Comparing Seven Methodologies for Rigid Alignment of Point Clouds with Focus on Frame-to-Frame Registration in Depth Sequences

Fernando A. de A. Yamada
Gilson A. Giraldi
National Laboratory for Scientific Computing
Petrópolis, Brasil
akio@lncc.br
gilson@lncc.br

Marcelo B. Vieira
Liliane R. de Almeida
Federal University of Juiz de Fora
Juiz de Fora, Brasil
marcelo.bernardes@ufjf.edu.br
lili.rodrigues.ufjf@gmail.com

Antonio L. Apolinário Jr.
Federal University of Bahia
Salvador, Brasil
antonio.apolinario@ufba.br

***Abstract*—Pairwise rigid registration aims to find the rigid transformation that best registers two surfaces represented by point clouds. This work presents a comparison between seven algorithms, with different strategies to tackle rigid registration tasks. We focus on the frame-to-frame problem, in which the point clouds are extracted from a video sequence with depth information generating partial overlapping $3D$ data. We use both point clouds and RGB-D video streams in the experimental results. The former is considered under different viewpoints with the addition of a case-study simulating missing data. Since the ground truth rotation is provided, we discuss four different metrics to measure the rotation error in this case. Among the seven considered techniques, the Sparse ICP and Sparse ICP-CTSF outperform the other five ones in the point cloud registration experiments without considering incomplete data. However, the evaluation facing missing data indicates sensitivity for these methods against this problem and favors ICP-CTSF in such situations. In the tests with video sequences, the depth information is segmented in the first step, to get the target region. Next, the registration algorithms are applied and the average root mean squared error, rotation and translation errors are computed. Besides, we analyze the robustness of the algorithms against spatial and temporal sampling rates. We conclude from the experiments using a depth video sequences that ICP-CTSF is the best technique for frame-to-frame registration.**

***Index Terms*—Rigid registration; Iterative Closest Point; Frame-to-Frame Registration, Depth Images, Rotation Error Metric, Gaussian Mixture, Tensor Shape Descriptors.**

I. Introduction

Surface registration is a common computer vision problem, with applications in computer graphics, robotics, quality inspection, photogrammetry, augmented reality, pose estimation, among others [1]. Rigid registration is a sub-problem, dealing only with sets that differ by a rigid motion. In this problem, given two point clouds, named source set $P = \{\mathbf{p}_i | \mathbf{p}_i = (p_{ix}, p_{iy}, p_{iz})\}$ and target set $Q = \{\mathbf{q}_j | \mathbf{q}_j = (q_{jx}, q_{jy}, q_{jz})\}$, we need to find a motion transformation ψ, composed by a rotation R and a translation $\mathbf{t}$, that applied to P best aligns both clouds ($\psi(P) \approx Q$), according to a distance metric.

The classical and most cited algorithm in the literature to rigid registration is the Iterative Closest Point (ICP) [2]. This algorithm takes as input the point clouds P and Q, and consists of the iteration of two major steps: matching between the point clouds and transformation estimation. The matching searches the closest point in P for every point in Q. This set of correspondences is used to estimate a rigid transformation. These two steps are iterated until a termination criterion is satisfied.

Although simple in concept, ICP assumes that there is a correct correspondence between the points of both clouds. This assumption easily fails on real applications because, in general, the acquired data is noisy and we need to scan the object from multiple directions, due to self-occlusion as well as limited sensor range, producing only partially overlapped point clouds. Another issue of ICP and some variants is that they expect that the point clouds are already coarsely aligned.

The mentioned issues have been more or less addressed by more recent methods that have been produced by researchers in computer graphics, computational geometry and computer vision communities, as we can see in related surveys [3]–[5]. The large variety of such techniques poses a problem to decide a specific technique for a specific application.

In this paper, our goal is to compare the convergence characteristics of surface registration methods in the frame-to-frame problem, where the frames are obtained from a video stream of range images. The video sequence is processed to extract the point clouds that represent sample sets of the target object surface. The objective is to register point clouds in consecutive frames. In this application, we observe the following problems: partial overlapping point clouds, noise, outliers, scale variation, and missing data.

In order to limit the scope of the problem, and avoid a combinatorial explosion in the number of possibilities to test, we focus on rigid transformation techniques that fulfill at least one of the following requirements: (a) Incorporate local geometric features to enhance the quality of the matching step; (b) Estimate the transformation using a distance different from the Euclidean one; (c) Perform registration without correspondence.

The former is motivated by the fact that ICP, and many other registration techniques, use just the criterion of minimizing point-to-point Euclidean distances between the sets P and Q to compute the matching between the point clouds. This approach might not be efficient in cases of partial overlapping, because only a subset of each point cloud has a correct correspondent instead of all the points. We are also supposing that the video is acquired by simply waving the capture device at the scene following smooth and slow motion paths. Therefore we can discard scale changes when registering two consecutive frames, since they should be very small, which justifies only contemplate rigid transformations. Moreover, the characteristics of the solution to the rigid registration problem depends on the used notion of distance in the environment space. Usually, registration techniques apply the Euclidean distance that is derived from the L_2 norm. However, when using the L_2 norm, we get an optimization problem in the least-squares sense, imposing a fundamental assumption that the error residuals assume a normal distribution, where inliers are typical events whereas outliers rarely happen. Another paradigm, that motivates requirement (b), would be to use a norm that maximizes the number of zero distances between correspondences. Besides, the requirement (c) comes because we would like to test a method that attempts to align the given two point sets without establishing the explicit point correspondence. A trick in this case is to model each of the two point sets by a probability distribution, in order to get a procedure less sensitive to missing correspondences and outliers. Obviously, we must consider the ICP in order to obtain a relative measure about how efficient each chosen methodology is against the difficulties of the frame-to-frame registration problem.

Based on the aforementioned requirements, we choose the classical ICP, a combination between the ICP and the Comparative Tensor Shape Factor (ICP-CTSF), Shape-based Weighting Covariance ICP (SWC-ICP), Gaussian mixture model (GMM), Sparse ICP, Super 4PCS (S4PCS), and Sparse ICP combined with CTSF (Sparse ICP CTSF) [2], [6]–[10].

To evaluate each algorithm in the target application, we firstly consider point clouds acquired through a Cyberware 3030 MS scanner [11] available in the Stanford $3D$ scanning repository [12]. The Bunny model was chosen for the tests and the corresponding point clouds captured considering four viewpoints of it. In this case, the ground truth rotation is available and, as a consequence, we could evaluate four different metrics to measure the rotation error (Section IV). Results show better performance for Sparse ICP and Sparse ICP CTSF in these experiments in the inner product of unit quaternions metric. Besides the original data, a case-study is generated to simulate missing data. Visual results are shown in order to link the error measurements with the results of the methods on the chosen examples. However, when simulating missing data with the Bunny model, we notice a decrease in the registration precision of Sparse ICP and Sparse ICP CTSF. In this case, the ICP-CTSF obtain outstanding results. Moreover, we present the CPU time spent on the executions, in order to highlight the computational complexity of each technique.

Next, we evaluate the alignment techniques for frame-to-frame registration using three video sequences with depth information. We perform the segmentation of each frame of the sequence through a simple depth threshold operation. The obtained result generates a point cloud which we must register with the previous one. We use the average root mean squared error, average rotation and translation errors as measures to analyze the results. The tests show that ICP-CTSF is more reliable for this application.

This work is an extended version of the material published in [13]. In the current version we have improved the introduction and we add a related works section. Besides, the Section III (Registration Algorithms) is augmented with a description of each target technique to make the material self-contained. Also, in Section III, we offer details about tensor elements behind ICT-CTSF and SWC-ICP, with a complete derivation of the latter based on fundamental results in point clouds registration in $\mathbb{R}^3$. In the Experimental Results (Section V) we include one more case in the point clouds experiments, incorporate more details about the CPU time and the influence of trimming parameter. We substitute the scenario generated using noise and outliers used in [13] to new example involving missing points. With this, we can complete the results presented in [13] having tested the registration techniques against noise, outliers and missing data, that are common problems in frame-to-frame registration. Moreover, we have added new experiments to evaluate the techniques using two benchmark videos of the database available in the web site [14], that are accompanied with the ground truth for the rigid registration. Differently from the work [13], which was not conclusive in this point, the frame-to-frame registration results presented in Sections V-B and V-C, show that ICP-CTSF is the best method to register point clouds extracted from depth sequences.

The remainder of this paper is organized as follows. The Section II describes related works dealing with comparisons and qualitative analysis of rigid surface registration methods. Then, in Section III, we summarize the considered methods. Next, Section IV describes four different metrics to measure the rotation error. The Section V shows the experimental results obtained by applying the registration methods to point clouds and to depth video sequence. Section VI presents the conclusions and future researches.

II. Related Works

The survey of Sabata and Aggarwal [15] was one of the first works to list methods to compute 3D rigid motions between two sets, whether they are points, lines or surfaces. Points are the most common representation drawing attention from most papers of the rigid registration literature. They also classify the solution found by the methods in iterative or closed form. However, the listed methods are not compared. Eggert *et al.* [16] compare quantitatively four closed solution for estimating rigid transformations using controlled synthetic experiments: singular value decomposition [17], unit quaternion [18], dual quaternion [19], and orthonormal matrices [20]. No significant differences were observed in the accuracy and robustness of the algorithms for non-degenerate 3-D point sets with various levels of noise. In terms of stability, for non-degenerate cases, the unit quaternions and singular value decomposition methods were superior than the other methods, with the latter marginally more stable than the former.

Some variants of the ICP were surveyed by Rusinkiewicz and Levoy [21], that classified them in six stages where optimizations could be made: selection of points, matching, weighting correspondences, rejection of pairs, error metric and minimization of error metric. They compare the variants regarding the RMS error, number of iterations and the time until correct convergence, in order to propose a high-speed ICP, using the best strategy in each stage, to address real time registration.

Dalley and Flynn [22] presented a quantitative analysis of two methods to reject pairs of matched points, on partially overlapping range images. In these cases, there is an expected number of points without homologous correspondence, justifying the need of such methods.

Salvi *et al.* [3] proposed a classification of methods in fine registration and coarse registration. In fine registration, the methods try to find the most accurate solution as possible, refining an already computed initial guess. The latter is a class of algorithms that aim to find an initial estimation of the correct alignment between point sets. These methods tend to be more robust to noise once make no assumptions about the relative position of the point sets. However, in general, their solutions must be improved by a fine registration technique, that takes the coarse transformation as an initial estimation of the motion (a guess), and iterate until convergence to a more accurate solution. This way, new methodologies are generated through the combination of coarse and fine registration techniques, called coarse-to-fine schemes [7]. After reviewing some methods of each class, Salvi *et al.* [3] compare them measuring root mean squared error (RMS), rotation error, translation error and computational time.

Moreover, considering the specific point of rotation error, Huynh [23] presents a detailed analysis of six known functions for measuring distance between 3D rotations considering metric and group concepts ($SO(3)$; the group of orthogonal matrices with determinant $+1$). The conclusions favor quaternions for $3D$ rotations representation. Besides, according to Besl and McKay [2], for two and three dimensions, the quaternion-based method is preferred, since reflections are not desired.

In this paper, we show how some recent approaches to rigid registration perform in frame-to-frame application cases. To the best of our knowledge, it is the first work to address this kind of comparison. Besides the chosen techniques, we must take into account other recent works that could be also used in the target application. In [24] it is described an algorithm, based on a probabilistic model, for joint registration of multiple point clouds (JR-MPC). The technique shares with the GMM (Gaussian Mixture Model) [9] the idea of using Gaussian mixtures to represent point sets. However, differently from GMM, the JR-MPC assumes that all the point sets are generated from the same Gaussian mixture model, that includes also an uniform distribution parameterized by the volume of the convex hull encompassing the clouds. In our application we have a video stream V with $|V|$ frames, each one defining a point cloud in $\mathbb{R}^3$. The application of JR-MPC to jointly register these point sets is impractical. Besides, the assumption that such point clouds could be jointly registered could be false in such application due to scene changes along the frames.

Still in the scenario of probabilistic mixture models, the technique presented in [25] proposes a joint distribution associated to the observations that allow to incorporate color information associated with each $3D$ point. Despite of its theoretical generality, in practice this strategy cannot be directly employed for high dimensional 3D shape features due to complexity problems. Thus, in [26] the authors proposes an adaptation in the spirit of the bag-of-words paradigm in order to build a computationally efficient mixture model for the common joint distribution that originates the $3D$ points as well as the corresponding features. All these probabilistic mixture models suffer from both computational and memory cost issues for large point sets (tens of thousands or millions of points) due to the increase in the number of mixture components. The deterministic model [27] also associates RGB information and depth measurements through a four dimensional approach that allows to design an ICP version in RGB-D space without the computational complexity of mixture approaches.

Besides, in the case of cross-source point clouds, the performance of feature-based methods like [26] deteriorates due to the difficult to reliably extract similar features from point clouds acquired through different sensors. Such application motivates the CSGM technique [28], that applies a graph framework to organize and encode data information, which allows to convert the registration into a graph matching problem. In [28], the CSGM is also compared with ICP and JR-MPC for $3D$ data from the same kinds of sensor, outperforming the latter and achieving lower rate of error than JR-MPC in some tests.

In our work we avoid usual problems with RGB information (sensitivity against illumination conditions and shadows) by keeping only $3D$ data and shape features. We focus on point clouds acquired through a single sensor and apply shape features only to improve the match between point sets. Consequently, we consider only the methods already selected, which are reviewed in the next section.

III. Registration Algorithms

We compare in this work seven different algorithms to frame-to-frame rigid registration: the classical ICP [2] and four variants (the ICP-CTSF [6], SWC-ICP [7], Sparse ICP [8], and Sparse ICP CTSF [6]), the Super 4PCS [10], and the GMM framework [9]. In this section, we aim to establish the necessary notation and the mathematical formulation behind these techniques.

Hence, the bold uppercase symbols represent tensor objects, such as $\mathbf{T}, \mathbf{S}$; the normal uppercase symbols represent matrices, data sets and subspaces (P, U, D, Σ, etc.); the bold lowercase symbols denote vectors (represented by column arrays) such as $\mathbf{x}$, $\mathbf{y}$. The normal lowercase symbols are used to represent functions as well as scalar numbers (f, ψ, λ, α, etc.). Also, given a matrix $A \in \mathbb{R}^{m \times m}$ and a set S, then $tr\left(A\right) = A_{11} + A_{22} + \ldots + A_{mm}$ is the trace of A, and $|S|$ means the number of elements of S. Besides, I_m represents the $m \times m$ identity matrix.

Our focus is rigid registration in the frame-to-frame problem. So, let the source and target point clouds in $\mathbb{R}^m$ be represented, respectively, by $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{n_P}\} \subset \mathbb{R}^m$ and $Q = \left\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{n_Q}\right\} \subset \mathbb{R}^m$. A rigid transformation $\psi : \mathbb{R}^m \to \mathbb{R}^m$ is given by:

$$\psi\left(\mathbf{x}\right) = R\mathbf{x} + \mathbf{t}, \tag{1}$$

with $R \in SO(m)$ and $\mathbf{t} \in \mathbb{R}^m$ being the rotation matrix and translation vector, respectively.

The registration problem aims at finding a rigid transformation $\psi : \mathbb{R}^m \to \mathbb{R}^m$ that brings set P as close as possible to set Q in terms of a designated set distance, computed using a suitable metric $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$, usually the Euclidean one denoted by $d\left(\mathbf{p}, \mathbf{q}\right) = \left\|\mathbf{p} - \mathbf{q}\right\|_2$. To solve this task, the first step is to compute the matching relation $C\left(P, Q\right) \subset P \times Q$ that denotes the set of all correspondence pairs to be used as input in the procedure to compute the transformation ψ. Formally, we consider:

$$\begin{aligned} C(P, Q) = \{&(\mathbf{x}_{\mathbf{i}_1}, \mathbf{y}_{\mathbf{i}_1}) \in P \times Q; \\ &d(\mathbf{x}_{\mathbf{i}_1}, \mathbf{y}_{\mathbf{i}_1}) \leq d(\mathbf{x}, \mathbf{y}_{\mathbf{i}_1}), \forall \mathbf{x} \in P\} \end{aligned} \tag{2}$$

where $P \times Q$ denotes the Cartesian product between sets P and Q. We can check that $|C\left(P, Q\right)| = |Q|$. However, in the remaining text we say that $|C\left(P, Q\right)| = c$ to simplify the expressions.

Moreover, in the focused application only partial matches are expected in general. Therefore, it is desirable a trimmed approach that discards a percentage of the worst matches [29]. So, we sort the pairs of the set $C\left(P, Q\right)$ such that $d\left(\mathbf{x}_{i_1}, \mathbf{y}_{i_1}\right) \leq d\left(\mathbf{x}_{i_2}, \mathbf{y}_{i_2}\right) \leq \cdots \leq d\left(\mathbf{x}_{i_c}, \mathbf{y}_{i_c}\right)$ and consider a trimming parameter $0 \leq \tau \leq 1$ and the new correspondence relation:

$$\begin{aligned} C_1(P, Q, \tau) = \{&(\mathbf{x}_i, \mathbf{y}_i) \in C(P, Q); \\ &d\left(\mathbf{x_i}, \mathbf{y_i}\right) \leq d\left(\mathbf{x}_{i_{c(1-\tau)}}, \mathbf{y}_{i_{c(1-\tau)}}\right)\}, \end{aligned} \tag{3}$$

which is supposed to have $|C_1\left(P, Q, \tau\right)| = n$. We must notice that $C_1\left(P, Q, \tau\right) = C\left(P, Q\right)$ if $\tau = 0$.

