Comparing Different Strategies for Frame-to-Frame Rigid Registration of Point Clouds

Fernando A. de A. Yamada*†, Marcelo B. Vieira*, Gilson A. Giraldi†, Antonio L. Apolinário Jr.‡
*Federal University of Juiz de Fora, Computer Science Department, Juiz de Fora, MG, ZIP 36036-900
Email: akio@ice.ufjf.br and marcelo.bernardes@ufjf.edu.br
†National Laboratory for Scientific Computing Petropolis, RJ, ZIP 25651-075
Email: gilson@lncc.br
‡Federal University of Bahia, Computer Science Department, Salvador, BA, ZIP 40170-110
Email: antonio.apolinario@ufba.br

***Abstract*—Pairwise surface rigid registration aims to find the rigid transformation that best register 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 by using both point clouds and a RGB-D video stream in the experimental results. The former, is considered under different viewpoints, with the addition of outliers and noise. Once the ground truth rotation is provided, we discuss four different metrics to measure the rotation error in this case. The video sequence with depth information is segmented to get the target object. Next, the registration algorithms are applied and the average root mean squared error is computed. Since the ground truth is not available in this case, we develop a superposition strategy to visually check performance of the algorithms. Besides, we analyse the robustness of the techniques against spatial and temporal sampling rates.**

Keywords: Rigid Registration, Point Clouds, ICP, Frame-to-Frame

1. Introduction

Surface registration is a common computer vision problem, with applications in computer graphics, virtual/augmented reality, robotics, quality inspection, photogrammetry, pose estimation, among others, as we can see in related surveys [1], [2], [3], [4]. Rigid registration is a sub-problem, dealing only with sets that differ by a rigid motion, composed by rotation R and a translation $\mathbf{t}$. The sets may come from the same or from different kinds of sensors [5].

The classical and most cited algorithm in the literature to rigid registration is the Iterative Closest Point (ICP) [6]. This algorithm takes as input two point clouds P and Q, named source and target sets, respectively, 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.

The ICP approach, although breakthrough at its time, presented several possible optimizations and improvements. It assumes that there is a correct correspondence between the points of both clouds which is an assumption that easily fails on real applications [4]. Another issue of ICP and some variants is that they expect that the point clouds are already coarsely aligned [2], [7] .

In this paper, our goal is to compare the convergence characteristics of surface registration methods in the frame-to-frame problem, where the point clouds are obtained from a video stream of range images. In this case, we observe the following problems: partial overlapping between point clouds, noise, outliers, scale variation, and missing data. Based on the corresponding requirements (see [8] for details), we choose the classical ICP [6], a combination between the ICP and shape descriptors based on CTSF (ICP-CTSF) [7], Shape-based Weighting Covariance ICP (SWC-ICP) [8], Gaussian mixture model (GMM) [9], Sparse ICP (uses L_p norms) [10] as well as its combination with CTSF (Sparse ICP CTSF) [7] and Super 4PCS that is robust to partial overlapping [11].

To evaluate each algorithm in the target application, we firstly consider point clouds acquired through a Cyberware 3030 MS scanner available in the Stanford $3D$ scanning repository [12]. In this case, the ground truth rotation is available and, as a consequence, we could evaluate four different metrics, presented in [13], to measure the rotation error. Results show better performance for Sparse ICP and Sparse ICP CTSF in these experiments in the inner product of unit quaternions metric.

Next, we evaluate the registration techniques for frame-to-frame registration using a video sequence with depth information. We follow the literature [2], [14], and use the average root mean squared error ($MRMS$) as well as a visual inspection procedure to analyse the results. These experiments show that the $MRMS$ of Sparse ICP and the Sparse ICP CTSF are the highest ones, contrasting with the results of the previous experiments. However, the visual inspection does not agree with this conclusion demanding

further analysis.

Closely related to our work is the study presented by Dalley and Flynn [14] that analyses iterative closest point (ICP) variants in fine registration tasks using partially overlapping range image pairs. However, differently from [14], our study involves an outdoor video, acquired through an RGB-D sensor, and different rotation error metrics when the ground truth rotation is available.

The remainder of this paper is organized as follows. In section 2 we summarize the considered methods. The section 3 shows the experimental results obtained by applying the registration methods to point clouds and as well as a depth video sequence. Section 4 presents the conclusions and future researches. An extended version of this material is available in [8].

2. Registration Algorithms

Along the text, given a set S , the symbol $|S|$ means the number of elements of S. Besides, I_m represents the $m \times m$ identity matrix. Let the source and target point clouds in $\mathbb{R}^m$ represented, respectively, by $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{n_P}\} \subset \mathbb{R}^m$ and $Q = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{n_Q}\} \subset \mathbb{R}^m$. The registration problem aims at finding a rigid transformation $\mu : \mathbb{R}^m \rightarrow \mathbb{R}^m$ that brings set P as close as possible to set Q in terms of a designated set distance, computed using a suitable metric $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^+$, usually the Euclidean one denoted by $d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2$. A rigid transformation $\mu : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is given by $\mu(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$, with R being an element in the group of rotations in $\mathbb{R}^m$ ($SO(m)$ group) and $\mathbf{t}$ is the translation vector.

To solve the registration task, the first step is to compute the usual ICP matching relation $C(P, Q) \subset P \times Q$, that is based on the nearest neighbor computation:

$$C(P,Q) = \left\{ (\mathbf{x}_{i_l}, \mathbf{y}_{i_l}) \in P \times Q;\ \mathbf{x}_{i_l} = \arg\min_{\mathbf{x} \in P} (d(\mathbf{x}, \mathbf{y}_{i_l})) \right\}. \tag{1}$$

In the remaining text we are assuming that $|C(P,Q)| = c$. However, 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 [15]. So, we sort the pairs of the set $C(P,Q)$ such that $d(\mathbf{x}_{i_1}, \mathbf{y}_{i_1}) \leq d(\mathbf{x}_{i_2}, \mathbf{y}_{i_2}) \leq \cdots \leq d(\mathbf{x}_{i_c}, \mathbf{y}_{i_c})$ and consider a trimming parameter $0 \leq \tau \leq 1$ and a trimming boolean function: $f^{trim}(\mathbf{p}, \mathbf{q}, \tau) = 1$ if $d(\mathbf{p}, \mathbf{q}) \leq d\left(\mathbf{x}_{i_{c \cdot (1-\tau)}}, \mathbf{y}_{i_{c \cdot (1-\tau)}}\right)$ and $f^{trim}(\mathbf{p}, \mathbf{q}, \tau) = 0$, otherwise. So, we can build a new correspondence relation as:

$$C_1(P,Q,\tau) = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in C(P,Q);\ f^{trim}(\mathbf{x}_i, \mathbf{y}_i, \tau) = 1 \right\}, \tag{2}$$

