Access 8, 42010–42020 (2020)
11. Low, K.L.: Linear least-squares optimization for point-to-plane ICP surface registration (01 2004), https://www.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf
12. Mellado, N., Aiger, D., Mitra, N.J.: Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. Computer Graphics Forum 33(5), 205–215 (2014)
13. Miranda, D.L., Rainbow, M.J., Leventhal, E.L., Crisco, J.J., Fleming, B.C.: Automatic Determination of Anatomical Coordinate Systems for Three-Dimensional Bone Models of the Isolated Human Knee. Journal of biomechanics 43(8), 1623–1626 (2010)
14. Nolte, D., Tsang, C.K., Zhang, K.Y., Ding, Z., Kedgley, A.E., Bull, A.M.: Non-linear scaling of a musculoskeletal model of the lower limb using statistical shape models. Journal of Biomechanics 49(14), 3576–3581 (2016)
15. OpenGR (Aug 2022), https://github.com/STORM-IRIT/OpenGR, original-date: 2018-04-23T06:36:52Z
16. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing ICP variants on real-world data sets Open-source library and experimental protocol. Autonomous Robots 34(3), 133–148 (2013), publisher: Springer
17. Rodrigues, P., Antunes, M., Raposo, C., Marques, P., Fonseca, F., Barreto, J.P.: Deep segmentation leverages geometric pose estimation in computer-aided total knee arthroplasty. Healthcare Technology Letters 6(6), 226–230 (2019)
18. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China (May 9–13 2011)
19. Smith&Nephew: Navio surgical system, surgical technique for total knee arthroplasty p. 44 (2018), https://www.smith-nephew.com/global/surgicaltechniques/recon/14529%20v1%20500095%20revc%20navio%20tka%20surgical%20technique%200718.pdf

20. Stryker: Mako®pka, application user guide p. 108 (2018), https://www.strykermeded.com/media/2943/mako-pka-application-user-guide-pka-30-003.pdf
21. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(11), 2241–2254 (Nov 2016), arXiv:1605.03344 [cs]
22. ZimmerBiomet: Othosoft®, knee 2.4 universal p. 66, https://www.zimmerbiomet.com/content/dam/zimmer-biomet/medical-professionals/knee/optical-navigation-system/orthosoft-knee-2-4-universal-user-manual.pdf
23. ZimmerBiomet: Rosa®knee system, surgical technique v1.1 p. 60 (2020), https://www.rosaknee.com.br/assets/pdfs/ROSA%20Knee%20Surgical%20Technique%20v1.1%20(ingles).pdf