The relationship defined by the expression (3) is based on the distance function and nearest neighbor computation. We could also consider shape descriptors computed over each point cloud. Generally speaking, given a point cloud S, the shape descriptors can be formulated as a function $f : S \to \mathbb{P}\left(\mathbb{R}\right)$, where $\mathbb{P}\left(\mathbb{R}\right)$ is the set of all subsets of $\mathbb{R}$, named the power set of $\mathbb{R}$. In this case, besides the distance criterion, we can also include shape information in the correspondence computation by applying a boolean correspondence function $f^c : P \times Q \to \{0, 1\}$ such that [5]:

$$f^c(\mathbf{p}, \mathbf{q}) = \begin{cases} 1 & if\ f\left(\mathbf{p}\right) \approx f\left(\mathbf{q}\right) \\ 0 & otherwise \end{cases}. \tag{4}$$

Also, before building $C\left(P, Q\right)$ in expression (2) we could perform a down-sampling in the two point sets, based on the selection of key points through the shape function, or through a naive interlaced sampling over same spatial data structure [30].

A. Iterative Closest Point

The classical ICP [2], described in the Algorithm 1, receives the source P and target Q point clouds and each iteration of the main loop is composed by two major steps: matching between the point clouds and transformation estimation. The former is performed by computing the set $C_1\left(P_{s+1}, Q, \tau\right)$ through equation (2). At the end of the matching process we get a base of the set P, denoted by $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset P$, and a base of the set Q, denoted by $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \subset Q$ such that $C_1\left(P, Q, \tau\right)$ stands for the set of n correspondence pairs $\left(\mathbf{x}_i, \mathbf{y}_i\right) \in X \times Y$. This matching relation will be used to estimate a rigid transformation that aligns the point clouds P and Q. Specifically, ICP seeks for a rotation matrix R and a translation $\mathbf{t}$ that minimizes the mean squared distance:

$$e^2\left(R, \mathbf{t}\right) = \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{y}_i - \left(R\mathbf{x}_i + \mathbf{t}\right)\right\|_2^2, \tag{5}$$

which is used as a measure of the distance between the target set Q and the transformed source point cloud $\psi\left(P\right) = \{\psi\left(\mathbf{p}_1\right), \psi\left(\mathbf{p}_2\right), \ldots, \psi\left(\mathbf{p}_n\right)\}$, with ψ defined by equation (1). Now, we focus in the specific three-dimensional case ($m = 3$) and state the fundamental theorem that steers most of the solutions for the registration problem in $\mathbb{R}^3$.

Theorem 1: Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^3$ and $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \subset \mathbb{R}^3$, the centers of mass μ_x, μ_y for the respective point sets X and Y, the cross-covariance Σ_{xy}, and the matrices A and M, given by:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i, \tag{6}$$

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i, \tag{7}$$

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n} \left(\mathbf{y}_i - \mu_y\right)\left(\mathbf{x}_i - \mu_x\right)^T. \tag{8}$$

$$A = \left(\Sigma_{xy} - \Sigma_{xy}^T\right), \tag{9}$$

$$M\left(\Sigma_{xy}\right) = \left[\begin{array}{cccc} tr\left(\Sigma_{xy}\right) & A_{23} & A_{31} & A_{12} \\ A_{23} & & & \\ A_{31} & & \Sigma_{xy} + \Sigma_{xy}^T - tr\left(\Sigma_{xy}\right) I_3 & \\ A_{12} & & & \end{array}\right]. \tag{10}$$

Hence, the optimum rotation R and translation $\mathbf{t}$ vector that minimizes the error in expression (5) are determined uniquely as follows [18]. The matrix R is computed through the unit eigenvector $\mathbf{v} = \begin{pmatrix} v_0 & v_1 & v_2 & v_3 \end{pmatrix}^T$ of M, corresponding to its maximum eigenvalue:

$$R = \begin{bmatrix} 1-2(v_1^2+v_2^2) & 2(v_0v_1 - v_2v_3) & 2(v_0v_2 + v_1v_3) \\ 2(v_0v_1 + v_2v_3) & 1-(v_0^2+v_2^2) & 2(v_1v_2 - v_0v_3) \\ 2(v_0v_2 - v_1v_3) & 2(v_2v_2 + v_0v_3) & 1-(v_0^2+v_1^2) \end{bmatrix}, \tag{11}$$

and t is calculated through R and centroids in expressions (6)-(7) as:

$$\mathbf{t} = \mu_y - R\mu_x. \tag{12}$$

□

Based on the above theorem, in the second stage, the ICP estimates the rigid transformation by computing the rotation matrix and translation vector using equations (11) and (12). The matching and transformation estimation are repeated until the allowed maximum number of iterations is achieved or the error falls bellow a pre-defined threshold. The ICP technique is summarized in the Algorithm 1.

```
Algorithm 1: Iterative Closest Point
Data: $P = \left\{ \mathbf{p}_i \in \mathbb{R}^3; \mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^T \right\}$,
      $Q = \left\{ \mathbf{q}_i \in \mathbb{R}^3; \mathbf{q}_i = (q_{i_1}, q_{i_2}, q_{i_3})^T \right\}$; trimming $\tau$;
begin
    $P_0 = P$, $s = 0$.
    $\varepsilon_0 = \infty$.
    $R_0 = I_3$, $\mathbf{t}_0 = (0,0,0)^T$.
    repeat
        Apply the transformation to all points of the source:
        $P_{s+1} = R_s P_s + \mathbf{t}_s \equiv \{R_s \mathbf{p} + \mathbf{t}_s, \quad \mathbf{p} \in P_s\}$.
        Compute the matching relation $C_1(P_{s+1}, Q, \tau)$ through expression (2).
        Compute the principal eigenvector $\mathbf{v}$ of the matrix $M$ defined in (10).
        Calculate the rotation matrix $R_{s+1}$ and translation vector $\mathbf{t}_{s+1}$ using expressions (11)-(12).
        Compute the error between the two point sets:
        $\varepsilon_{s+1} = e^2(R_{s+1}, \mathbf{t}_{s+1})$, from (5).
        $s \leftarrow s + 1$.
    until $\varepsilon_s > \varepsilon_{s-1}$;
    return $R, \mathbf{t}$.
end
```

B. ICP-CTSF

The ICP-CTSF [6] implements a matching strategy using a feature invariant to rigid transformations, based on the shape of second-order orientation tensors associated to each point. A voting algorithm is used, divided into an isotropic and an anisotropic voting field. So, given a cloud point $\mathbf{p} \in P$, let $L_k(\mathbf{p}) \subset P$ be the set of $k\%$ nearest neighbor of $\mathbf{p}$ and $\mathbf{s} \in L_k(\mathbf{p})$. We can define $\mathbf{v}_{ps} = (\mathbf{s} - \mathbf{p})$, $\widehat{\mathbf{v}}_{ps} = \mathbf{v}_{ps}/||\mathbf{v}_{ps}||_2$, as well as the function:

$$\sigma(\mathbf{p}) = \sqrt{\frac{||\mathbf{s}_f - \mathbf{p}||_2^2}{\ln 0.01}}. \tag{13}$$

where $\mathbf{s}_f$ is farthest neighbor of $\mathbf{p}$, which has influence 0.01. Given these elements, we can compute the second-order tensor field:

$$\mathbf{T}(\mathbf{p}) = \sum_{\mathbf{s} \in L_k(\mathbf{p})} \exp\left[\frac{-||\mathbf{v}_{ps}||_2^2}{\sigma^2(\mathbf{p})}\right] \cdot \left(\widehat{\mathbf{v}}_{ps} \cdot \widehat{\mathbf{v}}_{ps}^T\right), \tag{14}$$

which is the isotropic voting field computed through a weighted sum of tensors $\widehat{\mathbf{v}}_{ps} \cdot \widehat{\mathbf{v}}_{ps}^T$, built from the function (13) and from the vote vectors $\mathbf{v}_{ps}$, $\mathbf{s} \in L_k(\mathbf{p})$.

Let the orthonormal basis generated by the eigenvectors $(\mathbf{e}_1(\mathbf{p}), \mathbf{e}_2(\mathbf{p}), \mathbf{e}_3(\mathbf{p}))$ of $\mathbf{T}(\mathbf{p})$ and the corresponding eigenvalues supposed to satisfy $\lambda_3(\mathbf{p}) < \lambda_2(\mathbf{p}) \leq \lambda_1(\mathbf{p})$. In this case, the local geometry at the point $\mathbf{p}$ can be represented by the Figure 1 where we picture together the following elements: the coordinate system $\widehat{x}, \widehat{y}, \widehat{z}$ oriented through the eigenvectors $(\mathbf{e}_1(\mathbf{p}), \mathbf{e}_2(\mathbf{p}), \mathbf{e}_3(\mathbf{p}))$, the plane π that contains the point $\mathbf{p}$, its neighbor $\mathbf{s}$ and the axis $\widehat{z}$. Moreover, Figure 1 shows the unique ellipse $E \subset \pi$ that is tangent to the $\widehat{x}, \widehat{y}$ plane in $\mathbf{p}$, contains $\mathbf{s}$, and is centered at a point in $\widehat{z}$. The vector $\widehat{\xi}_s$, that is unitary, parallel to the plane π, and tangent to E at $\mathbf{s}$, gives a way to build a different structuring element that enhances coplanar structures in the sense that the angle $\beta \approx 0$ if $\mathbf{s}$ is close to the $\widehat{x}, \widehat{y}$ plane. Specifically, if $d_e(\mathbf{p}, \mathbf{s})$ is the length of the minor arc from $\mathbf{p}$ to $\mathbf{s}$ along the ellipse E in Figure 1, we define a new weighting function:

$$g(\mathbf{p}, \mathbf{s}) = \begin{cases} \exp\left[\dfrac{-d_e(\mathbf{p}, \mathbf{s})}{\sigma^2(\mathbf{p})}\right], & \tan\phi_s \leq \tan\phi_{max}, \\ 0.0 & , \tan\phi_s > \tan\phi_{max}, \end{cases} \tag{15}$$

where $\sigma^2(\mathbf{p})$ is calculated by expression (13), ϕ_s is the angle between $\mathbf{v}_{ps} = (\mathbf{s} - \mathbf{p})$ and the $\hat{x}, \hat{y}$ plane, ϕ_{max} constrains the influence of points misaligned to the $\hat{x}, \hat{y}$ plane, with $45°$ an ideal choice, as a mid term between smoother results and robustness to outliers [31], [32].

With the above elements in mind, it is defined the tensor field $\mathbf{S}(\mathbf{p})$, that is composed by the weighted sum of the tensors built from the votes received on the point, with weights computed by expression (15) for all the points that have $\mathbf{p}$ as a neighbor:

$$\mathbf{S}(\mathbf{p}) = \sum_{\mathbf{s} \in L_k(\mathbf{p})} g(\mathbf{p}, \mathbf{s}) \cdot \left(\widehat{\xi}_s \cdot \widehat{\xi}_s^T\right) \tag{16}$$

The tensor field in expression (16) can be seen as a shape function $\mathbf{S} : P \to \mathbb{R}^{3\times3}$ whose descriptors at a point $\mathbf{p} \in P$ are the eigenvalues $\lambda_i^{\mathbf{S}}(\mathbf{p}), i = 1, 2, 3$. Therefore, given two points $\mathbf{p}, \mathbf{q}$ such that $\mathbf{p} \in P$ and $\mathbf{q} \in Q$, we compare the

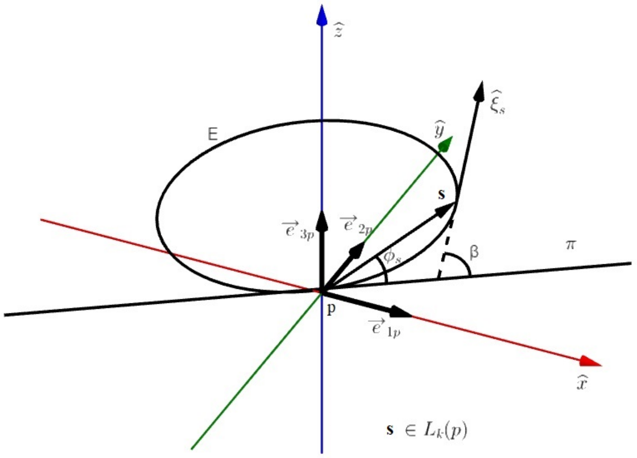


Fig. 1: Geometric representation of the angles ϕ_s, β and unitary vector $\widehat{\xi}_s$ of an arbitrary point $\mathbf{s}$.

corresponding (local) geometries using the comparative tensor shape factor (CTSF), defined as:

$$CTSF\left(\mathbf{p},\mathbf{q}\right) = \sum_{i=1}^{3}\left(\lambda_i^{\mathbf{S}_1}\left(\mathbf{p}\right) - \lambda_i^{\mathbf{S}_2}\left(\mathbf{q}\right)\right)^2, \qquad (17)$$

where $\mathbf{S}_1 : P \to \mathbb{R}^{3\times 3}$ and $\mathbf{S}_2 : Q \to \mathbb{R}^{3\times 3}$ are tensors computed following expression (16) and $\lambda_i^{\mathbf{S}_1}\left(\mathbf{p}\right)$ and $\lambda_i^{\mathbf{S}_2}\left(\mathbf{q}\right)$ are the ith eigenvalues calculated in the points $\mathbf{p} \in P$ and $\mathbf{q} \in Q$, respectively.

The CTSF is used side by side with the Euclidean distance to produce a correspondence set that takes into account not only the nearest point (like in expression (2)) but also the shape information:

$$d_{c,m}(\mathbf{p},\mathbf{q},m) = ||\mathbf{p}-\mathbf{q}||_2 + w_m \cdot CTSF\left(\mathbf{p},\mathbf{q}\right), \qquad (18)$$

where $CTSF\left(\mathbf{p},\mathbf{q}\right)$ is given by Equation (17), $w_m = w_0 b^m$, with $b < 1$, and $0 < w_m < w_0$.

The parameter w_0 is the initial weight given to the CTSF and b controls the update size of the weighting factor. To avoid numerical instabilities we set $w_m = 0$, when $w_m \approx 0$. This weighting strategy is responsible for its coarse-to-fine behavior when inserted in the matching step of the ICP algorithm, given by expression (2). Specifically, the ICP-CTSF procedure (Algorithm 2) calculates the correspondence relation:

$$\begin{aligned} C_2(P,Q,m) = \{(\mathbf{x}_{\mathbf{i}_1},\mathbf{y}_{\mathbf{i}_1}) \in P\times Q;& \\ \forall \mathbf{y}_{\mathbf{i}_\mathbf{k}} \in Q,& \\ d_{c,m}(\mathbf{x}_{\mathbf{i}_1},\mathbf{y}_{\mathbf{i}_\mathbf{k}},m) \geq d_{c,m}(\mathbf{x}_{\mathbf{i}_1},\mathbf{y}_{\mathbf{i}_1},m)\},& \end{aligned} \qquad (19)$$

and uses it to define the set:

$$\begin{aligned} C_3(P,Q,\tau,m) = \{(\mathbf{x}_\mathbf{i},\mathbf{y}_\mathbf{i}) \in C_2(P,Q,m);& \\ f^{trim}(\mathbf{x}_\mathbf{i},\mathbf{y}_\mathbf{i},\tau) = 1\},& \end{aligned} \qquad (20)$$

which is the correspondence set applied by the ICP-CTSF technique, which is summarized in the Algorithm 2.