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

We could also consider shape descriptors computed over each point cloud. For instance, given two points $\mathbf{p}, \mathbf{q}$ such that $\mathbf{p} \in P$ and $\mathbf{q} \in Q$, we can compare the corresponding (local) geometries using the comparative tensor shape factor (CTSF), defined as [7]:

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

where $\mathbf{S}_1 : P \rightarrow \mathbb{R}^{m \times m}$ and $\mathbf{S}_2 : Q \rightarrow \mathbb{R}^{m \times m}$ are second-order orientation tensors, $\lambda_i^{\mathbf{S}_1}(\mathbf{p})$ and $\lambda_i^{\mathbf{S}_2}(\mathbf{q})$ are the $i-th$ eigenvalues calculated in the points $\mathbf{p} \in P$ and $\mathbf{q} \in Q$, respectively, using the k nearest-neighbors of each point. In [7], k is a percentage of the total number of points being processed (see also [8]).

In this way, besides the correspondence relation $C(P,Q) \subset P \times Q$, defined in (1), we can also use the correspondence set:

$$C_{CTSF}(P,Q) = \{ (\mathbf{s}_i, \mathbf{y}_i) \in P \times Q;\ \mathbf{s}_i = \arg\min_{\mathbf{p} \in P} (CTSF(\mathbf{p}, \mathbf{y}_i)) \} \tag{4}$$

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 (3). At the end of the matching processes defined by expressions (1)-(4), we get two bases of the set P, denoted by $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset P$, and $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\} \subset P$, as well as one basis of the set Q, denoted by $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \subset Q$. In order to combine both nearest neighborhood and shape information, we can consider a parameter $\omega \in \mathbb{R}$ and the mean squared error:

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

which, for $\omega = 0$, offers the usual measure of the distance between the target set Q and the transformed source point cloud $\mu(P) = \{\mu(\mathbf{p}_1), \mu(\mathbf{p}_2), \ldots, \mu(\mathbf{p}_n)\}$, with μ being the rigid transformation [6].

Considering the definitions above, the ICP-CTSF technique [7] uses the CTSF side by side with the Euclidean distance:

$$d_{c,\xi}(\mathbf{p}, \mathbf{q}, \xi) = ||\mathbf{p} - \mathbf{q}||_2 + \omega_\xi \cdot CTSF(\mathbf{p}, \mathbf{q}), \tag{6}$$

where $CTSF(\mathbf{p}, \mathbf{q})$ is given by equation (3), $\xi \in \mathbb{N}$, $\omega_\xi = \omega_0 b^\xi$, with $b < 1$. The parameter ω_0 is the initial weight given to the CTSF and b controls the update size of the weighting factor. This weighting strategy is responsible for the coarse-to-fine behavior of ICP-CTSF. Specifically, the ICP-CTSF procedure (Algorithm 1) calculates the correspondence relation:

$$C_2(P,Q,\xi) = \{ (\mathbf{x}_{i_l}, \mathbf{y}_{i_l}) \in P \times Q;\ \forall \mathbf{y}_{i_k} \in Q,\ d_{c,\xi}(\mathbf{x}_{i_l}, \mathbf{y}_{i_k}, \xi) \geq d_{c,\xi}(\mathbf{x}_{i_l}, \mathbf{y}_{i_l}, \xi) \}, \tag{7}$$

and uses it to define the set:

$$C_3(P,Q,\tau,\xi) = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in C_2(P,Q,\xi);\ f^{trim}(\mathbf{x}_i, \mathbf{y}_i, \tau) = 1 \right\}, \tag{8}$$

which is the correspondence set applied by the ICP-CTSF technique, summarized in the Algorithm 1.

The classical ICP [6] is a simplified version of the Algorithm 1, obtained by setting $\omega_0 = 0$ and using the matching relation $C_1(P_{s+1}, Q, \tau)$ instead of correspondence set $C_3(P_{s+1}, Q, \tau, \xi)$. On the other hand, the SWC-ICP methodology achieves a coarse-to-fine behavior through the use of the weighting strategy of the ICP-CTSF ($\omega_\xi \leftarrow \omega_0 b^\xi$). In this case, the set $C_3(P_{s+1}, Q, \tau, \xi)$ is replaced by the matching relation in expression (2) and the shape correspondence (4). With the obtained sets $C_1(P_{s+1}, Q, \tau)$ and $C_{CTSF}(P_{s+1}, Q)$, each iteration of SWC-ICP calculates the

Algorithm 1: 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 τ; b, such that $0 < b < 1$; $\omega_0 \gg 0$;

begin

$P_0 = P$, $s = 0$, $\xi = 1$.
$\varepsilon_0 = \infty$.
$R_0 = I_3$, $\mathbf{t}_0 = (0,0,0)^T$.
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
$C_3(P_{s+1}, Q, \tau, \xi)$ through expression (8).
Take expression (5) and minimize $e^2(R, \mathbf{t}, 0)$ to calculate the rotation R_{s+1} and translation $\mathbf{t}_{s+1}$.
Compute the error $\varepsilon_{s+1} = e^2(R_{s+1}, \mathbf{t}_{s+1}, 0)$, from (5).
if $\varepsilon_{s+1} > \varepsilon_s$ **then**
$\xi \leftarrow \xi + 1$.
$\omega_\xi \leftarrow \omega_0 b^\xi$.
end if
$s \leftarrow s + 1$.
until $\varepsilon_s > \varepsilon_{s-1}$;
return $R_s, \mathbf{t}$.

end

rigid transformation that minimizes $e^2\left(R, \mathbf{t}, \omega_0 b^\xi\right)$ given by expression (5). In the three-dimensional case ($m = 3$), if we perform the variable change $\mathbf{z}_i \leftarrow \mathbf{x}_i + \omega_\xi \mathbf{s}_i$ in expression (5), we can compute the optimum rotation and translation that best aligns the clouds $\{\mathbf{y}_i\}$ and $\{\mathbf{z}_i\}$ in the same way ICP does (see Theorem 1 in [8]). To achieve the final goal, that is, to register the clouds $\{\mathbf{y}_i\}$ and $\{\mathbf{x}_i\}$ it is just a matter to put all these operations together in an iterative scheme that varies the parameter ω from an initial value to zero (see Algorithm 3 in [8]).

