

# Using 2 Point+Normal Sets for Fast Registration of Point Clouds with Small Overlap

Carolina Raposo and João P. Barreto

**Abstract**—Global 3D point cloud registration has been solved by finding putative matches between the point clouds for establishing alignment hypotheses. A naive approach would try to perform exhaustive search of triplets with a cubic runtime complexity in the number of data points. Super4PCS reduces this complexity to linear by making use of sets of 4 coplanar points. This paper proposes 2-Point-Normal Sets (2PNS), a new global 3D registration approach that advances Super4PCS by using 2 points and their normals for generating alignment hypotheses. The dramatic improvement in the complexity of 2PNS when compared to Super4PCS is demonstrated by the experiments that show speed-ups of two orders of magnitude in noise-free datasets and up to  $5.2\times$  in Kinect scans, while improving robustness and alignment accuracy, even in datasets with overlaps as low as 5%.

## I. INTRODUCTION

3D Registration is the process of finding the rigid transformation that best aligns two point clouds (PCs) such that their overlapping areas match as well as possible. It is a well studied problem in computer vision due to its applications in object detection and recognition, tracking, and SLAM.

The oldest and best established algorithm for solving the PC registration problem is ICP [1]. Since it relies in iterative optimization, if the initialization is poor, it might converge to local minima. Many authors have worked in improving the resilience of ICP to outliers and missing data, often observed in 3D scans. Examples include the work [2] where local quadratic approximations of the squared distance function are used and the SparseICP method [3] that formulates the registration problem using sparsity-induced norms. Similarly to the standard ICP, all of these methods require good initialization of the rigid transformation.

Other works, such as Go-ICP [4], have tried to avoid the initialization issue by assuring global convergence. It is based on a branch-and-bound scheme that searches the entire 3D motion space. Unfortunately, this requires large overlap between the point clouds, which does not always happen due to occlusions or acquisitions in very different viewpoints, and scales poorly, becoming prohibitive for large datasets.

In order to avoid the requirements of good initialization and large overlap, some authors have suggested to find sparse correspondences between PCs and use them in an hypothesize-and-test framework, like RANSAC, to perform

an initial alignment that is subsequently refined by ICP. Such approaches handle completely misaligned PCs, do not require them to have large overlap, and are inherently robust to outliers. The difficulty is in establishing plausible correspondences between PCs. It is well known that a rigid transformation that aligns two PCs can be computed in a closed form manner from 3 correct 3D point correspondences [5]. Thus, a naive approach would select 3 random points in one point cloud and search for all possible triplets in the other. This search would have complexity of  $\mathcal{O}(N^3)$ , with  $N$  being the number of points in the second model, easily becoming prohibitive. This chapter proposes a method to address this issue, named 2-Point-Normal Sets (2PNS).

### A. Related work

An approach inspired by what is commonly done in 2D images is to find salient regions/landmarks in the PCs, characterize them by a descriptor for performing association and establish alignment hypotheses for each match in a RANSAC-like framework. Such approaches fail mainly because PCs are often smooth and/or noisy, which hampers finding repeatable saliences that can be matched.

Aiger *et al.* [6] proposed 4-Point Congruent Sets (4PCS) that, given 4 coplanar points in one PC, enables to exhaustively search for correspondences in the other PC with a complexity of  $\mathcal{O}(N^2)$ . The key idea is to use the relations between 4 coplanar points to define affine invariants that are preserved by a rigid displacement. More recently, Mellado *et al.* proposed the Super4PCS algorithm [7] that decreases complexity from quadratic to linear time by using a number of improvements in the search stage. Super4PCS has inspired the Super Generalized 4PCS [8] that removes the coplanarity constraint and considers general 4-point bases, leading to speed-ups up to  $6.5\times$  when compared to Super4PCS.

In this paper, we advance the Super4PCS framework by using not 4 points to establish an alignment hypothesis but 2 points and their normals. The normals to points in a point cloud are relatively inexpensive to compute [9] and have already been used in the context of registration mainly to define local descriptors. We show that it is possible to compute the rigid transformation  $R, t$  from 2 points and 1 normal and that matching 2 points+normals can be significantly more efficient and robust than searching for sets of 4 points forming a congruent base. The results show that we can obtain speed-ups up to two orders of magnitude in noise-free datasets and up to  $5.2\times$  in Kinect scans with respect to Super4PCS, while improving robustness and alignment accuracy.