Algorithm 2: ICP-CTSF Procedure

```
Data: $P = \left\{\mathbf{p}_i \in \mathbb{R}^3; \mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^T\right\}$,
      $Q = \left\{\mathbf{q}_i \in \mathbb{R}^3; \mathbf{q}_i = (q_{i_1}, q_{i_2}, q_{i_3})^T\right\}$; trimming $\tau$;
      $b$, such that $0 < b < 1$; $w_0 \gg 0$;
begin
    $P_0 = P$, $s = 0$, $m = 1$.
    $\varepsilon_0 = \infty$.
    $R_0 = I_3$, $\mathbf{t}_0 = (0,0,0)^T$.
    repeat
        Apply the transformation to all points of the source:
        $P_{s+1} = R_s P_s + \mathbf{t}_s \equiv \{R_s\mathbf{p} + \mathbf{t}_s, \quad \mathbf{p} \in P_s\}$.
        Compute the matching relation
        $C_3\left(P_{s+1}, Q, \tau, m\right)$ through expression (20).
        Compute the principal eigenvector $\mathbf{v}$ of the matrix $M$ defined in (10).
        Calculate the matrix rotation matrix $R_{s+1}$ and translation vector $\mathbf{t}_{s+1}$ using expressions (11)-(12).
        Compute the error between the two point sets:
        $\varepsilon_{s+1} = e^2\left(R_{s+1}, \mathbf{t}_{s+1}\right)$, from (5).
        if $\varepsilon_{s+1} > \varepsilon_s$ then
            $m \leftarrow m + 1$.
            $w_m \leftarrow w_0 b^m$.
        end if
        $s \leftarrow s + 1$.
    until $\varepsilon_s > \varepsilon_{s-1}$;
    return $R, \mathbf{t}$.
end
```

C. SWC-ICP Technique

In this technique, besides the correspondence relation (2), we also use the correspondence set:

$$\begin{aligned} C_{CTSF}(P,Q) = \{\ (\mathbf{s}_\mathbf{i},\mathbf{y}_\mathbf{i}) \in P\times Q;& \\ \mathbf{s}_\mathbf{i} = \arg\min_{\mathbf{p}\in P}(CTSF(\mathbf{p},\mathbf{y}_\mathbf{i}))\},& \end{aligned} \qquad (21)$$

which contains the pairs of points $(\mathbf{s}_i, \mathbf{y}_i) \in P \times Q$ whose local shapes are the most similar, according to the $CTSF$ criterion calculated by expression (17). In order to combine both correspondence sets, we firstly develop expression (8) to get:

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{y}_i\mathbf{x}_i^T\right) - \mu_y\mu_x^T. \qquad (22)$$

So, if we take expression (5) and perform the substitution:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \omega_n\mathbf{s}_i \qquad (23)$$

with $\omega_n \in \mathbb{R}$, we can write the mean squared error (5) as:

$$e^2\left(R,\mathbf{t}\right) = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{y}_i - \left[R\left(\mathbf{x}_i + \omega_n\mathbf{s}_i\right) + \mathbf{t}\right]\|_2^2. \qquad (24)$$

Also, by substituting the variable change (23) in expression (6) we get:

$$\begin{aligned}\mu_{x+\omega_n\mathbf{s}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x_i}+\omega_n\mathbf{s_i}\right)\\ &= \left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i}\right)+\omega_n\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{s_i}\right)\equiv\mu_{\mathbf{x}}+\omega_n\mu_{\mathbf{s}},\end{aligned} \tag{25}$$

and, consequently:

$$\Sigma_{x+\omega_n\mathbf{s},y}=\frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{y}_i\left(\mathbf{x}_i+\omega_n\mathbf{s}_i\right)^T\right]-\mu_y\left(\mu_x+\omega_n\mu_s\right)^T, \tag{26}$$

where μ_y is computed by equation (7). We shall notice that the matrix (26) combines the matching relations (2) and (21) being fundamental for the SWC-ICP described in [7]. According to the Theorem 1, the optimum rotation matrix R and translation vector $\mathbf{t}$ that minimizes the error in expression (24) are uniquely determined by equations (11)-(12) where $\mathbf{v} = \begin{pmatrix} v_0 & v_1 & v_2 & v_3\end{pmatrix}^T$ is the unit eigenvector of $M\left(\Sigma_{x+\omega_n\mathbf{s},y}\right)$ corresponding to the maximum eigenvalue. However, the SWC-ICP methodology achieves a coarse-to-fine behavior through the use of the weighting strategy of the ICP-CTSF. The SWC-ICP technique can be summarized in the Algorithm 3.

D. Sparse ICP

The Sparse ICP [8] is formulated as recovering a rigid transformation that maximizes the number of null residuals $\mathbf{z}_i = \mathbf{Rx_i}+\mathbf{t}-\mathbf{y_i}$, where R is the rotation matrix and $\mathbf{t}$ is a translation vector. The Sparse ICP uses L^p norm, $p\in[0,1]$, to implement this idea. So, given the correspondence set $C_1(P,Q)$ in expression (2) and the residual vector $\mathbf{z}=[||\mathbf{z}_1||_2^p,...,||\mathbf{z}_n||_2^p]^T$, the objective is to find a large set of inliers, $||\mathbf{z}_i||_2^p\approx 0$, and a small set of outliers, $||\mathbf{z}_i||_2^p >> 0$. This can be written as:

$$\min_{R,\mathbf{t},Z}\sum_{i=1}^{n}||\mathbf{z}_i||_2^p,\ \text{such that, } \delta_\mathbf{i}=\mathbf{0}, \tag{27}$$

where $\delta_\mathbf{i} = \mathbf{Rx_i}+\mathbf{t}-\mathbf{y_i}-\mathbf{z}_i$, and $Z=(\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_n)$ represents a generic point in the residual space. This constrained problem can be solved using an *augmented Lagrangian* method, which uses the Lagrangian:

$$\mathcal{L}_A(R,\mathbf{t},Z,\Lambda)=\sum_{i=1}^{n}\left(||\mathbf{z}_i||_2^p+\lambda_i^T\delta_i+\frac{\varrho}{2}||\delta_i||_2^2\right), \tag{28}$$

with Lagrange multipliers $\Lambda=\{\lambda_i\in\mathbb{R}^m, i=1...n\}$, penalty weight $\varrho>0$, and the restriction that R is a rotation matrix. Equation (28) is optimized using an alternating direction method of multipliers (ADMM). The Algorithm 4 summarizes the Sparse ICP procedure.

Algorithm 3: SWC-ICP Technique

```
Data: $P = \left\{\mathbf{p}_i\in\mathbb{R}^3; \mathbf{p}_i=(p_{i_1},p_{i_2},p_{i_3})^T\right\}$,
      $Q = \left\{\mathbf{q}_i\in\mathbb{R}^3; \mathbf{q}_i=(q_{i_1},q_{i_2},q_{i_3})^T\right\}$; trimming $\tau$;
      $b$, such that $0<b<1$; $w_0\gg 0$;
begin
    $P_0 = P$, $s=0$, $m=1$.
    $\varepsilon_0=\infty$.
    $R_0=I_3$, $\mathbf{t}_0=(0,0,0)^T$.
    Compute the matching relations $C_{CTSF}(P_j,Q)$
    through expression (21).
    repeat
        Apply the transformation to all points of the
        source:
        $P_{s+1}=R_sP_s+\mathbf{t}_s\equiv\{R_s\mathbf{p}+\mathbf{t}_s,\quad \mathbf{p}\in P_s\}$.
        Compute the matching relation $C_1(P_{s+1},Q,\tau)$
        through expression (2).
        Build the covariance matrix from (26) using the
        shape correspondences (21) and the nearest
        neighbors (2).
        Compute the matrix $M$ in expression (10) using
        (26).
        Compute the principal eigenvector $\mathbf{v}$ of the
        matrix $M$.
        Calculate the rotation matrix $R_{s+1}$ and translation
        vector $\mathbf{t}_{s+1}$ using expressions (11)-(12).
        Compute the error between the two point sets:
        $\varepsilon_{s+1}=e^2(R_{s+1},\mathbf{t}_{s+1})$, from (5).
        if $\varepsilon_{s+1}>\varepsilon_s$ then
            $m\leftarrow m+1$.
            $w_m\leftarrow w_0b^m$.
        end if
        $s\leftarrow s+1$.
    until $\varepsilon_s>\varepsilon_{s-1}$;
    return $R,\mathbf{t}$.
end
```

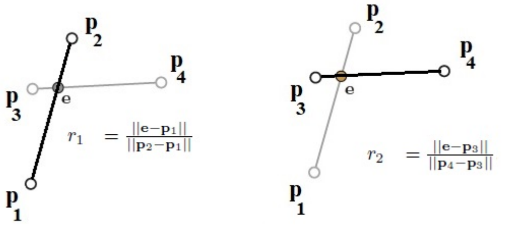
E. Super 4PCS

The Super 4PCS [10] is an improved version of the 4PCS [33] algorithm for global registration, or coarse registration according to Salvi [3]. Both methods follow the same idea of the RANSAC [34], [35], but instead of finding triplets of points, they search for all coplanar 4-points that are approximately congruent. The key property behind 4PCS is the fact that, given a set of coplanar points $B=\{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4\}\subset P$, not all collinear, it is always possible to define two lines such that they cross at an intermediate point $\mathbf{e}$, like in Figure 2.

The intersection point $\mathbf{e}$ can be computed considering the lines $s(t_1)=\mathbf{p}_1+t_1(\mathbf{p}_2-\mathbf{p}_1)$ and $s(t_2)=\mathbf{p}_3+t_2(\mathbf{p}_4-\mathbf{p}_3)$ and the solution of the linear system defined by the equation $s(t_1)=s(t_2)$. If $t_1=\hat{t}_1$ and $t_2=\hat{t}_2$ are the obtained solutions, then $\mathbf{e}=\mathbf{p}_1+\hat{t}_1(\mathbf{p}_2-\mathbf{p}_1)$ and $\mathbf{e}=\mathbf{p}_3+\hat{t}_2(\mathbf{p}_4-\mathbf{p}_3)$ and, consequently, we can compute the two corresponding

Algorithm 4: Sparse ICP Method

```
Data: $P = \left\{ \mathbf{p}_i \in \mathbb{R}^3; \mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^T \right\}$,
      $Q = \left\{ \mathbf{q}_i \in \mathbb{R}^3; \mathbf{q}_i = (q_{i_1}, q_{i_2}, q_{i_3})^T \right\}$; trimming $\tau$;
begin
    $P_0 = P$, $s = 0$.
    $\varepsilon_0 = \infty$.
    $R_0 = I_3$, $\mathbf{t}_0 = (0,0,0)^T$.
    repeat
        Apply the transformation to all points of the source:
        $P_{s+1} = R_s P_s + \mathbf{t}_s \equiv \{R_s \mathbf{p} + \mathbf{t}_s, \quad \mathbf{p} \in P_s\}$.
        Compute the matching relation $C_1(P_{s+1}, Q, \tau)$ through expression (2).
        Solve the problem in (27).
        Compute the error between the two point sets:
        $\varepsilon_{s+1} = e^2(R_{s+1}, \mathbf{t}_{s+1})$, from (5).
        $s \leftarrow s + 1$.
    until $\varepsilon_s > \varepsilon_{s-1}$;
    return $R, \mathbf{t}$.
end
```



$\mathbf{p}_2$
$\mathbf{p}_4$
$\mathbf{p}_3$
$\mathbf{e}$
$\mathbf{p}_1$
$r_1 = \frac{||\mathbf{e}-\mathbf{p}_1||}{||\mathbf{p}_2-\mathbf{p}_1||}$
$\mathbf{p}_2$
$\mathbf{p}_4$
$\mathbf{p}_3$
$\mathbf{e}$
$\mathbf{p}_1$
$r_2 = \frac{||\mathbf{e}-\mathbf{p}_3||}{||\mathbf{p}_4-\mathbf{p}_3||}$

Fig. 2: Four coplanar intersecting in $\mathbf{e}$ and affine invariant ratios $r_1$and r_2 (source [33]).

ratios:

$$\begin{aligned} \left|\hat{t}_1\right| \equiv r_1 &= \frac{||\mathbf{e}-\mathbf{p}_1||}{||\mathbf{p}_2-\mathbf{p}_1||}, \\ \left|\hat{t}_2\right| \equiv r_2 &= \frac{||\mathbf{e}-\mathbf{p}_3||}{||\mathbf{p}_4-\mathbf{p}_3||}, \end{aligned} \tag{29}$$

which are affine invariant because, given an affine transformation $\psi(\mathbf{p}) = A\mathbf{p} + \mathbf{t}$, with $det(A) \neq 0$, we can write $\psi(\mathbf{e}) = A\mathbf{e} + \mathbf{t} = A\left(\mathbf{p}_1 + \widehat{t_1}(\mathbf{p}_2 - \mathbf{p}_1)\right) + \mathbf{t}$ and $\psi(\mathbf{e}) = A\mathbf{e}+\mathbf{t} = A\left(\mathbf{p}_3 + \widehat{t_2}(\mathbf{p}_4 - \mathbf{p}_3)\right)+\mathbf{t}$, and perform the same algebra behind the demonstration of expressions (29) to get:

$$\begin{aligned} r_1 &= ||A\mathbf{e} - A\mathbf{p}_1||/||A\mathbf{p}_2 - A\mathbf{p}_1||, \\ r_2 &= ||A\mathbf{e} - A\mathbf{p}_3||/||A\mathbf{p}_4 - A\mathbf{p}_3||, \end{aligned} \tag{30}$$

which proofs that the ratios r_1 and r_2 are invariants under affine transformations.

Then, given the target Q, the main step of 4PCS algorithm is to extract the set U of all 4-points from Q that are approximately congruent to B, up to an approximation level δ. This search is performed by noticing that, for each pair of points $\mathbf{q}_1, \mathbf{q}_2 \in Q$, two intermediate points are computed using the affine invariants (29):

$$\mathbf{e}_1 = \mathbf{q}_1 + r_1(\mathbf{q}_2 - \mathbf{q}_1),$$

$$\mathbf{e}_2 = \mathbf{q}_1 + r_2(\mathbf{q}_2 - \mathbf{q}_1),$$

Whenever we have $\mathbf{e}_1 \simeq \mathbf{e}_2$, for any two pairs of points, then probably $\{\mathbf{q}_1, \mathbf{q}_2\} \subset Q$ belongs to a 4-points set that is an affine transformed copy of B. The set U defines a set T of rigid transformations $\psi_i(\mathbf{x}) = R_i\mathbf{x} + \mathbf{t}_i$ that best aligns B with some 4-points set in U. The solution of the registration problem is a rigid transformation $\psi \in T$ that brings set P as close as possible to set Q, in the sense defined by the Algorithm 5 that is found in [33].

Algorithm 5: 4PCS Procedure.

```
Data: $P = \left\{ \mathbf{p}_i \in \mathbb{R}^3; \mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^T \right\}$,
      $Q = \left\{ \mathbf{q}_i \in \mathbb{R}^3; \mathbf{q}_i = (q_{i_1}, q_{i_2}, q_{i_3})^T \right\}$, $\delta > 0$;
begin
    $h \leftarrow 0$.
    for $i = 1$ to $L$ do
        $B \leftarrow$ SelectCoplanarBase($P$).
        $U \leftarrow$ FindCongruent($B, Q, \delta$).
        forall the 4-points coplanar sets $U_i \in U$ do
            $\psi_i \leftarrow$ best rigid transformation that aligns $B$ to $U_i$ in the least square sense (minimize (5)).
            Find $S_i \subseteq P$, such that
            $d(\psi_i(S_i), Q) = e^2(R_i, \mathbf{t}_i) \leq \delta$.
        end forall
        $k \leftarrow \arg\max_i |S_i|$.
        if $|S_i| > h$ then
            $h \leftarrow |S_k|$.
            $\psi_{opt} \leftarrow \psi_k$.
        end if
    end for
    return $\psi_{opt}$."
end
```

Although the results of the 4PCS are satisfactory, it has a quadratic time complexity, limiting its applicability. The Super 4PCS [10] solves two of the 4PCS main bottlenecks: finding all points in a given distance threshold in a point set, and removing the redundant 4-points that arise due to affine invariants. These two improvements reduce the time complexity to run in linear time, in the number of data points.

F. GMM Framework

All the previous techniques involve methods that align two point sets based on some procedure for establishing the explicit point set correspondence. The Gaussian mixture model framework (GMM) discards the matching step and thus may achieve more robustness against the missing correspondences and outliers. In this registration framework each input point set is represented using a Gaussian mixture model where the number of Gaussian components is the number of points [9]. Besides, the mean vectors of the components are given by the position of the points and all components share the same spherical covariance matrix. Formally, given a point set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_X}\} \subset \mathbb{R}^m$, the mixture of Gaussians used in GMM model is computed by:

$$G(\mathbf{x}, X, \Omega, \mathbf{w}) = \sum_{i=1}^{n_X} w_i \phi(\mathbf{x}|\mathbf{x}_i, \Omega), \tag{31}$$

where:

$$\begin{aligned} \phi(\mathbf{x}|\mathbf{x}_i, \Omega) = & \frac{1}{\sqrt{(2\pi)^m det(\Omega)}} \\ & \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T \Omega^{-1} (\mathbf{x} - \mathbf{x}_i)\right), \end{aligned} \tag{32}$$

$\mathbf{w} = (w_1, w_2, \ldots, w_{n_X})^T$ is the vector of weights and Ω is the covariance matrix of the model. Without prior knowledge, all mixture components are weighted equally ($w_1 = w_2 = \ldots = w_{n_X}$) and $\Omega = diag\begin{pmatrix} \sigma & \sigma & \ldots & \sigma \end{pmatrix}$, with $\sigma > 0$ being the scale (or variance) of the model.