The Sparse ICP [10] is formulated as recovering a rigid transformation that maximizes the number of null residuals $\mathbf{b}_i = \mathbf{R}\mathbf{x}_\mathbf{i} + \mathbf{t} - \mathbf{y}_\mathbf{i}$, 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 nearest neighbor correspondence set $C(P, Q)$ (expression (1)) and residual vector $\mathbf{b} = [||\mathbf{b}_1||_2^p, ..., ||\mathbf{b}_n||_2^p]^T$, the objective is to find a large set of inliers, $||\mathbf{b}_i||_2^p \approx 0$, and a small set of outliers, $||\mathbf{b}_i||_2^p >> 0$. This can be written as:

$$\min_{R,\mathbf{t},\mathbf{B}} \sum_{i=1}^{n} ||\mathbf{b}_i||_2^p, \text{ such that, } \delta_\mathbf{i} = \mathbf{0}, \tag{9}$$

where $\delta_\mathbf{i} = \mathbf{R}\mathbf{x}_\mathbf{i} + \mathbf{t} - \mathbf{y}_\mathbf{i} - \mathbf{b}_i$, and $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ represents a generic point in the residual space. The Algorithm that summarizes the Sparse ICP procedure is just the ICP one with the optimization problem replaced by expression (9). In the Sparse ICP CTSF [7], we keep the Sparse ICP methodology but we replace the nearest neighbor correspondence set $C(P, Q)$, given in expression (1), by the set $C_3(P, Q, \tau, \xi)$, defined by equation (8). All these algorithms are described in [8].

The Super 4PCS [11] is an improved version of the 4PCS that runs in linear time, in the number of data points. Both 4PCS and Super 4PCS follow the same idea of the RANSAC, but instead of finding triplets of points, they search for all coplanar 4-points that are approximately congruent. Then, given the target Q, the source P, and a set of coplanar points $B = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\} \subset P$, the main step of 4PCS algorithm is to extract the set U of all 4-points from Q that are approximately congruent to a set B, up to an approximation level δ.

The set U defines a set T of rigid transformations that best aligns B with some 4-points set in U. The solution of the registration problem is a rigid transformation $\mu \in T$ that brings set P as close as possible to set Q.

All the above techniques involve methods that align two point sets based on some procedure for establishing the explicit point set correspondence. Probabilistic models, like the references [9], [16], [17], discard the matching step and thus may achieve more robustness against the missing correspondences and outliers. Specifically, in the Gaussian mixture model (GMM) registration framework described in [9], each input point set is represented using a GMM model where the number of Gaussian components is the number of points.

In this context, given source (P), target (Q) point clouds, a covariance matrix Ω, and the Gaussian mixture weights vector $\mathbf{w} = (w_1, w_2, \ldots, w_{n_X})^T$, the problem of point set registration is reformulated through the minimization of a statistical discrepancy measure between the corresponding mixtures. In the GMM proposed in [9] authors apply L_2 distance for measuring similarity between two Gaussian mixtures $G(Q, \Omega, \mathbf{w})$ and $G\left(\mu(P), R\Omega R^T, \mathbf{w}\right)$, representing the target Q and the rigidly transformed source $\mu(P) = \{R\mathbf{p}_1 + \mathbf{t}, R\mathbf{p}_2 + \mathbf{t}, \ldots, R\mathbf{p}_{n_P} + \mathbf{t}\}$, respectively.

3. Experimental Results

We evaluate the performance of the methods described on section 2 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 1 shows the superposition of the clouds $0°$-$45°$ (Figure 1a) and $45°$- $90°$ (Figure 1b), where in black we picture the initial pose (source) and in red the target one. The size of the original clouds are larger than $40,000$ 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.

The web documentation in [12] offers the transformation to align each consecutive pair of views. However, we have

noticed that the precision of the translation vector is not suitable. Also, the models do not have a ground truth correspondence list. Therefore, we take only the given rotation and compare it with the ones generated by the focused techniques. For each method, we measure the computational time to calculate the alignment and the rotation error obtained using the following metrics described in [13]:

- Norm of the Difference between Quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$. Defined by $\phi_1 : SO(3) \times SO(3) \rightarrow \mathbb{R}^+$, where:

$$\phi_1(\mathbf{q}_1, \mathbf{q}_2) = \min\{||\mathbf{q}_1 - \mathbf{q}_2||_2, ||\mathbf{q}_1 + \mathbf{q}_2||_2\}, \quad (10)$$

  with $||\cdot||_2$ as the Euclidean norm and $SO(3)$ the group of 3D rotations, represented here through quaternions.
- Inner Product of Unit Quaternions. Defined by $\phi_2 : SO(3) \times SO(3) \rightarrow \mathbb{R}^+$, where:

$$\phi_2(\mathbf{q}_1, \mathbf{q}_2) = 1 - |\mathbf{q}_1 \cdot \mathbf{q}_2|, \quad (11)$$

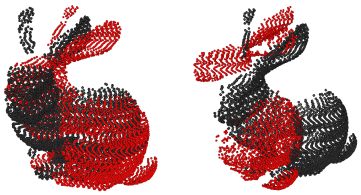
- Euclidean Difference between Euler Angles. If $A_1 = (\alpha_1, \beta_1, \gamma_1)$ and $A_2 = (\alpha_2, \beta_2, \gamma_2)$ are two sets of Euler angles, such that $\alpha, \gamma \in [-\pi, \pi)$ and $\beta \in [-\pi/2, \pi/2)$, we can define the metric $\phi_3 : E \times E \rightarrow \mathbb{R}^+$ as:

$$\phi_3(A_1, A_2) = \sqrt{d(\alpha_1, \alpha_2)^2 + d(\beta_1, \beta_2)^2 + d(\gamma_1, \gamma_2)^2}, \quad (12)$$

  where $d(a, b) = \min\{|a - b|, \quad 2\pi - |a - b|\}$.
- Deviation from the Identity Matrix. Given two rotation matrices $R_1, R_2 \in SO(3)$, the metric function $\phi_4 : SO(3) \times SO(3) \rightarrow \mathbb{R}^+$ is calculated by:

$$\phi_4(R_1, R_2) = ||I - R_1 R_2^T||_F, \quad (13)$$

  where $||\cdot||_F$ denotes the Frobenius norm in matrix space.



(a) 0°-45°. (b) 45°-90°.

Figure 1. Two alignment cases tested in the first experiment.

The second type of experiments is performed using a video sequence with RGB-D information captured using a PrimeSense Carmine camera. This video belongs to the Large Dataset of Object Scans, and the sequence used is the #03118, containing 1489 depth frames. Among others possibilities [18], this choice was based on how easy it was to segment the target and discard the background. Figure 2 illustrates the sequence. The frames have resolution of 640×480 pixels, yielding a depth image with $307,200$ points. All the experiments were carried out using an Intel Core i7-4790 CPU with 16GB RAM.