Carolina Raposo acknowledges the Portuguese Science Foundation (FCT) for funding her PhD under grant SFRH/BD/88446/2012. The authors also thank FCT and COMPETE2020 program for generous funding through project VisArthro with reference PTDC/EEL-AUT/3024/2014.

The authors are with the Institute of Systems and Robotics, Dept. of Electrical and Computer Engineering, University of Coimbra, Portugal. {carolinaraposo, jpbar}@isr.uc.pt

## II. REVIEW OF SUPER4PCS [7]

Global 3D PC registration works by finding putative point correspondences between PCs that enable to establish alignment hypotheses. These hypotheses are ranked according to some metric and the transformation  $T$  with the highest score is then refined by ICP. Since a rigid transformation can be estimated from a minimum of 3 points [5], a naive approach would try to perform exhaustive search of triplets with a runtime complexity of  $\mathcal{O}(N^3)$ ,  $N$  being the size of the PC. The algorithms 4PCS [6] and later Super4PCS [7] work with sets of 4 points, instead of 3, to make the search easier.

Let  $\mathcal{P}$  and  $\mathcal{Q}$  be the source and target PCs, respectively, to be registered. The goal of 4PCS and Super4PCS is to find the transformation  $T$  that best aligns them by solving the Largest Common Pointset (LCP) problem: maximize the cardinality of a transformed subset of  $\mathcal{P}$  according to the property that every point in that subset is within a predefined distance to  $\mathcal{Q}$ . These methods make no assumption about the starting poses of the PCs and are able to handle situations of small overlap. Also, they favour the use of points in the cloud that are far apart to increase resilience to noise and outliers.

### A. The original 4PCS algorithm

The 4PCS algorithm solves the global 3D registration problem by using coplanar sets of 4 points, rather than the minimum sets of 3 points, allowing the employment of a technique that efficiently matches pairs of affine invariant ratios in 3D. The approach works by selecting a base of 4 coplanar points in the source PC  $\mathcal{P}$  and finding all the 4-point sets in the target PC  $\mathcal{Q}$  that are approximately congruent with the base, *i.e.* related by rigid transformations. For each potential 4-point set from  $\mathcal{Q}$ , the aligning transformation  $T$  is computed and the best transformation according to the LCP score is retained. This process is repeated in a RANSAC [10] scheme until a good solution is found, or a maximum number of iterations is reached.

The first step of each RANSAC iteration is the selection of a random base of 4 coplanar points from  $\mathcal{P}$ . The algorithm starts by randomly selecting 3 points from  $\mathcal{P}$  that yield a wide triangle. The fourth point in the quadrilateral is selected as one that is close to be planar to the other 3 but still not too close to them. This is done by testing all the  $S$  points in  $\mathcal{P}$  and picking the one that best fits the criteria, with runtime complexity of  $\mathcal{O}(S)$ .

Let  $\mathcal{B} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$  in Figure 1 be the randomly selected coplanar base in source  $\mathcal{P}$ , such that  $\mathbf{AB}$  intersects  $\mathbf{CD}$  in the intermediate point  $\mathbf{E}$ . The key idea explored by the authors is that the ratios

$$r_1 = \|\mathbf{A} - \mathbf{E}\| / \|\mathbf{A} - \mathbf{B}\| \text{ and } r_2 = \|\mathbf{C} - \mathbf{E}\| / \|\mathbf{C} - \mathbf{D}\| \quad (1)$$

remain invariant under affine transformations, and hence under rigid motion. Since distances are also preserved under rigid transformations, these 4 invariants ( $r_1, r_2, d_1 = \|\mathbf{A} - \mathbf{B}\|, d_2 = \|\mathbf{C} - \mathbf{D}\|$ ) are used to constrain the search for congruent 4-point sets in target  $\mathcal{Q}$ . The algorithm starts by extracting all pairs of points at distance  $d_1$  or  $d_2$  from  $\mathcal{Q}$ , which is done in  $\mathcal{O}(N^2)$  time, where  $N$  is the cardinality

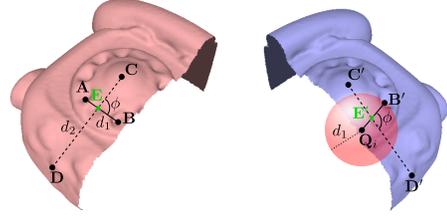


Fig. 1: A source  $\mathcal{P}$  (pink) and a target  $\mathcal{Q}$  (blue) PCs are shown with the corresponding 4-point sets used by 4PCS and Super4PCS. Super4PCS finds all pairs at distance  $d_1$  in the target PC by centring a sphere of radius  $d_1$  in all points and selecting the points that intersect with the PC.