In this context, given source (P) and target (Q) point clouds, the problem of point set registration is reformulated through the minimization of a statistical discrepancy measure between the corresponding mixtures, given by expression (31). In the GMM proposed in [9] authors apply L_2 distance for measuring similarity between two Gaussian mixtures $G(\mathbf{x}, Q, \Omega, \mathbf{w})$ and $G(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w})$, representing the target Q and the transformed source $\psi(P) = \{R\mathbf{p}_1 + \mathbf{t}, R\mathbf{p}_2 + \mathbf{t}, \ldots, R\mathbf{p}_{n_P} + \mathbf{t}\}$, respectively, where ψ is the rigid transformation in expression (1), defined by the rotation R and translation t. So:

$$\begin{aligned} d\left(G(\mathbf{x}, Q, \Omega, \mathbf{w}), G\left(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w}\right)\right) = \\ \int \left(G(\mathbf{x}, Q, \Omega, \mathbf{w}) - G\left(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w}\right)\right)^2 d\mathbf{x} = \\ \int (G(\mathbf{x}, Q, \Omega, \mathbf{w}))^2 dx - \\ 2\int G(\mathbf{x}, Q, \Omega, \mathbf{w}) G\left(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w}\right) dx + \\ \int \left(G\left(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w}\right)\right)^2 d\mathbf{x} \end{aligned}$$

The equation (33) becomes a function $f : \mathbb{R}^{2m} \rightarrow \mathbb{R}^+$ of the transformation parameters that can be grouped in a vector $\theta = (\alpha_1, \alpha_2, \ldots, \alpha_m; t_1, t_2, \ldots, t_m)$ where α_i, t_i give the parametrization of the rotation matrix R and the translation vector $\mathbf{t}$, respectively. Hence, the registration becomes an optimization problem, where the objective function is $f(\theta) = d(G(\mathbf{x}, Q, \Omega, \mathbf{w}), G\left(\mathbf{x}, \psi(P), R\Omega R^T, \mathbf{w}\right))$. In practice, this cost function can be expressed by a discrete Gauss transform [36], [37] and the minimization of f could be achieved through traditional gradient-based methods. However, there are no guaranties of convexity for f in the θ domain. To overcome this problem, it is recommended in [9] to start with a relatively large scale σ and a default initial setting of parameters in θ and then performing the numerical optimization to estimate the rigid transformation ψ. Then, we compute the correspondence set $C(P, \psi(P))$, defined by expression (2). If $|C(P, \psi(P))|$ is less than a threshold value repeat the process by randomly chosen another initialization for θ. Besides, a multiscale approach can be applied by decreasing the valuer of σ in a coarse to fine strategy. The optimization process stops until a sufficient number of correspondences are obtained. The Algorithm 6 describes the GMM procedure [9]. In this algorithm we follow the original GMM description [9] and summarize the optimization approach as an 'annealing step' .

Algorithm 6: GMM Framework.

```
Data: $P = \left\{\mathbf{p}_i \in \mathbb{R}^3; \mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^T\right\}$,
      $Q = \left\{\mathbf{q}_i \in \mathbb{R}^3; \mathbf{q}_i = (q_{i_1}, q_{i_2}, q_{i_3})^T\right\}$;
begin
    Estimate and initial scale $\sigma$ from the input point sets.
    Specify an initial parameter $\theta$, e.g., from the identity transform.
    repeat
        Set up the objective function $f$, using expression (33).
        Optimize the objective function $f$ with $\theta$ as the initial parameter.
        Update the parameter $\theta \leftarrow \arg\min_{\theta} f$.
        Decrease the scale $\sigma$ accordingly to an annealing step.
    until Until some stopping criterion is satisfied;
    return The transformation parameter $\theta$.
end
```

IV. Rotation Error Metrics

We use four different metrics to measure the rotation error when a ground truth rotation is provided. The metrics are based on the norm of difference between quaternions [38], inner product of unit quaternions [39], Euclidean distance between the Euler angles [40] and deviation from the identity matrix [41]. The paper from Huynh [23] provides more details and comparisons between them. In what follows, we restrict the discussions to rotations in $3D$, that are represented by unit quaternions or matrices in $SO(3)$.

A. Norm of the Difference between Quaternions

The first is obtained by using the norm of the difference between unit quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ representing the provided ground truth and the obtained rotation, respectively:

$$\begin{aligned} \phi_1 : S^3 \times S^3 \rightarrow \mathbb{R}^+, \\ \phi_1(\mathbf{q}_1, \mathbf{q}_2) = \min\{||\mathbf{q}_1 - \mathbf{q}_2||_2, ||\mathbf{q}_1 + \mathbf{q}_2||_2\}, \end{aligned} \tag{33}$$

with $||\cdot||_2$ as the Euclidean norm and $S^3 = \{\mathbf{q} \in \mathbb{R}^4 \mid ||\mathbf{q}||_2^2 = 1\}$. We can show that $0 \leq \phi_1 \leq \sqrt{2}$ [23]. As pointed out by Huynh [23], ϕ_1 is a pseudo-metric in S^3, since $\phi_1(\mathbf{q}, -\mathbf{q}) =$

$0 \not\Rightarrow \mathbf{q} = -\mathbf{q}$, but in $SO(3)$, the group of 3D rotations, ϕ_1 is a metric.

B. Inner Product of Unit Quaternions

Similarly to the metric given by expression (33), the second metric also uses quaternions as follows:

$$\phi_2 : S^3 \times S^3 \to \mathbb{R}^+, \tag{34}$$

$$\phi_2(\mathbf{q}_1, \mathbf{q}_2) = \min\{\cos^{-1}(\mathbf{q}_1 \cdot \mathbf{q}_2), \pi - \cos^{-1}(\mathbf{q}_1 \cdot \mathbf{q}_2)\}, \tag{35}$$

with $\mathbf{q}_1$ and $\mathbf{q}_2$ also representing the provided ground truth and the obtained rotation, respectively, and $\cdot$ is the inner product. Huynh [23] rewrites this function to be more computationally efficient as:

$$\phi_3(\mathbf{q}_1, \mathbf{q}_2) = 1 - |\mathbf{q}_1 \cdot \mathbf{q}_2|. \tag{36}$$

This function is also a pseudo-metric in S^3, but it is a metric in $SO(3)$. Once we consider unit quaternions in expression (36) it is straightforward that $\phi_2, \phi_3 \in [0, 1]$.

C. Euclidean Difference between Euler Angles

The difference between two rotations can also be measured in function of their Euler angles [42]. Let $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$ be two sets of Euler angles,

$$\phi_4 : E \times E \to \mathbb{R}^+,$$

$$\phi_4((\alpha_1, \beta_1, \gamma_1), (\alpha_2, \beta_2, \gamma_2))$$

$$= (d(\alpha_1, \alpha_2)^2 + d(\beta_1, \beta_2)^2 + d(\gamma_1, \gamma_2)^2)^{1/2}, \tag{37}$$

where $d(a, b) = \min\{|a - b|, \quad 2\pi - |a - b|\}$, and $E \subset \mathbb{R}$ is an appropriate domain for the three Euler angles. In order to turn ϕ_4 into a metric in $SO(3)$, avoiding ambiguities in the representation, the Euler angles must be constrained $\alpha, \gamma \in [-\pi, \pi)$ and $\beta \in [-\pi/2, \pi/2)$, so that range of values of ϕ_4 is $[0, \pi/3]$.

D. Deviation from the Identity Matrix

In this case, ground truth and the computed rotations are represented by matrices $R_1, R_2 \in SO(3)$, respectively, and the metric function is calculated by [41]:

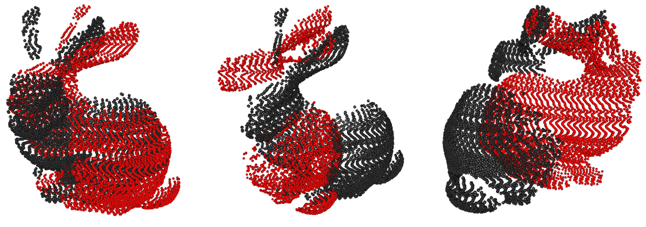
$$\phi_5 : SO(3) \times SO(3) \to \mathbb{R}^+, \tag{38}$$

$$\phi_5(R_1, R_2) = ||I - R_1 R_2^T||_F, \tag{39}$$

where $||\cdot||_F$ denotes the Frobenius norm of the matrix. We can prove that ϕ_5 is in fact a metric on SO(3) and that expression (39) gives values in the range $\left[0, 2\sqrt{2}\right]$ [23].

V. Experimental Results

We evaluate the performance of the methods described on Section III using two different setups. In the first one we compare the methods using point clouds captured in a controlled scenario. Our model is the Bunny, from the Stanford 3D Scanning Repository [12]. We use four clouds given by the views from 0°, 45°, 90° and 180°, and align the consecutive pairs. All point clouds lie in a unit bounding box. Figure 3 shows the three cases used, where in black we picture the initial pose (source) and in red the target one. The size of the original clouds are larger than 40000 points, which makes their processing too computational involved. Therefore, we uniformly sample these point clouds, selecting one point at each 10 and discarding the others, in order to reduce the computational time of each method.

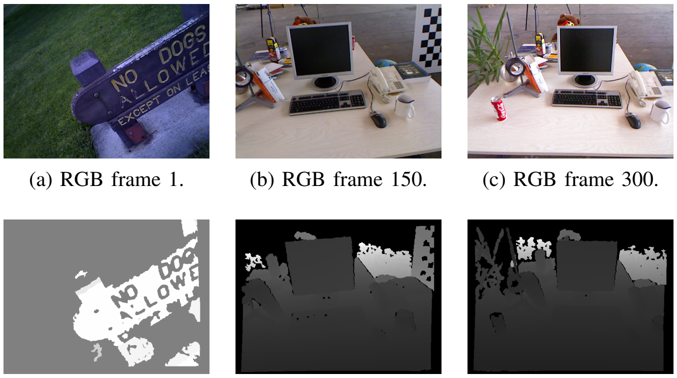


(a) 0°-45°. (b) 45°-90°. (c) 90°-180°.

Fig. 3: The three alignment cases tested in the first experiment.

The web documentation [12] offers the transformation to align the pairs of clouds shown in Figure 3. However, we have noticed that the precision of the translation vector is not suitable to perform a specific evaluation of the translation computed by the registration methods. Also, the models do not have a ground truth correspondence list. Hence, we firstly take the rotation given and compare it with the ones generated by the focused methods. Then, we take the 0° Bunny model configuration and build a case-study for registration under missing data. Furthermore, for each method, we measure the computational time to calculate the alignment and the rotation error obtained using the metrics described on Section IV.

The second experiments are performed using video sequences with RGB-D information. The first case-study is frame sequence captured using a PrimeSense Carmine camera [43]. This video belongs to the Large Dataset of Object Scans [44], and the sequence used is the #03118, containing 1489 depth frames. The choice was based on how easy it was to segment the background. Figure 4 illustrates the sequence. The next tests are implemented using two videos, named ‘freiburg2_xyz’ and ‘freiburg2_rpy’, from database available in [14] that, differently from the sequence #03118, provides information for debugging translations and rotations, which motivates their choices. Kinect is the acquisition hardware and frames have resolution of 640 × 480 pixels, yielding a depth image with 307200 points. All the experiments were carried out using an Intel Core i7-4790 CPU with 16GB RAM.



(a) RGB frame 1. (b) RGB frame 150. (c) RGB frame 300.

(d) Depth frame 1. (e) Depth frame 150. (f) Depth frame 300.

Fig. 4: (a)-(d) Sample from the sequence #03118, showing the RGB frame and its respective depth data (source [44]). (b)-(e) RGB and depth fields for frame sequence 'freiburg2_xyz' (source [14]). (c)-(f) RGB and depth information for video 'freiburg2_rpy' (source [14]).

A. Point Cloud Registration

In this section we have the following aims: (a) Analyze the different rotation error metrics (Section IV) to decide the best one(s) for the frame-to-frame registration problem; (b) Use the best error metric to compare the performance of the registration techniques described in Section III. In these tests, we consider the following degrees-of-freedom: (1) Registration technique; (2) Percentage k of neighbors; (3) Error metric; (4) Trimming parameter τ. Moreover, we wish to compare techniques using RMS criterion in a controlled setup with missing data.

The other parameters, besides k and τ, are set as follows. The update size of the weighting factor used in the ICP-CTSF and SWC-ICP is $b = 0.1$, $w_0 = 10^5$. It is an intermediate value that does not take too many updates, and neither finishes the update too soon, without proper exploration of the search space. The Super 4PCS was set with: $\delta = 0.005$, terminate threshold 0.8, without filtering by angle, normals, distance or color. Also, no further sampling of the point cloud is performed. The Sparse ICP and Sparse ICP with CTSF were set with parameters the same parameters used in [13]. The GMM setup follows the default values of the GMM implementation [45].

The SWC-ICP, ICP-CTSF and the Sparse ICP CTSF use tensors to match points through the computation of the C_{CTSF} relation given by expression (21). In these cases, we can evaluate the CTSF criterion using the isotropic voting field $\mathbf{T}$ or the anisotropic voting tensor $\mathbf{S}$. According to [7] better results have been obtained by applying the former in the SWC-ICP. However, the ICP-CTSF and Sparse ICP CTSF use the $\mathbf{S}$ field to compute the C_{CTSF} correspondence set [6].

To perform the task (a) we choose a pair of consecutive viewpoints, compute the error for each registration method using all the available metrics and visually compare the best alignment obtained according to each error metric. The best error metric is considered as the one which assigns the minimum error to the best visual alignment.

The visual inspection of the point clouds in Figure 3 indicates that the case pictured in Figure 3a is suitable as a case-study for the task (a) because, differently from Figures 3b and 3c, it is the easiest one with a large overlapping region and no discontinuities.

So, considering the degrees-of-freedom listed above, we set the trimming parameter $\tau = 0$ (no trimming) and compute the error metric for each registration technique using $k = 1\%, 5\%, 10\%, 25\%, 50\%, 75\%, 100\%$. In order to allow a fair comparison between the metrics we report the relative error, obtained by dividing the absolute error by the maximum value in the range of the focused rotation metric IV. We shall notice that $\phi_3 \in [0, 1]$, consequently the absolute and relative errors are the same in this case. Table I shows the minimum relative error according to each metric for $0° - 45°$. The Sparse ICP gives the smaller rotation error when considering all the metrics except ϕ_4 which achieves the minimum value for the Sparse ICP CTSF with $k = 25\%$. In special, the smallest error in Table I is obtained by the Sparse ICP with value almost null, given by 9.0×10^{-9}.

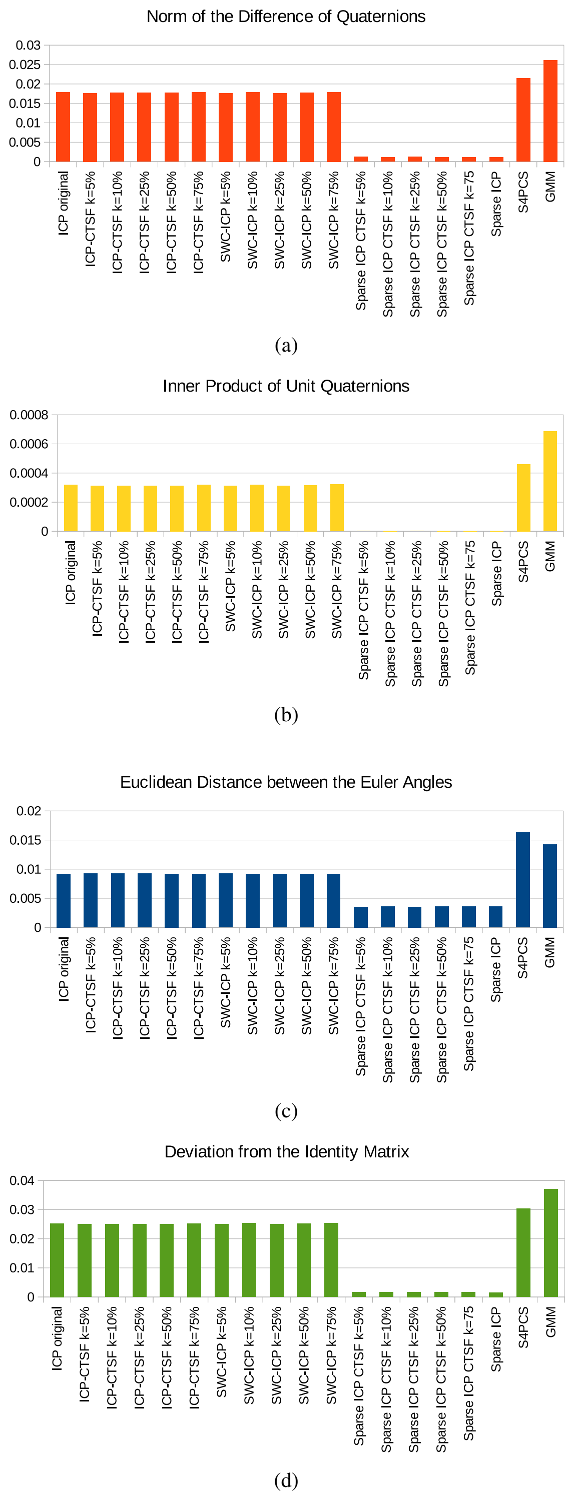
Figure 5 shows the absolute error obtained for the tests in the case $0° - 45°$, excluding $k = 1\%$ and $k = 100\%$ because they do not offer best results and sometimes they generate too large errors bringing scale problems in the visualization of smaller bars. The Sparse ICP and Sparse ICP CTSF algorithms presented the smaller errors, which agree with the results reported in Table I.

The visualization of Figures 5a-5d indicates that the ICP, ICP-CTSF and SWC-ICP achieve the second place in terms of rotation errors. Table II reports the minimum and maximum errors for these methods, according to each metric of Section IV. We can notice that, when considering the change $0° - 45°$, the variation of the parameter k did not influenced in the rotation error measured.