(a) RGB frame 1. (b) Depth frame 1.

Figure 2. Sample from the video sequence #03118, showing the RGB frame and its respective depth data.

3.1. Point Cloud Registration

In this section we have the following aims: (a) Analyse the different rotation error metrics (equations (10)-(13)) to decide the best one for the frame-to-frame registration problem; (b) Use the best error metric to compare the performance of the registration techniques described in section 2, in a controlled setup. In these tests, we consider the following degrees-of-freedom: (1) Registration technique; (2) Percentage k of neighbors used to calculate expression (3); (3) Error metric; (4) Trimming parameter τ.

The others 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 $b = 0.1$, $w_0 = 10^5$ [7], [8]. The Super 4PCS was set with: $\delta = 0.005$, terminate threshold 0.8, without filtering by angle, normals, distance or color [11]. Also, no further sampling of the point cloud is performed. The Sparse ICP and Sparse ICP CTSF were set with parameters: $p = 0.4$, $\mu = 15.0$, $\alpha = 1.5$, $max_\mu = 10^5$, $max_{icp} = 100$, $max_{outer} = 100$, $max_{inner} = 1$, $stop = 10^{-4}$ [7], [10]. The GMM setup follows the default values of the GMM implementation [19].

The SWC-ICP, ICP-CTSF and the Sparse ICP CTSF [7], [8] use tensors to match points through the computation of the C_{CTSF} relation given by expression (4). In these cases, we can evaluate the CTSF criterion using the isotropic voting field $\mathbf{T}$ or the anisotropic voting tensor $\mathbf{S}$, both described also in [8] . According to [8] better results have been obtained by applying only 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 [7].

To perform the task (a) we choose a pair of consecutive viewpoints of Bunny, 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 assign the minimum error to the best visual alignment.

The visual inspection of the point clouds in Figure 1 indicates that the case pictured in Figure 1a is suitable as a case-study for the task (a) because, differently from Figures 1b, 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 $k = 1\%, 5\%, 10\%, 25\%, 50\%, 75\%, 100\%$. Table 1

shows the minimum error according to each metric for $0° - 45°$. The Sparse ICP gives the smallest rotation error when considering all the metrics except ϕ_3 which achieves the minimum value for the Sparse ICP CTSF with $k = 25\%$. In special, the smallest error in Table 1 is obtained by the Sparse ICP with value almost null, given by 9.0×10^{-9}.

Figure 3 shows the 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. The Sparse ICP and Sparse ICP CTSF algorithms presented the smaller errors, which agree with the results reported in Table 1. From Figures 3a-3d it is not possible to visualize the influence of the parameter k in the rotation error, when considering the change $0° - 45°$.

The visualization of Figures 3a-3d indicates that the ICP, ICP-CTSF and SWC-ICP achieve the second place in terms of rotation errors. The Table 2 reports the minimum and maximum error for these methods, according to each considered metric (expressions (10)-(13)).

In order to check the results reported in Table 1 and Figure 3a-3d we show in Figures 4a-4b 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 4 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_2, the error of the Sparse ICP is almost null. These observations indicate that Sparse ICP performs better than the other technique and favor the choice

TABLE 1. BEST METHOD AND MINIMUM ERROR COMPUTED BY EACH METRIC FOR THE ALIGNMENT $0° - 45°$.

Metric	Best Reg. Method (section 2)	k	Error
ϕ_1	Sparse ICP	-	0.0011129861
ϕ_2	Sparse ICP	-	0.0000000090
ϕ_3	Sparse ICP CTSF	$k = 25\%$	0.0035294912
ϕ_4	Sparse ICP	-	0.0044508013

TABLE 2. PERFORMANCE OF ICP, ICP-CTSF AND SWC-ICP FOR ALIGNMENT $0° - 45°$.

		Min		Max	
	Metric	Error	k	Error	k
ICP	ϕ_1	0.0178769966	-	0.0178769966	-
	ϕ_2	0.00032	-	0.000319	-
	ϕ_3	0.00919	-	0.009194	-
	ϕ_4	0.02528	-	0.025280	-
ICP-CTSF	ϕ_1	0.017685	5%	0.017883	75%
	ϕ_2	0.000312	5%	0.000319	75%
	ϕ_3	0.009197	75%	0.009268	10%
	ϕ_4	0.025008	5%	0.025289	75%
SWC-ICP	ϕ_1	0.017669	25%	0.017925	75%
	ϕ_2	0.000312	25%	0.000321	75%
	ϕ_3	0.009155	50%	0.009262	5%
	ϕ_4	0.024987	25%	0.025348	75%

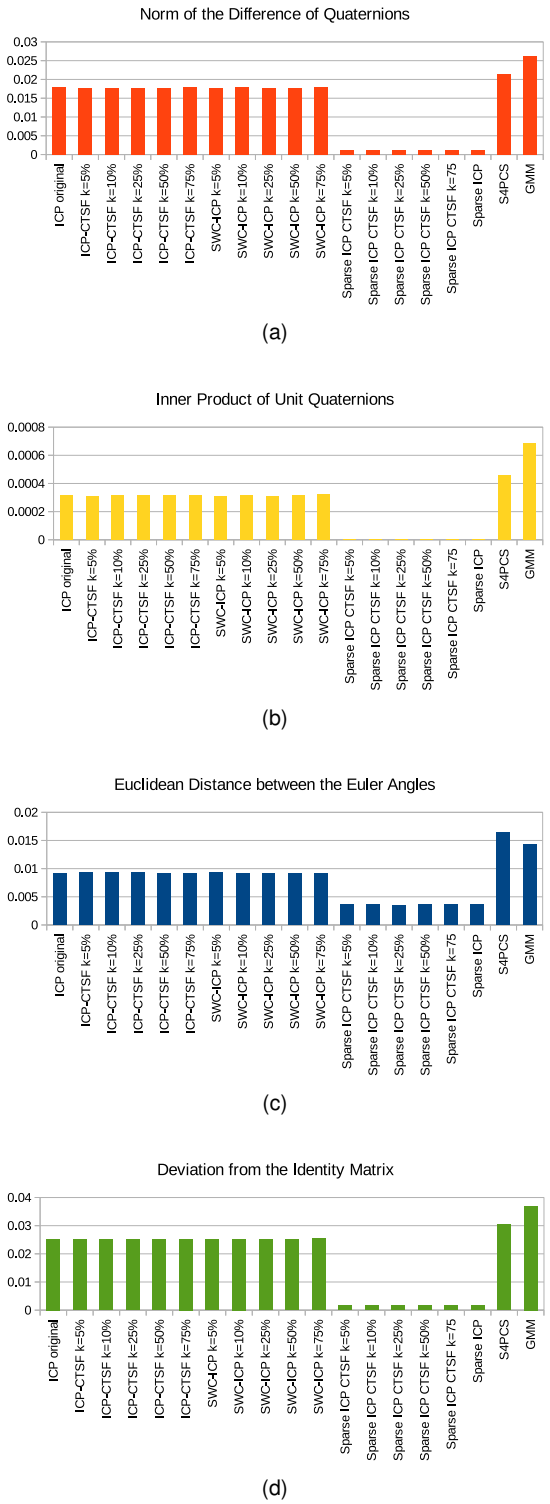
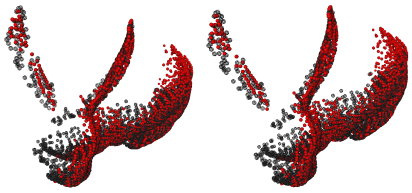


Figure 3. Rotation error for the registration techniques (Trimming parameter $\tau = 0$), for the case $0° - 45°$, computed using: (a) ϕ_1. (b) ϕ_2; (c) ϕ_3; (d) ϕ_4. The unit of ϕ_3 is radians; the other metrics are dimensionless.

of the ϕ_2 to measure the rotation error.

Figure 5 shows the errors obtained in the next exper-



(a) (b)

Figure 4. Visualization of the best cases reported in Table 1: (a) Sparse ICP. (b) Sparse ICP CTSF with $k = 25\%$.

iments, for the case $45°$-$90°$. Differently from the case $0° - 45°$, we observe that only the Sparse ICP CTSF achieves the smaller rotation errors for all the metrics which is significantly smaller than the Sparse ICP rotation error. Moreover, the effect of the parameter k in the rotation error can be perceived in the plots of Figure 5. For instance, the Table 3 reports the minimum and maximum errors achieved by the Sparse ICP CTSF regarding the considered metrics and the corresponding k values. Likewise in the above case, the smallest error happens for the metric ϕ_2, as well as the smallest error interval $[Min, Max]$, but now with $k = 5\%$ and $k = 50\%$.

TABLE 3. SPARSE ICP CTSF ROTATION ERROR FOR ALIGNMENT $45° - 90°$.

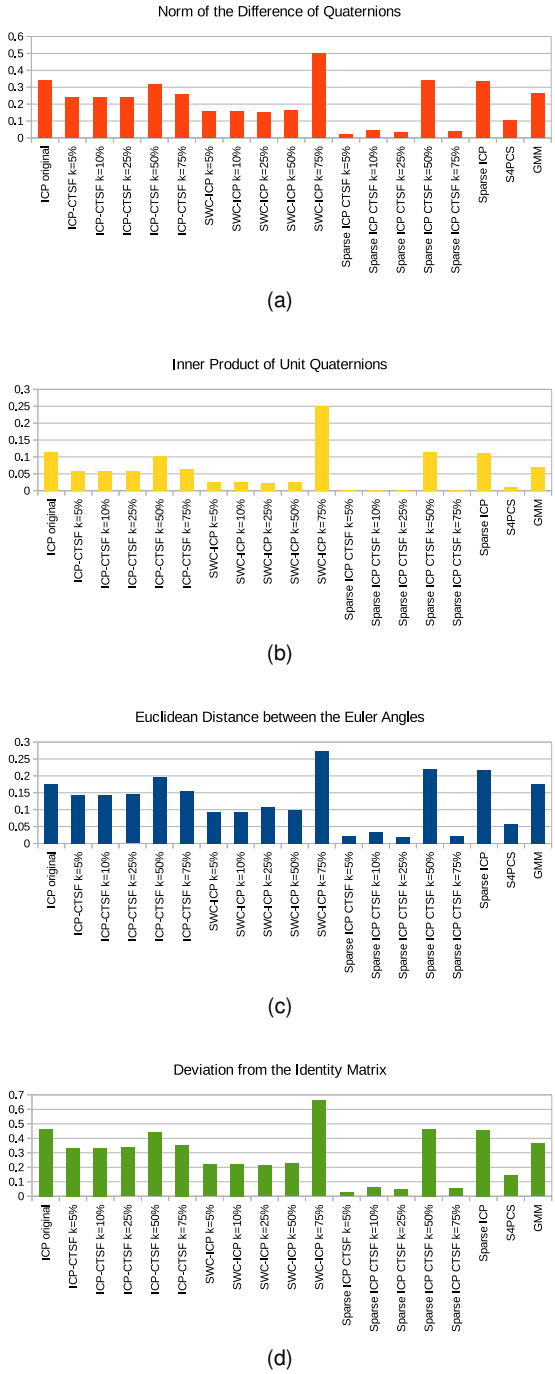
		Min		Max	
	Metric	Error	k	Error	k
Sparse ICP CTSF	ϕ_1	0.019265	5%	0.339323	50%
	ϕ_2	0.000370	5%	0.115139	50%
	ϕ_3	0.019526	25%	0.220425	50%
	ϕ_4	0.02724	5%	0.465857	50%

The Figure 6(a) allows us visually check the alignment obtained by the Sparse ICP CTSF using $k = 5\%$. Also, Figure 6(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.

We consider also a third registration test using the Bunny point clouds obtained by the views from $90°$ and $180°$. It is the hardest test, 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 registration. All methods failed to obtain a correct registration in this case as we can see in [8].

Figure 7 shows the CPU time (in seconds) for the execution of each technique in the case $0°$-$45°$. 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, although the Sparse ICP CTSF has good performance in case $0°$-$45°$, its computational time is higher. The same is true for the $45°$-$90°$ and $90°$-$180°$ alignments reported in [8].

The influence of the trimming parameter can be discussed through Figure 8, when considering the registration for $45°$-$90°$. We calculate the rotation error using function



(a) (b) (c) (d)

Figure 5. Rotation error for the registration techniques (Trimming parameter $\tau = 0$), for the case $45° - 90°$, computed using metric: (a) ϕ_1. (b) ϕ_2; (c) ϕ_3; (d) ϕ_4. The unit of ϕ_3 is radians; the other metrics are dimensionless.

ϕ_2, shown on Figure 8a. 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

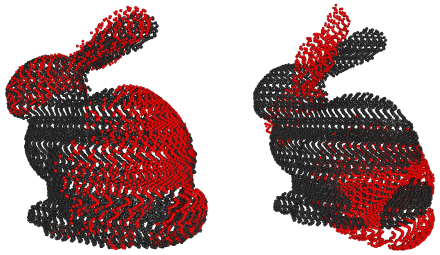


Figure 6. Case $45° - 90°$: (a) Registration obtained with Sparse ICP CTSF using $k = 5\%$. (b) Final alignment generated by Sparse ICP.