In order to check the results reported in Table I and Figure 5a-5d we show in Figures 6a-6b the overlapping of the source cloud ($0°$ view) and the target set ($45°$ view) after the application of the best transformations obtained. The visual inspection of Figure 6 agrees with the fact that Sparse ICP and Sparse ICP CTSF with $k = 25\%$ offer suitable alignments. However, the visualization is not precise enough to decide the best one. However, the Sparse ICP errors were the smallest ones for three of the four considered metrics. Also, according to metric ϕ_3, the error of the Sparse ICP is almost null. These observations indicate that Sparse ICP performs better than the

TABLE I: Best method and minimum relative error computed by each metric (Section IV) for the alignment $0° - 45°$.

Metric	Best Reg. Method	k	Relative Error
ϕ_1	Sparse ICP	-	0.000787
ϕ_3	Sparse ICP	-	0.0000000090
ϕ_4	Sparse ICP CTSF	$k = 25\%$	0.0033704
ϕ_5	Sparse ICP	-	0.0015736



(a)

(b)

(c)

(d)

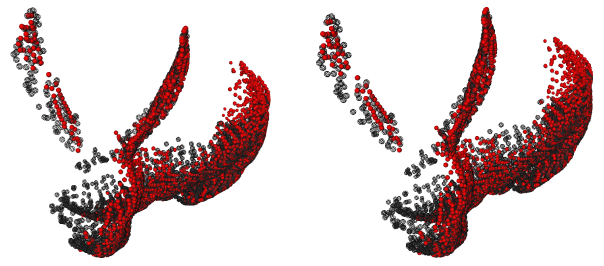
Fig. 5: Rotation error for the registration techniques (Trimming parameter $\tau = 0$), for the case $0° - 45°$, computed using: (a) ϕ_1. (b) ϕ_3; (c) ϕ_4; (d) ϕ_5.

other techniques and favor the choice of the ϕ_3 to measure the rotation error.

Figure 7 shows the errors obtained in the next experiments, for the case 45°-90°. Differently from the case $0° - 45°$, we observe that the Sparse ICP CTSF achieves the smallest rotation errors for all the metrics which is significantly smaller than the Sparse ICP rotation error. Table III reports the minimum and maximum relative errors achieved by the Sparse ICP CTSF with respect to the considered metrics. Likewise in the above case, the smallest relative error happens for the metric ϕ_3, as well as the smallest error interval $[Min, Max]$, but now with $k = 5\%$ and $k = 50\%$.

TABLE II: Performance of ICP, ICP-CTSF and SWC-ICP for alignment $0° - 45°$.

Method	Metric	Min		Max	
		Error	***k***	***Error***	***k***
ICP	ϕ_1	0.0178769966	-	0.0178769966	-
	ϕ_3	0.0003192528	-	0.0003192528	-
	ϕ_4	0.0091941769	-	0.0091941769	-
	ϕ_5	0.0252800561	-	0.0252800561	-
ICP-CTSF	ϕ_1	0.0176847562	5%	0.0178832039	75%
	ϕ_3	0.0003124164	5%	0.0003194748	75%
	ϕ_4	0.0091972439	75%	0.0092677951	10%
	ϕ_5	0.0250082513	5%	0.0252888326	75%
SWC-ICP	ϕ_1	0.0176695492	25%	0.0179247959	75%
	ϕ_3	0.0003118788	25%	0.0003209641	75%
	ϕ_4	0.0091550806	50%	0.0092616098	5%
	ϕ_5	0.0249867506	25%	0.0253476385	75%



(a) (b)

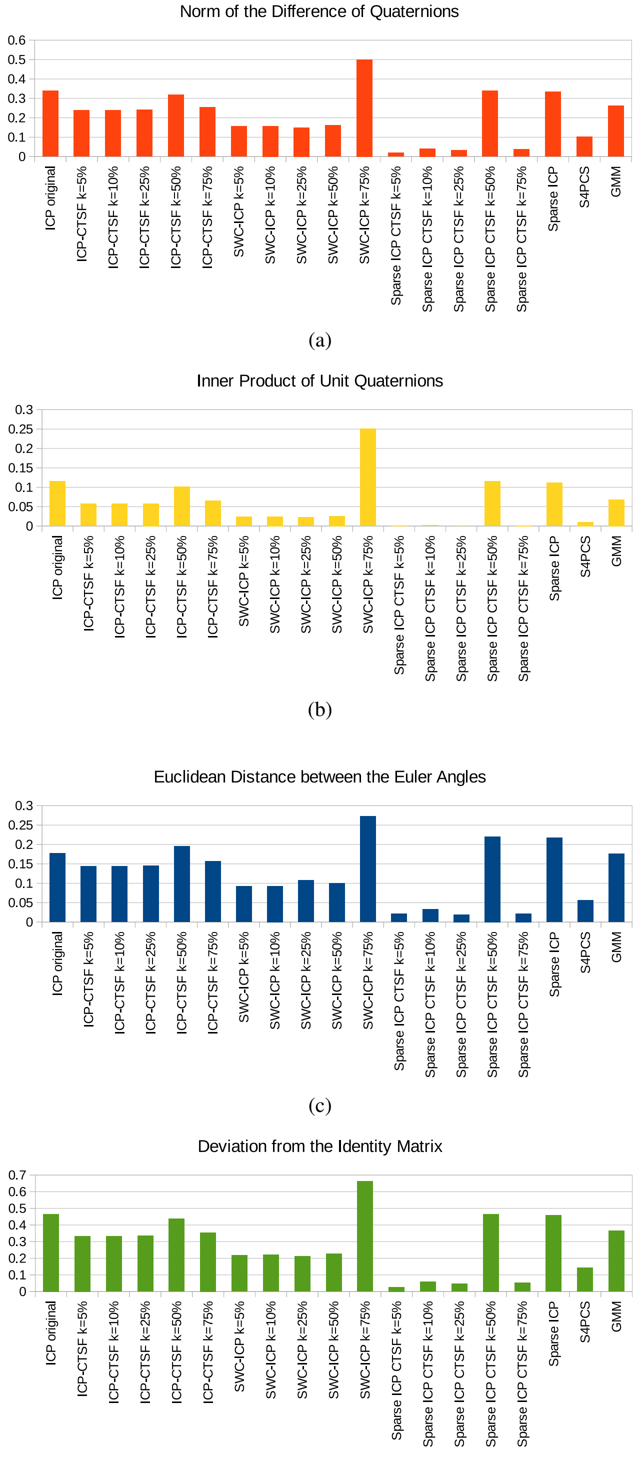
Fig. 6: Visualization of the best cases reported in Table I: (a) Sparse ICP. (b) Sparse ICP CTSF with $k = 25\%$.

TABLE III: Sparse ICP CTSF rotation relative error for alignment $45° - 90°$.

Method	Metric	Min		Max	
		Relative Error	***k***	***Relative Error***	***k***
Sparse ICP CTSF	ϕ_1	0.0136222	5%	0.2399373	50%
	ϕ_3	0.0003704	5%	0.1151393	50%
	ϕ_4	0.0186458	25%	0.2104903	50%
	ϕ_5	0.009632	5%	0.16470541	50%

Figure 8(a) allows to visually check the alignment obtained by the Sparse ICP CTSF using $k = 5\%$. Also, Figure 8(b) allows to compare that result with the Sparse ICP registration in order to confirm that, different from the case $0° - 45°$, the alignment of the former is really better than the alignment generated by the latter in this case.

The third registration test is the hardest one, since there is a 90° variation between the two point sets. It implies also in a smaller overlapping, which is a complicating factor in rigid



(a)

(b)

(c)

(d)

Fig. 7: Rotation error for the registration techniques (Trimming parameter $\tau = 0$), for the case $45° - 90°$, computed using metric: (a) ϕ_1. (b) ϕ_3; (c) ϕ_4; (d) ϕ_5.

registration. All methods failed to obtain a correct registration in this case. The minimum and maximum absolute errors of the best two methods are reported in Table IV for the case 90°-180°.

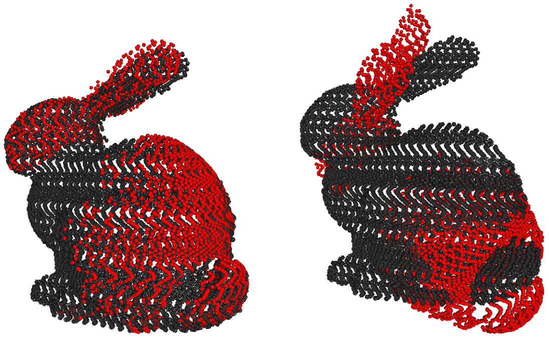


Fig. 8: Case $45° - 90°$: (a) Registration obtained with Sparse ICP CTSF using $k = 5\%$. (b) Final alignment generated by Sparse ICP.

TABLE IV: Best two methods, with minimum and maximum errors computed by each metric for the alignment $90° - 180°$.

Method	Metric	Min		Max	
		Error	*k*	*Error*	*k*
ICP CTSF	ϕ_1	0.1244098475	75%	0.81487218	100%
	ϕ_3	0.0154772929	75%	0.664016846	100%
	ϕ_4	0.3132339459	50%	0.6542840582	75%
	ϕ_5	0.1752597383	75%	0.9418682848	100%
SWC-ICP	ϕ_1	0.1318543928	75%	0.9498184948	1%
	ϕ_3	0.0173850661	75%	0.9021556037	1%
	ϕ_4	0.161463837	75%	0.5907273744	1%
	ϕ_5	0.1856579836	75%	0.9952022544	1%

The minimum rotation error is achieve by ICP CTSF, with $k = 75\%$, in the metric ϕ_3. However, Figure 9 shows that the obtained alignment is not correct.

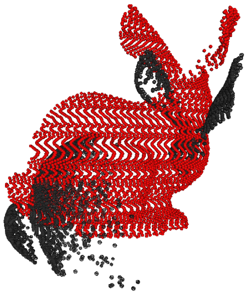
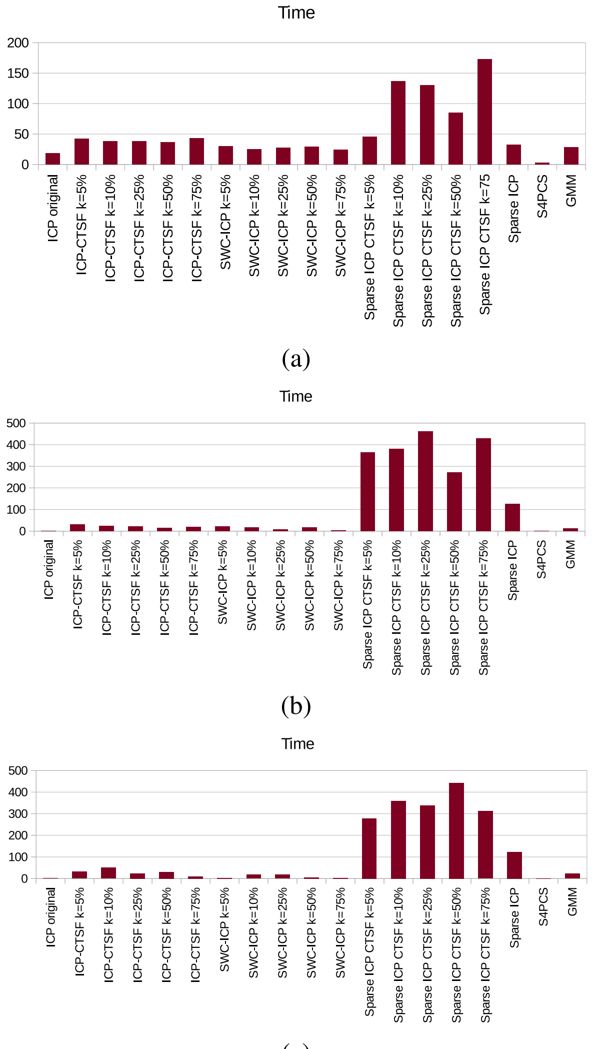


Fig. 9: Best registration (ICP CTSF, with $k = 75\%$) obtained for the 90°-180° case. All the registration methods of Section III get incorrect results in this case.

Figure 10 shows the CPU time for the execution of each technique in the experiments reported in this section. We can notice that Sparse ICP CTSF computational time is much longer. Also, Sparse ICP CTSF is followed by the Sparse ICP, ICT-CTSF and SWC-ICP in terms of computational time. So, despite of the good performance of Sparse ICP CTSF in cases 0°-45° and 45°-90°, its computational time is higher.



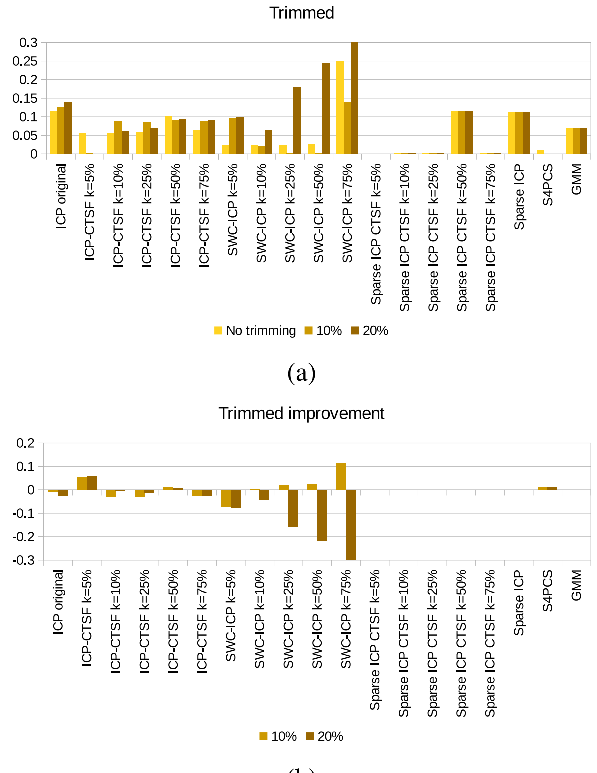
(a)

(b)

(c)

Fig. 10: CPU time in seconds obtained for each method when computing alignment for: (a) 0°-45°. (b) 45°-90°. (c) 90°-180°.

The influence of the trimming parameter can be discussed through Figure 11, when considering the registration for 45°-90°. We calculate the rotation error using function ϕ_3, shown on Figure 11a. To complement the information, Figure 11b pictures the error variation. We shall observe that the SWC-ICP with $k = 75\%$ undergoes the larger registration improvement (0.112109), for trimming $\tau = 10\%$, but it also suffers the larger error increasing if $\tau = 20\%$. On the other hand, the Sparse ICP CTSF with $k = 5\%$, that achieves the smallest error without trimming, remains almost unchanged once it gets a difference of $-1.411663 \cdot 10^{-6}$ with both $\tau = 10\%$ and $\tau = 20\%$. However, the SWC-ICP, that gets the second place in the 45°-90° alignment, increases its efficiency for trimming 10% and $k = 25\%, 50\%, 75\%$ but decreases for all the other cases when incorporating trimming. Therefore, it is not possible to figure out a tendency to the influence of the trimming procedure in the registration error.

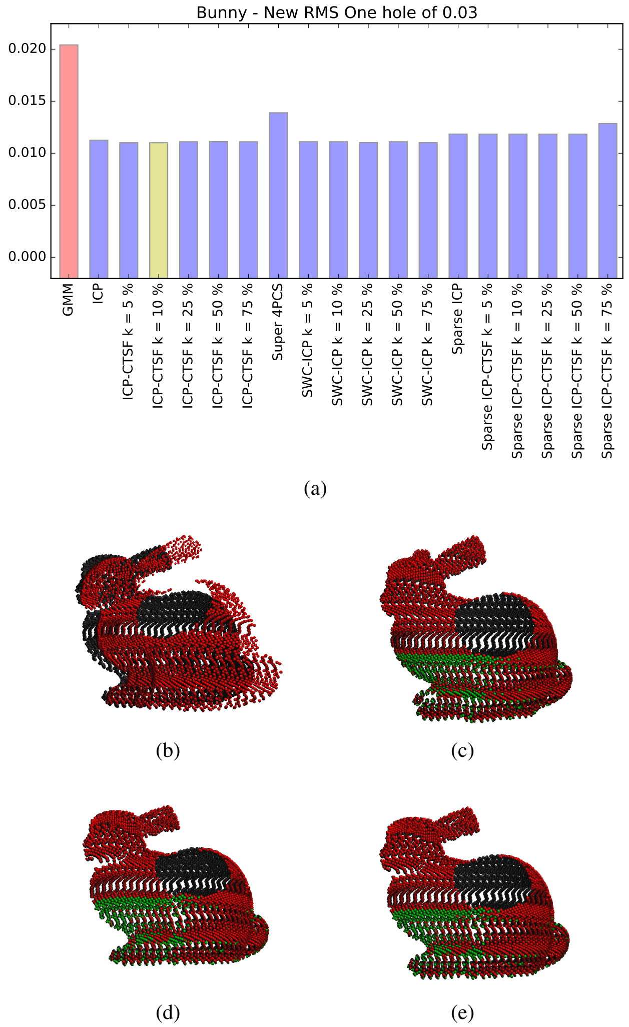


(a)

(b)

Fig. 11: Influence of the trimming parameter in the case 45°-90°. (a) Rotation error according to ϕ_3. (b) Visualization of the error variations, with respect to the trimming null.

Missing data is simulated by taking the 0° view of the Bunny and moving it using the rotation (45°) and translation available in the web site [12]. In this way, we have the ground truth for the correspondence set which allows to compare the RMS ($\sqrt{e^2\,(R, \mathbf{t})}$) of the techniques without ambiguities. Then, we remove a set of points inside a ball centered in a specific point in the cloud, with $radius = 0.03$, and update the correspondence set. Figure 12.(b) shows the two clouds before alignment. Figure 12.(a) allows to analyse the performance of the registration techniques, regarding the RMS, against missing. In this figure we indicate in yellow the best technique and in magenta de worst one. It is noticeable the GMM gets the larger RMS error while ICP-CTSF with $k = 10\%$ presents outstanding performance. With exception of GMM, the other methods perform close one to each other. Table V allows to get a better idea about the numeric differences between the RMS errors. The Sparse ICP-CTSF $k = 5\%$ and Sparse ICP, which were the best methods in the previous experiments, achieve the $14th$ and $16th$ place in the RMS rank for incomplete Bunny data. We shall be careful because the difference between the Sparse ICP and the best technique in Table 12 is approximately 0.0008, which is not too important, considering that the clouds are normalized in the unitary cube. The Figures 12.(c)-(e) agrees with this observation once it is hard to notice differences between the alignments. However, the relative decrease of performance of Sparse ICP-CTSF and Sparse ICP may indicate some sensitivity against incomplete data, which could impact their efficiency for frame-to-frame registration.



(a)

(b) (c)

(d) (e)

Fig. 12: (a) RMS for missing data simulation (b) Point cloud 0° and its version rotated/translated and corrupted by removing data. (c) Best alignment obtained by ICP-CTSF with $k = 10\%$. (d) Alignment generated by Sparse ICP-CTSF with $k = \%5$. (e) Worst alignment obtained by Sparse ICP.

TABLE V: RMS for alignment of clouds with missing data.

Rank	Method	RMS
1	ICP-CTSF $k = 10\%$	0.0110090321
14	Sparse ICP-CTSF $k = 5\%$	0, 0118308936
16	Sparse ICP	0, 0118385019
19	GMM	0, 0204105014

Table VI summarizes the main results reported in this section. All the rotation errors presented in Table VI are computed using the ϕ_3 (Section IV-B) once the discussion related to Table I, Figures 5 and 6 points out this metric as the more appropriate.

TABLE VI: Conclusions of the point cloud registration experiments.

Test	Best Method	Error
$0° - 45°$	Sparse ICP	$9.03625 \cdot 10^{-7}$ (ϕ_3)
$45° - 90°$	Sparse ICP CTSF, with $k = 5\%$	0.000370 (ϕ_3)
$90° - 180°$	All fail	—
Trimming	No conclusion	—
Missing data	ICP-CTSF, with $k = 10\%$	0.0110090321 (RMS)

B. Frame-To-Frame Registration

According to Section V-A, the best methods in the performed experiments are Sparse ICP, Sparse ICP CTSF, without considering incomplete data, and ICP-CTSF otherwise, with the parameters reported in Table VI. In this section, we must check the obtained conclusions, but now in the frame-to-frame registration, which composes the second sequence of experiments. These tests are executed using the 640×480 pixels of the frames extracted from the #03118 video, as reported at the beginning of this section. The image resolution yields a depth array with 307200 elements which increase the computational cost of the registration algorithms. Therefore, we uniformly sampled each frame of the video, with sampling rates $r = 8$, to reduce the total number of points, generating new video sequence V. This way, each frame C_m, $1 \leq m \leq |V|$ becomes a matrix $C_m \in \mathbb{R}^{M_1 \times M_2 \times 4}$, where $M_1 = integer\,(640/r)$, $M_2 = integer\,(480/r)$, $C_m\,(i, j, 1)\,, \cdots, C_m\,(i, j, 3)$ hold the R,G and B channels, respectively, and $C_m\,(i, j, 4)$ corresponds to the depth information, captured with 8-bit resolution.

The sequence #03118 was chosen because of how easy it is to segment the background. In this video, the sign is the only meaningful object in the scene, with respect to the depth information (see Figure 4). The grass in the background is too deep to be captured and yields null depth values. Hence, we take the set $S_m = \{(i, j, C_m\,(i, j, 4))\,;\, C_m\,(i, j, 4) > 0,\ 1 \leq i \leq M_1 \text{ and } 1 \leq j \leq M_2\}$ and interpret it as a point cloud in $\mathbb{R}^3$. Besides, a temporal sampling was made, selecting one frame at each ς consecutive frames. This approach pushes the difficulty of the registration, as a simulated larger camera movement, that implies in a smaller overlapping region. So, we set $P = S_{\varsigma \cdot i}$, $Q = S_{\varsigma \cdot (i+1)}$ as the pair source/target in each iteration of the frame-to-frame registration that generates the pair $(R_{\varsigma \cdot i}, \mathbf{t}_{\varsigma \cdot i})$ that best aligns the source cloud $S_{\varsigma \cdot i}$ with the target one $S_{\varsigma \cdot (i+1)}$.

The root mean squared error (RMS) after the registration of the clouds $S_{\varsigma \cdot i}$ and $S_{\varsigma \cdot (i+1)}$ is given by:

$$RMS_{reg}(S_{\varsigma \cdot i}, S_{\varsigma \cdot (i+1)}) = \sqrt{e^2\,(R_{\varsigma \cdot i}, \mathbf{t}_{\varsigma \cdot i})}, \qquad (40)$$

with e^2 being the error computed by expression (5). The equation (40) allows to compute the average root mean squared error, denoted by $MRMS$, through the expression:

$$MRMS\,(\varsigma, V) = \left(\frac{1}{\frac{|V|}{\varsigma}}\right) \sum_{i=0}^{\frac{|V|}{\varsigma}-1} \mathit{RMS}_{reg}(S_{\varsigma\cdot i}, S_{\varsigma\cdot(i+1)}), \tag{41}$$

which can be used to measure the quality of the whole sequence registration.

Since the choice of the parameter k of the ICP-CTSF, the SWC-ICP and the modified Sparse ICP with CTSF impacts on the results, we show how they change with $k = 75\%$, $k = 50\%$, $k = 25\%$, $k = 10\%$ and $k = 5\%$ of the total number of points. All methods were set with the same parameters of the previous experiment.

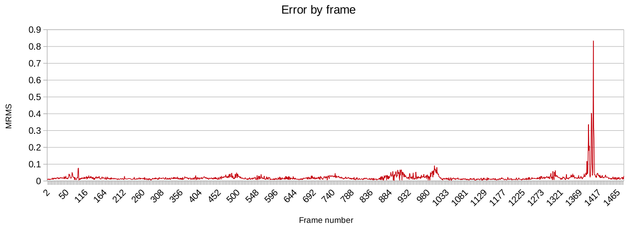
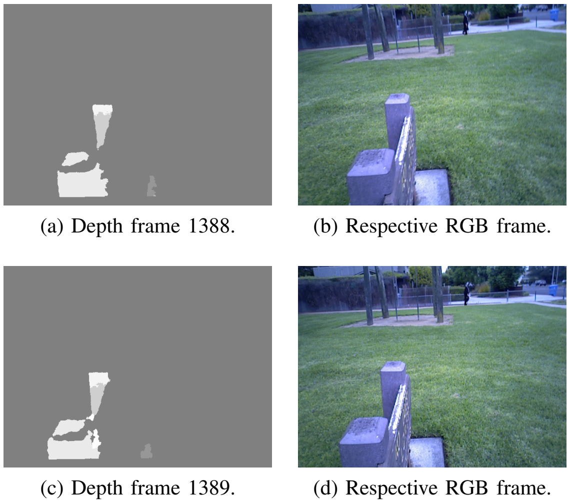


Fig. 13: RMS error by frame of the original ICP on the sequence #03118.

From the Figure 13, which shows the registration error along the video sequence for ICP, we notice that the higher errors occur near the end of the video sequence, where there is a rough movement unlike in the rest of the video. The same happens for all the considered methods. In the corresponding frames, there is another complication because of the low number of points that the depth sensor was able to sample, increasing the chance of a bad alignment. Figure 14 illustrates this case.



(a) Depth frame 1388.

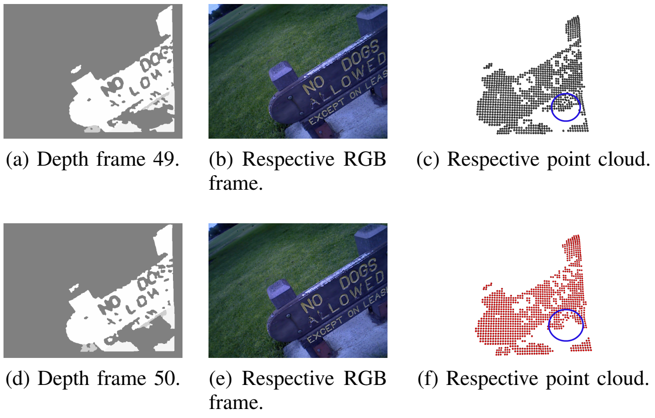
(b) Respective RGB frame.

(c) Depth frame 1389.

(d) Respective RGB frame.

Fig. 14: A pair of depth frames (14a, 14c) and their respective RGB frames (14b, 14d). Note the small area in the depth frames that the sensor was able to capture, yielding fewer points than those of Figure 4a, 4b and 4c, in comparison.

Missing data is a frequent problem when using raw depth data, due to uncertainty caused by reflections in the acquisition process. Figure 15 shows an example of a pair, in which one of the point clouds (Figure 15f) misses some points.



(a) Depth frame 49.

(b) Respective RGB frame.

(c) Respective point cloud.

(d) Depth frame 50.

(e) Respective RGB frame.

(f) Respective point cloud.

Fig. 15: A pair of depth frames (15a, 15d) and their respective RGB frames (15b, 15e) and point clouds (15c, 15f).

Figure 16 shows the $MRMS$ obtained for each method when varying the temporal sampling parameter ς and fixing the spacial sampling $r = 8$. We notice that the $MRMS$ errors of the Sparse ICP and the Sparse ICP CTSF are the highest, contrasting with the results of the previous experiments. Table VII reports the lower $MRMS$ values for results in Figure 16a with ICP-CTSF having the best score in this experiment for $k = 75\%$.

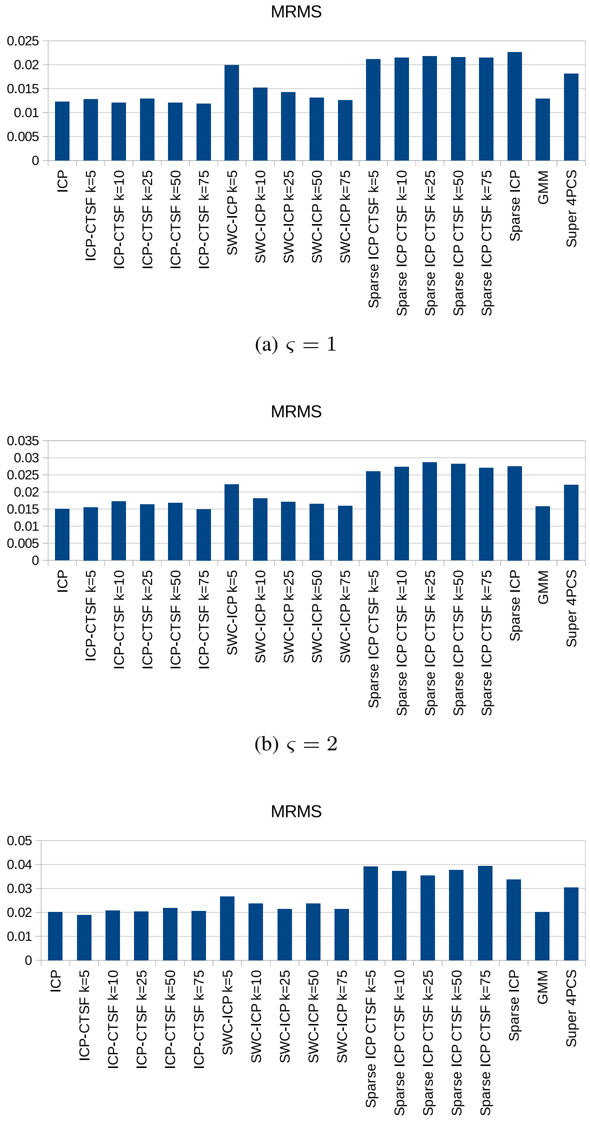
TABLE VII: Best methods, respect to MRMS, for frame-to-frame registration of video #03118 with $\varsigma = 1$ and $r = 8$.

Method	MRMS	Rank
ICP	0,0122449244	Fourth
ICP-CTSF $k = 10\%$	0,0121419904	Third
ICP-CTSF $k = 50\%$	0,0120212500	Second
ICP-CTSF $k = 75\%$	0,0118194802	First

Table VII indicates that variation of the parameter k does not have much effect for ICP-CTSF. In fact, Figure 16 shows that the same happens for the other methods, except for a small trend on the SWC-ICP, where smaller values of k yield higher $MRMS$ errors.

Figure 17 shows the $MRMS$ of the methods when an image sampling rate $r = 16$ is used. In this case we fixed the temporal sampling as $\varsigma = 1$, i.e., every frame i is registered with its consecutive $i+1$. A change of scale is perceived when comparing with Figure 16a, as some methods almost doubled its error. However, this result is expected, as with a higher image sampling, the pixels (and corresponding points) are farther from each other. Points without an exact correspondent, then, will increase the error value.

Since the $MRMS$ values presented some inconsistencies with the previous experiment regarding the Sparse ICP and Sparse ICP CTSF, we decided to discard the segment at the end of the video where all methods perform bad. So, we take



(a) $\varsigma = 1$

(b) $\varsigma = 2$

(c) $\varsigma = 4$

Fig. 16: Variation of the temporal sampling. All the three cases have video sampling rates $r = 8$. Note how the scale changes as ς increases.

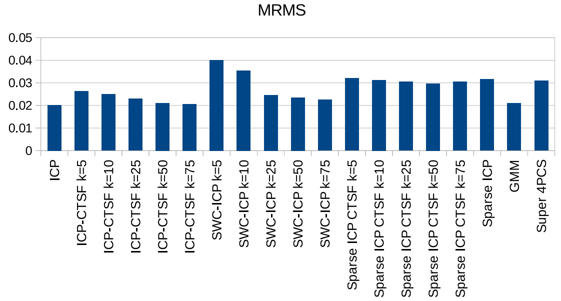


Fig. 17: MRMS of all methods using video sample $r = 16$, with $\varsigma = 1$.

the first 820 frames and recompute the $MRMS$ for the registration algorithms. In order to allow a more complete analysis and comparison with the $MRMS$ for the whole video, we report in Table VIII the $MRMS$ for both experiments. When comparing the values in the second and third columns, we notice that all methods improve the $MRMS$ if only the first 820 frames are used. However, the Sparse ICP and Sparse ICP-CTSF errors dropped to half when we took the first 820 frames only. This fact points towards the sensitivity of these methods against incomplete (and missing) data problems shown in Figures 14 and 15. In the tests of Section V-A, we figure out a similar conclusion when anaysing the results of Table VI. So, if we take a subsequence of frames free of this problem, we expect outstanding performance with Sparse ICP and Sparse ICP CTSF in these cases.

TABLE VIII: $MRMS$ for frame-to-frame registration of video #03118 with $\varsigma = 1$ and $r = 8$. Second column gives the MRMS using the complete video and the third column using the first 820 frames.

Method	MRMS (all)	MRMS (820)
ICP	0.0122449244	0.0097785502
ICP-CTSF $k = 5\%$	0.0127517351	0.0091452435
ICP-CTSF $k = 10\%$	0.0121419904	0.0090946880
ICP-CTSF $k = 25\%$	0.0127106841	**0.0090196982**
ICP-CTSF $k = 50\%$	0.0120212500	0.0092728001
ICP-CTSF $k = 75\%$	**0.0118194802**	0.0092269833
SWC-ICP $k = 5\%$	0.0191444196	0.0167142512
SWC-ICP$k = 10\%$	0.0146610187	0.0115323086
SWC-ICP$k = 25\%$	0.0137439724	0.0100330277
SWC-ICP$k = 50\%$	0.0129120949	0.0101483156
SWC-ICP$k = 75\%$	0.0125843444	0.0101779295
Sparse ICP-CTSF $k = 5\%$	0.0213255462	0.0102627931
Sparse ICP-CTSF $k = 10\%$	0.0214605022	0.0112728029
Sparse ICP-CTSF $k = 25\%$	0.0227560495	0.0106664626
Sparse ICP-CTSF $k = 50\%$	0.0220794083	0.0109903911
Sparse ICP-CTSF $k = 75\%$	0.0193417601	0.0107762084
Sparse ICP	0.0200493252	0.0108378611
GMM	0.0128413983	0.0104018284
Super 4PCS	0.0174477538	0.0137338492

We can use a visual inspection to check this conclusion. In the beginning of the video #03118, we notice less occurrence of incomplete data as shown in Figures 18a-d. So, we simulate an attempt to reconstruct the objects using the frames 1 to 4. Specifically, the resulting registration of the pairs (1,2), (1,3) and (1,4) were overlapped.