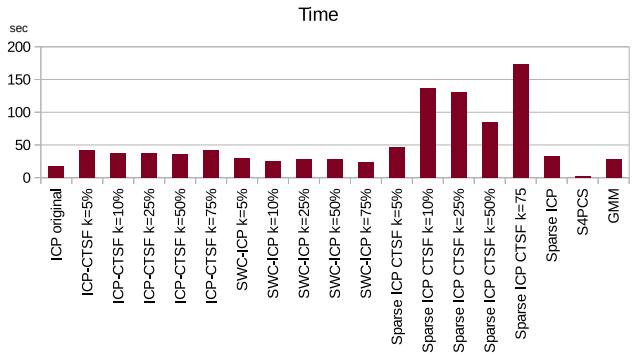


Figure 7. CPU time in seconds obtained for each method when computing alignment for $0°$-$45°$.

(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 $k \in \{25\%, 50\%, 75\%\}$ 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.

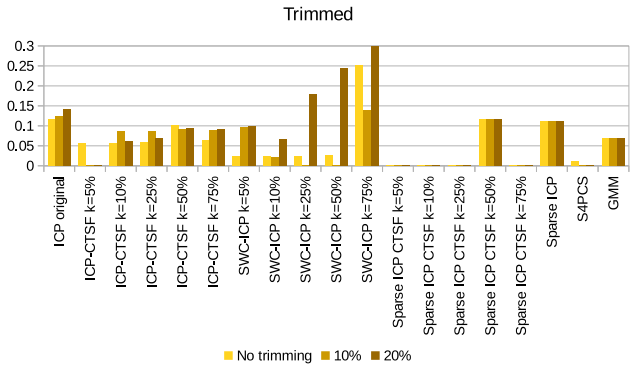


Figure 8. Influence of the trimming parameter in the case $45°$-$90°$. Rotation error according to ϕ_2 (dimensionless).

Noise is simulated adding to each point a vector $\mathbf{r} = \nu \cdot \vartheta \cdot \mathbf{u}$, where $\mathbf{u}$ is a random normalized isotropic vector, ϑ is a Gaussian random variable with null mean and variance equals to 1.0, and ν denotes a scale factor. Specifically, if $\mathbf{x}$ denotes a generic point in the set P (or Q) then its corrupted version is $\widehat{\mathbf{x}}$, given by: $\widehat{\mathbf{x}} = \mathbf{x} + \mathbf{r}$. The new point sets, denoted by P_ν and Q_ν, are composed by the noisy points so generated.

Outliers are generated using an uniform distribution over a ball, centered in the centroid of the point cloud, with diameter equals to the size of the biggest edge of the bounding box of the cloud (see [8] for details). For instance, in the case of the set P, given a percentage ζ, we uniformly sample $\zeta \cdot |P|$ points in the corresponding ball, generating the set O_P (analogous for the $\zeta \cdot |Q|$ points in O_Q, generated inside the corresponding ball). Finally, we build two new point sets named $P_\zeta = P \cup O_P$ and $Q_\zeta = Q \cup O_Q$.

We experiment with $\nu = 5\%$ to generate the noisy clouds P_ν and Q_ν and $\zeta = 20\%$ to include outliers for building P_ζ and Q_ζ. Besides, we generate two point clouds $P_{\nu,\zeta}$ and $Q_{\nu,\zeta}$ by firstly adding outliers ($\zeta = 20\%$) and, next, applying noise ($\nu = 5\%$).

We run 20 times for each configuration case (trial) and take the average rotation error to analyse the precision of each method. The Figure 9 allows to analyse the performance of the registration techniques against noise and outliers for the $0°$-$45°$ alignment. As expected, all the techniques loses precision which can be concluded when comparing the Figures 3b and 9. In the case of noise addition (Figure 9a), we can notice that the SWC-ICP achieves the worst performance. Moreover, according to Figure 9b, the experiment with outliers indicates that the ICP, SWC-ICP, Sparse ICP and S4PCS are more sensitive to this problem. The rotation error when combining noise and outliers, reported in Figure 9c, shows that ICP, SWC-ICP, and S4PCS gets worst performance.

On the other hand, the Table 4 reports the best results obtained for these experiments. We shall notice that Sparse ICP CTSF was the best technique for both noise, outliers as well as noise plus outliers. We must remember that Sparse ICP performs better than the other technique for the alignment $0°$-$45°$ without noise and outliers (Table 1). Hence, Sparse ICP performance decreases when the cloud is corrupted and it is outperformed by the Sparse ICP CTSF.

TABLE 4. BEST METHOD AND MINIMUM ERROR COMPUTED BY ϕ_2 FOR THE ALIGNMENT OF THE CLOUDS WITH NOISE AND OUTLIERS.

	Best Method	Rotation Error
Noise	Sparse ICP CTSF $k = 10\%$	0.000247
Outlier	Sparse ICP CTSF $k = 10\%$	0.000002
Noise plus Outlier	Sparse ICP CTSF $k = 5\%$	0.000117

3.2. Frame-To-Frame Registration

According to section 3.1, the best methods in the performed experiments are Sparse ICP, and Sparse ICP CTSF. In this section, we must check the obtained conclusions, but

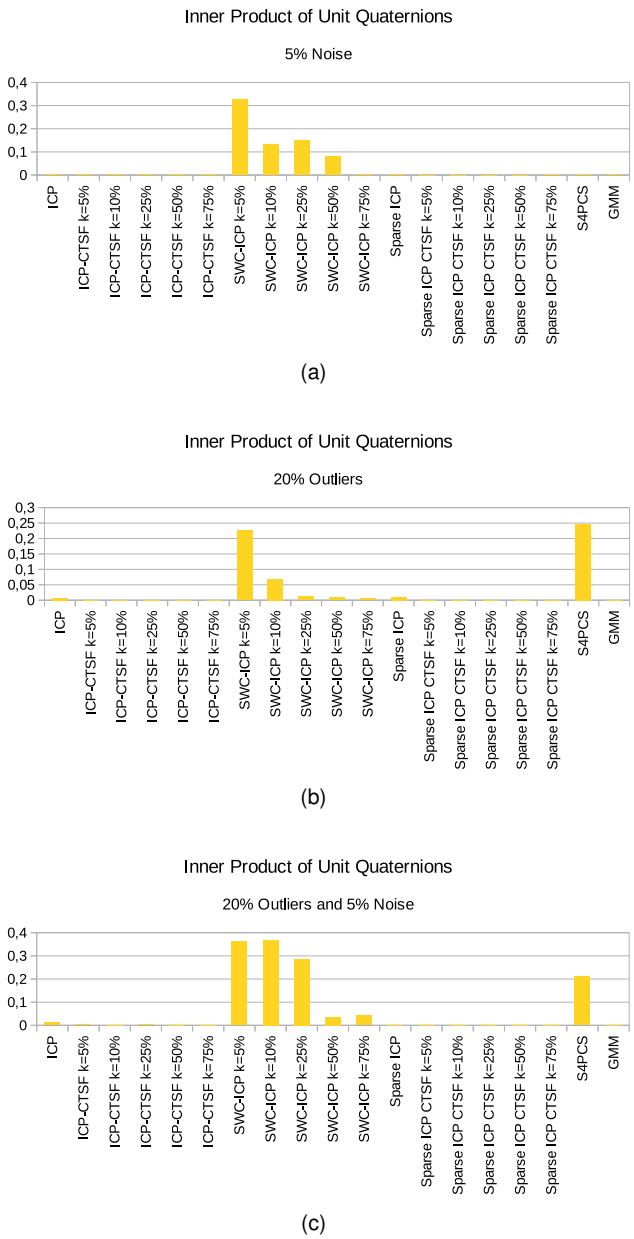


Figure 9. Rotation error for the case $0°$-$45°$ when adding: (a) Noisy points. (b) Outliers. (c) Outliers and noisy points.

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. The image resolution yields a depth array with $307,200$ elements which increase the computational cost of the registration algorithms. Therefore, we sample each frame of the video, with sampling rates $r \in \{8, 16\}$, to reduce the total number of points, generating new video sequence V.

The sequence $\#03118$ was chosen because of how easy it is to segment the target. In this video, the sign is the only meaningful object in the scene, with respect to the depth information (see Figure 2). The grass in the background is too deep to be captured and yields null depth values. Hence, we take the set $S_m = \{(i, j, C_m(i, j, 4)) ; C_m(i, j, 4) > 0\}$ and interpret it as a point cloud in $\mathbb{R}^3$. A frequent problem observed in the point clouds yielded in this processes is missing data, generated by errors in the depth field segmentation due to uncertainty caused by reflections in the camera acquisition process. Figure 10 shows a pair of consecutive clouds, in which the cloud in the frame 50 misses some points of the previous frame.

(a) Depth 49 Point cloud . (b) Depth 50 point cloud.

Figure 10. A pair of consecutive point clouds with missing data inside blue circle.

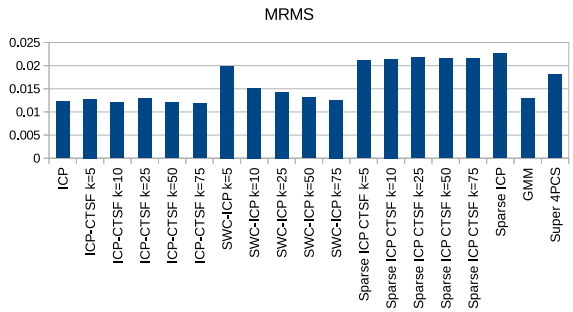
In particular, for the frames located near the end of the $\#03118$ video sequence there is another complication because of the low number of points with depth larger than zero, increasing the chance of a bad alignment. Figure 11 illustrates this case, yielding fewer points than those of Figure 2b, in comparison. Also, when the assumption that the camera follows smooth and slow motion paths is violated, which also happens near the end of the chosen sequence, scale changes between point clouds may interfere in the registration results (see Figure 13 of [8]).

(a) (b)

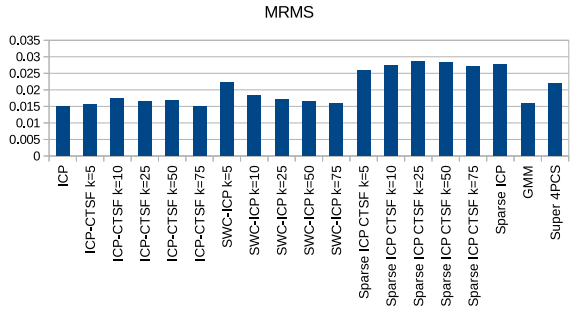
Figure 11. (a) Region with depth larger than zero in the frame 1388. (b) Respective RGB image.

We need a robust methodology to analyse the target techniques against these problems. Firstly, 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. Hence, we set $P = S_{\varsigma \cdot i}$, $Q = S_{\varsigma \cdot (i+1)}$ as the pair source/target in the frame-to-frame registration that generates the transformation pair $(R_{\varsigma \cdot i}, \mathbf{t}_{\varsigma \cdot i})$ that best aligns the source cloud $S_{\varsigma \cdot i}$ with the target one $S_{\varsigma \cdot (i+1)}$. Secondly, following the literature [2], [14], we consider the average root mean squared error, denoted by $MRMS$, to quantify the precision of the registration of the cloud pairs $S_{\varsigma \cdot i}$ and $S_{\varsigma \cdot (i+1)}$, in the whole video stream:

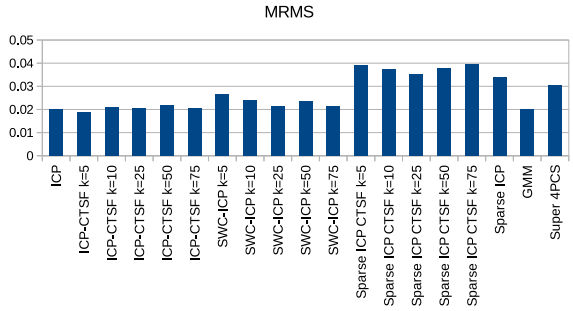
$$MRMS(\varsigma, V) = \left(\frac{1}{\frac{|V|}{\varsigma}} \right) \sum_{i=0}^{\frac{|V|}{\varsigma}-1} \sqrt{e^2 (R_{\varsigma \cdot i}, \mathbf{t}_{\varsigma \cdot i}, 0)}, \quad (14)$$



(a) $\varsigma = 1$



(b) $\varsigma = 2$



(c) $\varsigma = 4$

Figure 12. Variation of the temporal sampling. All the three cases have video sampling rates $r = 8$.

where $e^2\left(R_{\varsigma \cdot i}, \mathbf{t}_{\varsigma \cdot i}, 0\right)$ is calculated through expression (5). Moreover, since the choice of parameter k of the ICP-CTSF, SWC-ICP and Sparse ICP CTSF impacts on the results, we show how they change with $k = 75\%$, $k = 50\%$, $k = 25\%$, $k = 10\%$ and $k = 5\%$, like in section 3.1. All methods were set with the same parameters of the experiments in section 3.1.