Figure 19 shows the obtained result. The Sparse ICP CTSF with $k \in \{5\%, 10\%\}$ produces an image cleaner than other methods, like the blurred image produced from the ICP-CTSF with $k = 50\%$. In Figure 20 we highlight the fact that the result obtained by Sparse ICP CTSF with $k = 5\%$ is much better then the ICP-CTSF with $k = 50\%$, as the points are completely overlapped in the former (Figures 20c and 20d), differently from the latter.

Hence, the visual inspection considering the pairs (1,2), (1,3), (1,4) and the MRMS in Table VIII indicates that Sparse ICP and Sparse ICP CTSF suffer the influence of incomplete/missing data. If true, considering the requirements of the frame-to-frame registration, they could not be not

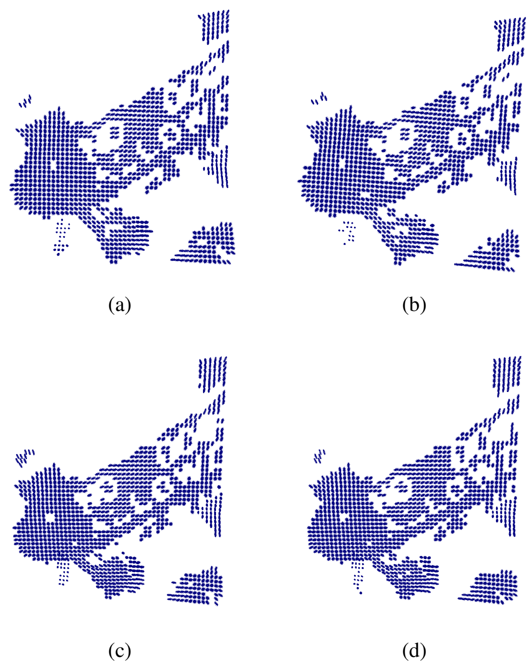


Fig. 18: (a) Target object in frame 1. (b) Visualization of the point cloud in frame 2. (c) Point cloud in frame 3. (d) Target point cloud in frame 4.

recommended for such applications. On the other hand, it seems that the ICP-CTSF is more reliable for this application considering that its performance is less sensitive to the mentioned problems, as observed in Tables VI and VIII. Next, we undertake new experiments to evaluate the focused techniques for frame-to-frame registration with availabe groud truth transformation, in order to check these conclusions.

C. Frame-To-Frame Alignment Tests with Ground-Truth

The frame-to-frame registration experiments of Section V-B was not conclusive with respect to the best technique for this application. Hence, in this section, we perform frame-to-frame registration tasks using public data sets, with ground truth available [14], [46], to complete the analysis. The geometry behind the data acquisition process is pictured on Figure 21, which represents the world reference system, denoted by W, and two others coordinate systems, named W_1 and W_2, attached to the camera and defining its location and orientation respect to the system W. Besides, we have a point $\mathbf{p} \in \mathbb{R}^3$, which has coordinates:

$$[\mathbf{p}]_{W_1} = (\alpha_1, \alpha_2, \alpha_3)^T \,, \quad [\mathbf{p}]_{W_2} = (\beta_1, \beta_2, \beta_3)^T \,,$$

respect to the systems W_1 and W_2, respectively. Also, let rotations R_1, R_2, and translations $\mathbf{t}_1$, $\mathbf{t}_2$, be such that:

$$[\mathbf{p}]_W = \mathbf{t}_1 + R_1 \, [\mathbf{p}]_{W_1} \,, \tag{42}$$

$$[\mathbf{p}]_W = \mathbf{t}_2 + R_2 \, [\mathbf{p}]_{W_2} \,, \tag{43}$$

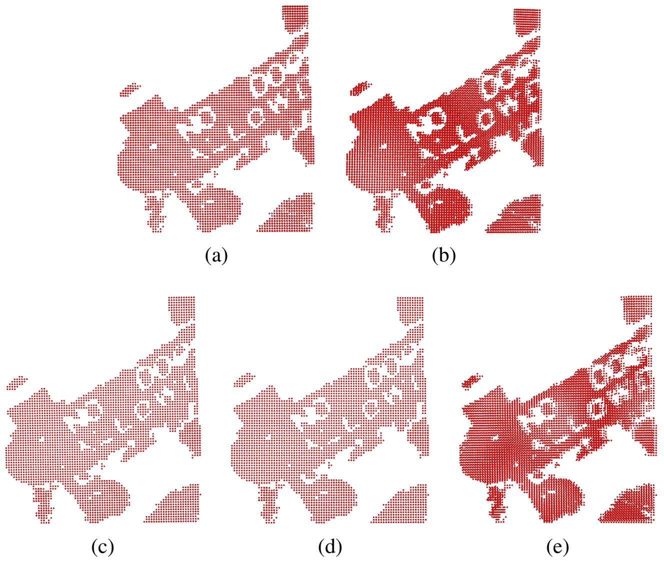


Fig. 19: Overlapping of the registered frames 1-4 by different algorithms. (a) Original ICP. (b) ICP-CTSF $k = 50\%$. (c) Sparse ICP CTSF $k = 5\%$. (d) Sparse ICP CTSF $k = 10\%$. (e) Sparse ICP CTSF $k = 75\%$.

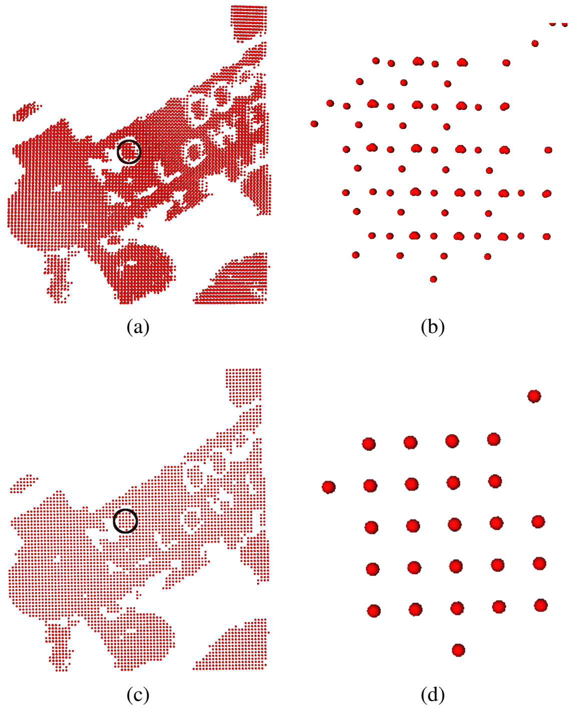


Fig. 20: Zoom-in the selected area showing how the points do not coincide in the ICP-CTSF with $k = 50\%$, producing a blurred image observed in Figure 19b. The same does not happen with the Sparse ICP CTSF with $k = 5\%$. (a) ICP-CTSF $k = 50\%$. (b) Zoon-in the selected region. (c) Sparse ICP CTSF $k = 5\%$. (d) Selected region in detail.

as well as the rotation $R_{1,2}$ and translation $\mathbf{t}_{1,2}$, computed by a registration algorithm, which allows to write:

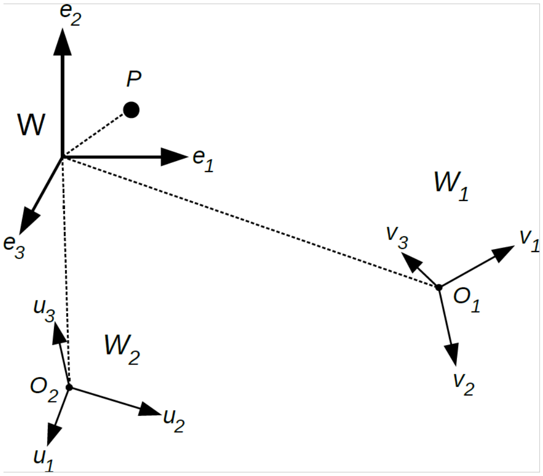


Fig. 21: Global (W) and camera reference systems for two consecutive frames.

$$[\mathbf{p}]_{W_2} = \mathbf{t}_{1,2} + R_{1,2}\,[\mathbf{p}]_{W_1}\,. \tag{44}$$

Consequently, given a point cloud $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{n_P}\} \subset \mathbb{R}^3$, the registration error can be computed as:

$$e^2\left(R_{1,2}, \mathbf{t}_{1,2}\right) = \frac{1}{n_P}\sum_{i=1}^{n_P}\left\|[\mathbf{p}_i]_{W_2} - \left(R_{1,2}\,[\mathbf{p}_i]_{W_1} + \mathbf{t}_{1,2}\right)\right\|_2^2. \tag{45}$$

Also, considering expressions (42)-(43), a simple algebra shows that:

$$[\mathbf{p}]_{W_2} = R_2^{-1}\left(\mathbf{t}_1 - \mathbf{t}_2\right) - R_2^{-1}R_1\,[\mathbf{p}]_{W_1}\,. \tag{46}$$

We interpret the first term of the right-hand side of expression (46) as the ground truth translation and the matrix $R_2^{-1}R_1$ as the ground truth rotation, that can be used to quantify the precision of the rigid transformation given by expression (44). The ground truth rigid transformation can be computed if we know the rotations and the translations that appear in equations (42)-(43). The database available in [14] provides these information for the video 'freiburg2_xyz', that contains very clean data for debugging translations, and for the video 'freiburg2_rpy' which contains suitable data for debugging rotations [14]. The former has 3615 RGB-D frames while the later encompasses 3221 RGB-D images, both with resolution 640×480 pixels.

We work analogously to the beginning of Section V-B, by setting $r = 16$ and $\varsigma = 3$, to generate the set $S_m = \{(i, j, C_m(i, j, 4));\ C_m(i, j, 4) > 0,\ 1 \leq i \leq M_1$ and $1 \leq j \leq M_2\}$ and perform the necessary transformation [47] to convert the 2D depth data to 3D point clouds in the reference system W_m, that defines the Kinect position and orientation when frame C_m was acquired. The result is a point cloud $S_{W_m} = \left\{[\mathbf{p}_{ij}]_{W_m}\right\}$, where $[\mathbf{p}_{ij}]_{W_m}$ is the coordinate vector of the point $\mathbf{p}_{ij} = (i, j, C_m(i, j, 4))$ respect to the coordinate system W_m .

So, we set $P = S_{W_{\varsigma\cdot i}}$, $Q = S_{W_{\varsigma\cdot(i+1)}}$ as the pair source/target in each iteration of the frame-to-frame registration that generates the pair $\left(R_{\varsigma\cdot i,\varsigma\cdot(i+1)}, \mathbf{t}_{\varsigma\cdot i,,\varsigma\cdot(i+1)}\right)$ that best aligns the source cloud P with the target one Q.

Let us consider the Figure 21, with W_1 replaced to $W_{\varsigma\cdot i}$ and W_2 replaced to $W_{\varsigma\cdot(i+1)}$. We must notice that the clouds $S_{W_{\varsigma\cdot i}}$ and $S_{W_{\varsigma\cdot(i+1)}}$ correspond to two views of the same scene S, represented in the system $W_{\varsigma\cdot i}$ and $W_{\varsigma\cdot(i+1)}$, respectively. So, in order to use expression (45) to compute the registration error, we must know the ground truth for the matching between clouds $S_{W_{\varsigma\cdot i}}$ and $S_{W_{\varsigma\cdot(i+1)}}$, which is not given in the database. In fact, we know only the ground truth for the transformations between the (camera) reference systems $W_{\varsigma\cdot i}$ and $W_{\varsigma\cdot(i+1)}$ and the world reference system W, denoted by $\left(R_{\varsigma\cdot i}, \mathbf{t}_{\varsigma\cdot i}\right)$ and $\left(R_{\varsigma\cdot(i+1)}, \mathbf{t}_{\varsigma\cdot(i+1)}\right)$, respectively (see Figure 21). So, we can straightforward compare the pair $\left(R_{\varsigma\cdot i,\varsigma\cdot(i+1)}, \mathbf{t}_{\varsigma\cdot i,\varsigma\cdot(i+1)}\right)$, obtained through the registration algorithm, with the ground truth, computed through the two terms in the right-hand side of expression (46). So, let $\mathbf{q}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)}$ be the quaternion that represents the ground truth rotation $(\left(R_{\varsigma\cdot(i+1)}\right)^{-1}R_{\varsigma\cdot i})$, and the vector $\mathbf{t}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)} = \left(R_{\varsigma\cdot(i+1)}\right)^{-1}\left(\mathbf{t}_{\varsigma\cdot i} - \mathbf{t}_{\varsigma\cdot(i+1)}\right)$ which gives the ground truth translation. Hence, we can define the mean rotation error:

$$MRot\left(\varsigma, V\right) = \left(\frac{1}{\frac{|V|}{\varsigma}}\right)\sum_{i=0}^{\frac{|V|}{\varsigma}-1}\phi_3(\mathbf{q}_{\varsigma\cdot i,\varsigma\cdot(i+1)}, \mathbf{q}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)}), \tag{47}$$

where $\mathbf{q}_{\varsigma\cdot i,\varsigma\cdot(i+1)}$ is the quaternion for rotation matrix $R_{\varsigma\cdot i,\varsigma\cdot(i+1)}$, and ϕ_3 is the metric defined by equation (36). Analogously to expression (47), we can define the mean translation error as:

$$MTr\left(\varsigma, V\right) = \left(\frac{1}{\frac{|V|}{\varsigma}}\right)\sum_{i=0}^{\frac{|V|}{\varsigma}-1}\left\|\mathbf{t}_{\varsigma\cdot i,\varsigma\cdot(i+1)} - \mathbf{t}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)}\right\|_2. \tag{48}$$

Moreover, we consider the standard deviations:

$$SRot\left(\varsigma, V\right) = \left[\left(\frac{1}{\frac{|V|}{\varsigma}}\right)\sum_{i=0}^{\frac{|V|}{\varsigma}-1}\left|\phi_3^{\varsigma,i} - MRot\left(\varsigma, V\right)\right|^2\right]^{1/2}, \tag{49}$$

$$STr\left(\varsigma, V\right) = \left[\left(\frac{1}{\frac{|V|}{\varsigma}}\right)\sum_{i=0}^{\frac{|V|}{\varsigma}-1}\left|D\left(\varsigma, i\right) - MTr\left(\varsigma, V\right)\right|^2\right]^{1/2}, \tag{50}$$

where $\phi_3^{\varsigma,i} = \phi_3(\mathbf{q}_{\varsigma\cdot i,\varsigma\cdot(i+1)}, \mathbf{q}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)})$, and $D\left(\varsigma, i\right) = \left\|\mathbf{t}_{\varsigma\cdot i,\varsigma\cdot(i+1)} - \mathbf{t}^g_{\varsigma\cdot i,\varsigma\cdot(i+1)}\right\|_2$, to analyze the statistical significance of the means (47) and (48).

All methods were set with the same parameters of the previous experiments. We start with the sequence 'freiburg2_rpy' and compute expression (47)-(48) with $\varsigma = 3$, whose results are shown in Figure 22. Although, according to the database information [14], in this case we have small translation effects, we decided to show the translation error in Figure 22.(b) in order to complete the analysis. The best techniques are highlighted with yellow bars and the worst with magenta bars. From Figure 22.(a) wee see that Sparse ICP-CTSF with $k = 50\%$ achieve the lowest rotation error while Figure 22.(b) shows the superiority of GMM for translation.

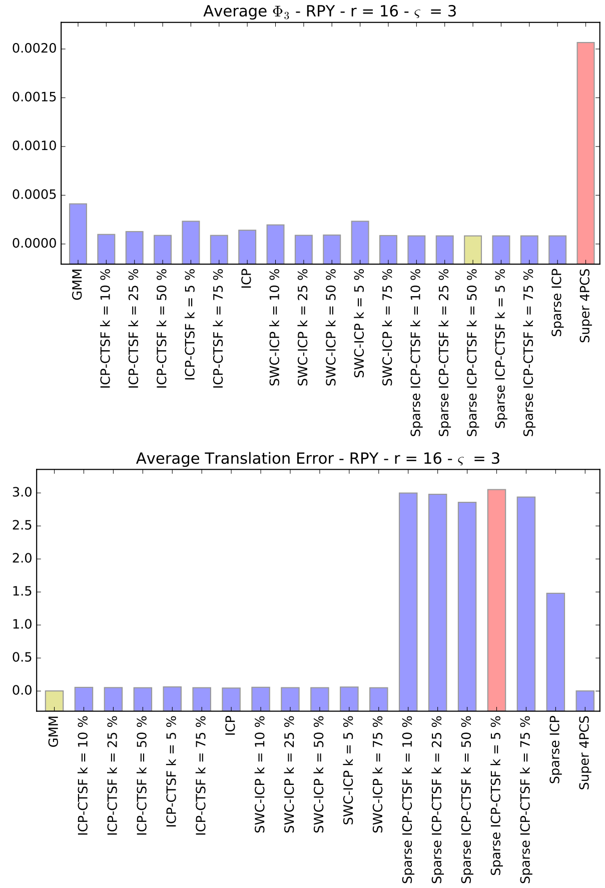


Fig. 22: Results for video ‘freiburg2_rpy’ : (a) Mean rotation error given by expression (47) . (b) Translation error computed by expression (48).

The scale of the mean rotation/translation errors in Figure 22 do not allow to rank the best techniques. To solve this problem, we report in Tables IX-X the best seven methods according to the rotation and translation errors with the corresponding standard deviations calculated by expression (49) and (50), respectively.