Figure 12 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 ones, contrasting with the results of the previous experiments. The variation of the parameter k also do not have much effect, except for a small trend on the SWC-ICP, where smaller values of k yields higher $MRMS$ errors.

Figure 13 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 $i + 1$. When comparing Figures 12a and 13 we notice that all methods almost doubled the 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.

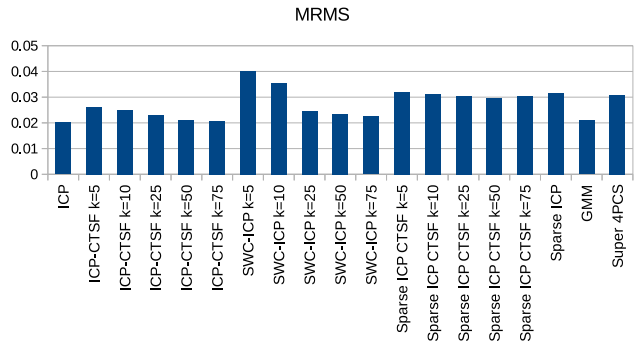


Figure 13. MRMS of all methods (in pixel unities) using video sample $r = 16$, with $\varsigma = 1$.

Since the $MRMS$ values presented some inconsistencies with the previous experiments regarding the Sparse ICP and Sparse ICP CTSF, we used a visual inspection to check if the results showed using the $MRMS$ correctly indicate the best methods. In this process, we simulate an attempt to reconstruct the objects using frames 1 to 4 from the sequence #03118. The resulting registration of the pairs (1,2), (1,3) and (1,4) were overlapped and visualized in red. It is expected that the density of red points would be higher in worst registrations. Figure 14 shows the obtained result. The Sparse ICP CTSF with $k \in \{5\%, 10\%\}$ produces an image lighter than other methods, like the darkest red image produced from the ICP-CTSF with $k = 50\%$. In Figure 15 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 15a and 15b), differently from the latter (Figures 15c and 15d).

Hence, although the ICP-CTSF yields smaller $MRMS$ errors than the Sparse ICP CTSF, the visual inspection shows the opposite result. Further works should be undertaken to find out the cause of the disparity between $MRMS$ values and visual inspection. Without a proper way to measure the distance of the ground truth correspondences, this kind of experiment needs a visual inspection to define which method is the most suitable one.

4. 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, Sparse ICP CTSF and Super 4PCS

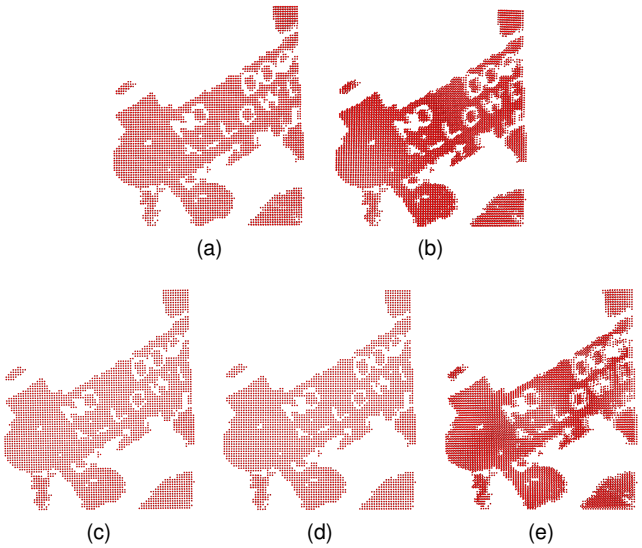


Figure 14. Overlapping of the registered frames 1-4 by different algorithms. (a) Original ICP. (b) ICP-CTSF $k = 50\%$. (c) Sparse ICP CTSF $k = 5\%$. (d) Sparse ICP CTSF $k = 10\%$. (e) Sparse ICP CTSF $k = 75\%$.

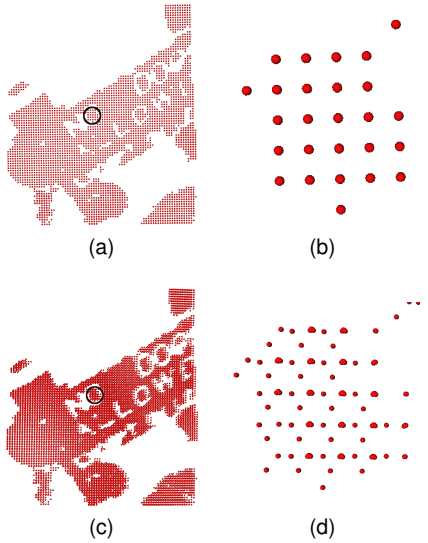


Figure 15. (a) Sparse ICP CTSF $k = 5\%$. (b) Selected region in detail. (c) ICP-CTSF $k = 50\%$. (d) Zoom-in the selected region.

(section 2). We use both point clouds and a RGB-D video stream in the experimental results. In the former, the ground truth rotation is provided which allows to analyse four different metrics, 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. In the second class of experiments, a video sequence with depth information is segmented and the registration algorithms are applied. The results show an inconsistency between the $MRMS$ and the visual inspection of the results demanding further analysis to understand the limitations of $MRMS$ as a metric error to tackle the analysis of frame-to-frame registration tasks. Moreover, scalability issues related to the focused methods must be considered in future works.

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] 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.

[3] 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.

[4] 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, p. 45, 2015.

[5] 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.

[6] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[7] 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, 2017.

[8] F. A. de A. Yamada, G. A. Giraldi, M. B. Vieira, and A. L. A. Jr., "Frame-to-frame rigid registration of point clouds extracted from depth sequences: Comparing different strategies," National Laboratory for Scientific Computing, Tech. Rep. 1/2017.

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

[11] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 205–215.

[12] M. Levoy, J. Gerth, B. Curless, and K. Pull, *Bunny Model*, Stanford University, 2014.

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

[14] 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.

[15] 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. IEEE, 2002, pp. 545–548.

[16] G. Evangelidis, D. Kounades-Bastian, R. Horaud, and P. E.Z., "A generative model for the joint registration of multiple point sets," in *ECCV*, 2014.

[17] M. Danelljan, G. Meneghetti, F. S. Khan, and M. Felsberg, "Aligning the dissimilar: A probabilistic method for feature-based point set registration," in *ICPR*, 2016, pp. 247–252.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IROS*. IEEE, 2012, pp. 573–580.

[19] B. Jian and B. C. Vemuri, *GMM Implementations*, 2011. Available at GitHub