Once the metric $\phi_3 \in [0, 1]$, it is straightforward that the second column of Table IX gives the absolute $(MTr \pm STr)$ and relative mean errors $((MTr \pm STr) / \phi_3^{max})$ for rotation. Also, considering that the clouds are normalized in the unitary cube we can also take $MTr \pm STr$ as both the absolute and relative translation error measure.

From Table IX, it is noticeable that Sparse ICP-CTSF and Sparse ICP are the best techniques for rotation, with a small advantage of former with $k = 50\%$ against the latter. We shall highlight that the mean error and standard deviation regarding rotation, reported in Table IX, are of order 10^{-5} and 10^{-4}, respectively, which show that the methods perform well in this item.

TABLE IX: Best seven methods for video ‘freiburg2_rpy’ according to the rotation average (equation (47)) with standard deviations given by expression (49).

Method	MRot ± SRot
Sparse ICP-CTSF k = 50 %	$8.31308 \times 10^{-5} \pm 1.01305 \times 10^{-4}$
Sparse ICP-CTSF k = 75 %	$8.31353 \times 10^{-5} \pm 1.01316 \times 10^{-4}$
Sparse ICP-CTSF k = 25 %	$8.31381 \times 10^{-5} \pm 1.01308 \times 10^{-4}$
Sparse ICP-CTSF k = 5 %	$8.31438 \times 10^{-5} \pm 1.01313 \times 10^{-4}$
Sparse ICP	$8.31525 \times 10^{-5} \pm 1.01330 \times 10^{-4}$
Sparse ICP-CTSF k = 10 %	$8.31545 \times 10^{-5} \pm 1.01312 \times 10^{-4}$
SWC-ICP k = 75 %	$8.61504 \times 10^{-5} \pm 1.05042 \times 10^{-4}$

TABLE X: Best seven methods for video ‘freiburg2_rpy’ according to translation average (equation (48)) with standard deviations given by expressions (50).

Method	MTr ± STr
GMM	0.00168 ± 0.00099
Super 4PCS	0.00168 ± 0.00099
ICP	0.04537 ± 0.05005
SWC-ICP k = 75 %	0.04970 ± 0.05068
ICP-CTSF k = 50 %	0.04992 ± 0.05470
ICP-CTSF k = 75 %	0.05045 ± 0.05465
SWC-ICP k = 50 %	0.05058 ± 0.05339

Regarding to translation, the best techniques reported in Table X are GMM and Super 4PCS. They work equivalent in the translation estimation once both achieve the same values for MTr and for the standard deviation STr, in the precision used in Table X. The Sparse ICP-CTSF and Sparse ICP do not appear in the list of seven better methods.

The next tests show the performance of the registration methods when using the video ’freiburg2_xyz’. Although the data set documentation [14] assures that this video is indicate for debugging translations, we reported both the rotation (Figure 23.(a)) and translation errors (Figure 23.(b)) to complete the analysis.

Likewise in the last tests, Sparse ICP-CTSF with $k = 50\%$ is the best methods for rotation as emphasized by the yellow bar in Figure 23.(a). In Table XI we also report the best seven methods according to the rotation mean errors for tests with video ‘freiburg2_xyz’ with the corresponding standard deviations. Considering the error mean $MRot$ and standard deviation $SRot$ we see that the performance of Sparse ICP is close to Sparse ICP-CTSF with $k = 50\%$ while both perform very well if we take into account that $\phi_3 \in [0, 1]$.

Regarding the errors for translation for video ‘freiburg2_xyz’ shown in Figure 23.(b), we notice that GMM outperforms all the other methods, likewise in the previous video. Also, the first column of Table XII shows that the GMM and Super 4PCS work equivalently in the precision used in this table.

Therefore, the results obtained for videos ’freiburg2_rpy’ and ’freiburg2_xyz’ show that Sparse ICP-CTSF with $k =$

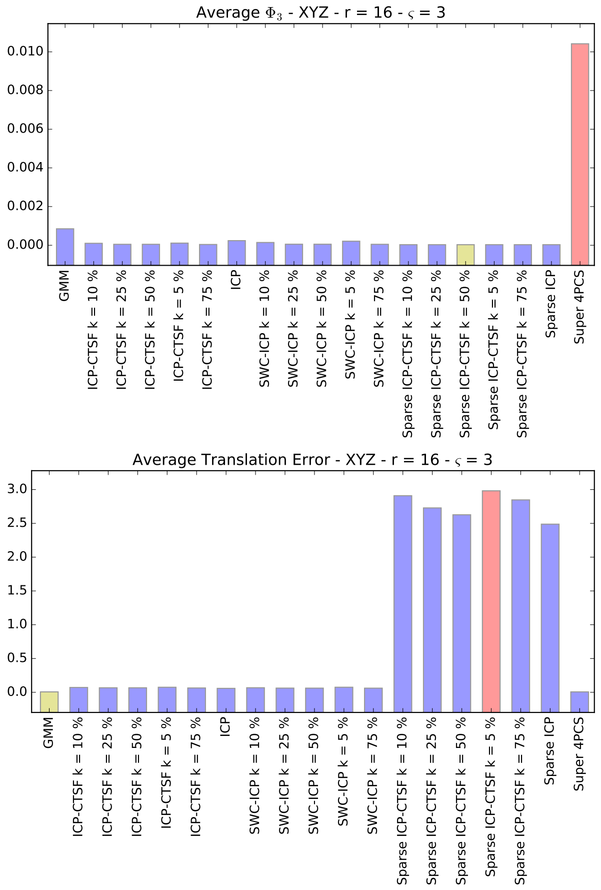


Fig. 23: Results for video 'freiburg2_xyz' : (a) Mean rotation error given by expression (47). (b) Translation error computed by expression (48).

TABLE XI: Best seven methods for video 'freiburg2_xyz' according to the rotation average (equation (47)) with standard deviations given by expressions (49).

Method	**MRot ± SRot**
Sparse ICP-CTSF k = 50 %	$2.81713 \times 10^{-5} \pm 3.91618 \times 10^{-5}$
Sparse ICP-CTSF k = 10 %	$2.81726 \times 10^{-5} \pm 3.91613 \times 10^{-5}$
Sparse ICP-CTSF k = 75 %	$2.81734 \times 10^{-5} \pm 3.91645 \times 10^{-5}$
Sparse ICP	$2.81739 \times 10^{-5} \pm 3.91651 \times 10^{-5}$
Sparse ICP-CTSF k = 25 %	$2.81769 \times 10^{-5} \pm 3.91652 \times 10^{-5}$
Sparse ICP-CTSF k = 5 %	$2.81778 \times 10^{-5} \pm 3.91715 \times 10^{-5}$
ICP-CTSF k = 75 %	$3.94410 \times 10^{-5} \pm 1.03781 \times 10^{-4}$

TABLE XII: Best seven methods for video 'freiburg2_xyz' according to translation average (equation (48)) with standard deviations given by expressions (50).

Method	**MTr ± STr**
GMM	0.00610 ± 0.00301
Super 4PCS	0.00610 ± 0.00301
ICP	0.05807 ± 0.06520
SWC-ICP k = 75 %	0.06107 ± 0.06515
SWC-ICP k = 50 %	0.06140 ± 0.06729
SWC-ICP k = 25 %	0.06182 ± 0.06727
ICP-CTSF k = 75 %	0.06378 ± 0.07572

50% is the best technique for rotation while GMM got outstand results for translation estimation. In order to put all this together to try a final conclusion, we apply the transformation (46) to the set P and take the correspondence relation (2) as the ground truth matching in order to compute the $MRMS$ error using equations (40)-(41) and (45). Tables XIII-XIV reports the obtained results. It is noticeable that ICP-CTSF with $k = 75\%$ achieves the best $MRMS$ for both tables. If we return to Figures 22 and 23 we observe that ICP-CTSF was among the best methods, as we can confirm by Tables XII, XI, and X.

TABLE XIII: Best seven methods for video 'freiburg2_rpy' according to the $MRMS$ given by expression (41).

Method	**MRMS**
ICP-CTSF k = 75 %	0.01350 ± 0.01077
ICP-CTSF k = 50 %	0.01350 ± 0.01081
ICP-CTSF k = 10 %	0.01356 ± 0.01089
SWC-ICP k = 75 %	0.01366 ± 0.01084
ICP-CTSF k = 25 %	0.01370 ± 0.01173
SWC-ICP k = 50 %	0.01372 ± 0.01085
ICP-CTSF k = 5 %	0.01373 ± 0.01122

TABLE XIV: Best seven methods for video 'freiburg2_xyz' according to the $MRMS$ given by expression (41).

Method	**MRMS**
ICP-CTSF k = 75 %	0.01559 ± 0.01490
ICP-CTSF k = 50 %	0.01562 ± 0.01494
ICP-CTSF k = 25 %	0.01562 ± 0.01494
ICP-CTSF k = 10 %	0.01565 ± 0.01508
SWC-ICP k = 75 %	0.01576 ± 0.01503
SWC-ICP k = 50 %	0.01576 ± 0.01507
ICP-CTSF k = 5 %	0.01579 ± 0.01542

If we assemble the results presented in Figures 22-23 and and TablesIX-41, we conclude that the best technique for rotation estimation is Sparse ICP-CTSF with $k = 50\%$ while GMM outperforms the other techniques for translation computation. However, considering rotation and translation together in the $MRMS$, the ICP-CTSF with $k = 75\%$ obtains the best results. We shall remember that in the end of Section V-B we pointed out that ICP-CTSF is more reliable for frame-to-frame registration applications considering that its performance seems to be less sensitive against missing/incomplete data, as also reported in Table VI. Besides, we must take into account that ICP-CTSF with $k = 75\%$ is among the seven best methods reported in Tables XII, XI, and X. So, all this together favor the ICP-CTSF as the best technique for frame-to-frame registration.

VI. Conclusion and Future Works

In this paper we consider the frame-to-frame registration problem, in which the point clouds are extracted from a video sequence with depth information. We compare seven techniques, named by the acronyms ICP, ICP-CTSF, SWC-ICP, GMM, Sparse ICP, S4PCS, and Sparse ICP CTSF (Section III). We use both point clouds and a RGB-D video streams in the experimental results. In the former, the ground truth rotation is provided which allows to analyse four different metrics, described on Section III, to measure the rotation error in this case. The results show better performance for Sparse ICP and Sparse ICP CTSF using the inner product of unit quaternions metric. However, when simulating missing data, the experiments show outstanding results for ICP-CTSF. Considering that missing/incomplete data is a common problem in frame-to-frame registration it was expected some influence of this fact in second class of experiments, where video sequences with depth information were segmented and the registration algorithms applied. I fact, the results show that ICP-CTSF is more reliable for frame-to-frame registration.

As further works, we should observe that the CTSF can be used as a dissimilarity factor between any second order tensors and applied in tasks other than rigid registration. Therefore, a new avenue is to apply this criterion in non-rigid alignments problems and compare its performance with counterpart ones [9], [48], [49] in a more general registration scenario.

Acknowledgment

We thank all the reviewers for the very constructive comments, which helped us a lot to improve this work. In addition, the authors would like to thank the support provided by PCI-LNCC.

References

[1] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, Wiley Online Library, 2016.

[2] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, Feb. 1992.

[3] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image and Vision Computing*, vol. 25, no. 5, pp. 578–596, 2007.

[4] G. Tam, Z.-Q. Cheng, Y.-K. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X.-F. Sun, and P. Rosin, "Registration of 3d point clouds and meshes: A survey from rigid to nonrigid," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 7, pp. 1199–1217, 2013.

[5] Y. Díez, F. Roure, X. Lladó, and J. Salvi, "A qualitative review on 3d coarse registration methods," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 45:1–45:36, 2015.

[6] L. W. X. Cejnog, F. A. A. Yamada, and M. B. Vieira, "Wide angle rigid registration using a comparative tensor shape factor," *International Journal of Image and Graphics*, vol. 17, no. 01, p. 1750006, 2017.

[7] F. A. de Araujo Yamada, "A shape-based strategy applied to the covariance estimation on the ICP," Master's thesis, Post-Graduation Program on Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil, 2016.

[8] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Computer Graphics Forum*, vol. 32, no. 5, pp. 113–123, 2013.

[9] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1633–1645, 2011.

[10] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*, vol. 33, pp. 205–215, Wiley Online Library, 2014.

[11] Cyberware, *Cyberware 3030 MS scanner*. 1999.

[12] M. Levoy, J. Gerth, B. Curless, and K. Pull, "Bunny Model," tech. rep., https://graphics.stanford.edu/data/3Dscanrep/, 1994.

[13] F. A. d. A. Yamada, M. B. Vieira, G. A. Giraldi, and A. L. Apolinário, "Comparing different strategies for frame-to-frame rigid registration of point clouds," in *19th Symp. on Virtual and Augmented Reality (SVR)*, pp. 87–96, Nov 2017.

[14] C. V. Group, "RGB-D SLAM Dataset and Benchmark," tech. rep., https://vision.in.tum.de/data/datasets/rgbd-dataset/download, 2012.

[15] B. Sabata and J. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 309–324, 1991.

[16] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[17] 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*, no. 5, pp. 698–700, 1987.

[18] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[19] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 358 – 367, 1991.

[20] B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, vol. 5, no. 7, pp. 1127–1135, 1988.

[21] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.

[22] G. Dalley and P. Flynn, "Pair-wise range image registration: a study in outlier classification," *Computer Vision and Image Understanding*, vol. 87, no. 1, pp. 104–115, 2002.

[23] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[24] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis, "A generative model for the joint registration of multiple point sets," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), pp. 109–122, Springer International Publishing, 2014.

[25] M. Danelljan, G. Meneghetti, F. S. Khan, and M. Felsberg, "A probabilistic framework for color-based point set registration," in *2016 IEEE Conf. on Comp. Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 1818–1826, 2016.

[26] M. Danelljan, G. Meneghetti, F. Khan, and M. Felsberg, "Aligning the Dissimilar: A Probabilistic Method for Feature-Based Point Set Registration," in *2016 23RD INT. CONF. ON PATTERN RECOGNITION (ICPR)*, International Conference on Pattern Recognition, pp. 247–252, 2016.

[27] F. I. Ireta Muñoz and A. I. Comport, "Point-to-hyperplane RGB-D Pose Estimation: Fusing Photometric and Geometric Measurements," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2016)*, Oct 2016.

[28] X. Huang, J. Zhang, L. Fan, Q. Wu, and C. Yuan, "A systematic approach for cross-source point cloud registration by preserving macro and micro structures," *IEEE Trans. Image Processing*, vol. 26, no. 7, pp. 3261–3276, 2017.

[29] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, pp. 545–548, IEEE, 2002.

[30] C. Li, J. Xue, N. Zheng, S. Du, J. Zhu, and Z. Tian, "Fast and robust isotropic scaling iterative closest point algorithm," in *18th IEEE International Conference on Image Processing, ICIP 2011, Brussels, Belgium, September 11-14, 2011*, pp. 1485–1488, 2011.

[31] L. W. X. Cejnog, "Rigid registration based on local geometric dissimilarity," Master's thesis, Post-Graduation Program on Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil, 2015.

[32] M. B. Vieira, P. Martins, A. Araujo, M. Cord, and S. Philipp-Foliguet, "Smooth surface reconstruction using tensor fields as structuring elements," in *Computer Graphics Forum*, vol. 23, pp. 813–823, Wiley Online Library, 2004.

[33] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," in *ACM Transactions on Graphics (TOG)*, vol. 27, pp. 85:1–85:10, ACM, 2008.

[34] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[35] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, "Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1229–1234, 1999.

[36] L. Greengard and J. Strain, “The fast gauss transform,” *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 1, pp. 79–94, 1991.

[37] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, “Improved fast gauss transform and efficient kernel density estimation,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 664–671, IEEE, 2003.

[38] B. Ravani and B. Roth, “Motion synthesis using kinematic mappings,” *Journal of mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 3, pp. 460–467, 1983.

[39] P. Wunsch, S. Winkler, and G. Hirzinger, “Real-time pose estimation of 3d objects from camera images using neural networks,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 3232–3237, IEEE, 1997.

[40] J. J. Kuffner, “Effective sampling and distance metrics for 3d rigid body path planning,” in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 4, pp. 3993–3998, IEEE, 2004.

[41] P. M. Larochelle, A. P. Murray, and J. Angeles, “A distance metric for finite sets of rigid-body displacements via the polar decomposition,” *Journal of Mechanical Design*, vol. 129, no. 8, pp. 883–886, 2007.

[42] H. Goldstein, *Classical Mechanics*. Addison-Wesley, 2nd ed., 1981.

[43] P. C. camera, *Cyberware 3030 MS scanner*. 2016.

[44] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, “A large dataset of object scans,” *arXiv:1602.02481*, 2016.

[45] B. Jian and B. C. Vemuri, *GMM Implementations*. 2011.

[46] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgbd slam systems,” in *In Int. Conf. on Intelligent Robot Systems (IROS*, 2012.

[47] D. Set, “File formats,” tech. rep., https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats, 2012.

[48] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas, “Non-rigid registration under isometric deformations,” *Comput. Graph. Forum*, vol. 27, pp. 1449–1457, 2008.

[49] M. Grogan and R. Dahyot, “Shape registration with directional data,” *Pattern Recognition*, vol. 79, pp. 452–466, 2018.