of  $\mathcal{Q}$ . For each extracted pair  $(\mathbf{Q}_1, \mathbf{Q}_2) \in \mathcal{Q}$  with distance  $d_1$  or  $d_2$ , the intermediate points  $\mathbf{E}_1$  and  $\mathbf{E}_2$  are computed. The authors explore the fact that two pairs that have their  $\mathbf{E}_1$  and  $\mathbf{E}_2$  coincident form a 4-point base that is related with  $\mathcal{B}$  by an affine transformation. Thus, the intermediate points obtained from pairs at distance  $d_1$  are used for building an approximate range tree structure (RS) in  $\mathbb{R}^3$ , which takes  $\mathcal{O}(M \log M)$  to be built, where  $M$  is the number of pairs, and has a query time of  $\mathcal{O}(\log M + K)$ ,  $K$  being the number of points to be reported. The intermediate points that arise from pairs at distance  $d_2$  are used as queries in this tree, yielding  $K$  4-point sets from the retrieved pairs. Since affine invariants were used, the set of extracted bases contains sets which are not related by a rigid transformation. These are then removed in a verification step that takes  $\mathcal{O}(K)$  time, by using the angle  $\phi$  between the line segments (Figure 1).

The authors claim that for large dense data sets, a heavy uniform sampling of the PCs can be performed in order to use only a small fraction of the points for computing the alignment, and the full data set is then employed for the computation of the LCP. Experiments reported in the paper show that using as few as 5% of the data points is sufficient. It is the use of the LCP measure that makes the method resilient to such heavy uniform sampling.

### B. Super4PCS

Recently, Mellado *et al.* [7] proposed the Super4PCS algorithm that builds on 4PCS and decreases its complexity to  $\mathcal{O}(N + M + K)$  by solving its two main bottlenecks: the pair extraction stage and the expensive verification step to remove non-rigid invariant 4-point candidates.

Efficient pair extraction (in  $\mathcal{O}(N)$  time) is achieved by finding the points close to the spheres centred in  $\mathbf{Q}_i \in \mathcal{Q}$  and with radius  $d_1 \pm \epsilon$  and  $d_2 \pm \epsilon$ , where  $\epsilon$  is a tolerance considered due to the noise in the data. Figure 1 illustrates this procedure, depicting a sphere of radius  $d_1$  centred in an arbitrary point  $\mathbf{Q}_i$ . The PC  $\mathcal{Q}$  is organized in a 3D grid, subdivided recursively and the intersection between the set of spheres and the subdivided volumes is computed. Pairs are then built between the points lying in the intersected volumes and the sphere centres.

Concerning the second bottleneck, the authors' idea is to extract the exact set of congruent 4-point bases that only

contains rigid-invariant candidates, avoiding the need to have a verification/filtering step. A quadrilateral is congruent to the base selected from  $\mathcal{P}$  if it is composed of pairs with the correct length ( $d_1$  and  $d_2$ ), and if the angle  $\phi$  between these pairs is similar to the angle formed by the two pairs in the base. This is accomplished as follows: each pair is represented by its intermediate point  $\mathbf{E}$  and its orientation, and the pairs at distance  $d_1$  are hashed by this position and orientation, with the directions being mapped to a spherical map. In the query stage, the position  $\mathbf{E}$  is used to access cells in a regular grid, such that the retrieved points lie in the same cell as the query. Also, the corresponding spherical map is queried using a  $d_2$  pair direction to find all pairs with angle  $\phi$  w.r.t. the query direction. This is done by intersecting a cone of aperture  $2\phi$  around the query direction with the spherical map. This stage has runtime complexity  $\mathcal{O}(M + K)$  due to the point retrieval and query of the spherical map steps.

Figure 3a shows the sequence of steps of a complete RANSAC iteration of Super4PCS. Since normals are often available for PCs, or can be easily computed, they can be used to further prune the number of extracted pairs. Whenever this information is available, Super4PCS includes it by computing the angles  $\theta_1$  and  $\theta_2$  between the normals of pairs of points with distance  $d_1$  and  $d_2$ , respectively, and extracting pairs that have similar distances and angles as the base set extracted from  $\mathcal{P}$ . We will consider this version of Super4PCS with normals throughout this paper.

### III. 2-POINT-NORMAL SETS (2PNS)

In this section we present 2PNS, our global 3D registration method that builds on Super4PCS. Motivated by the fact that normals to PCs can be computed in a robust, inexpensive manner, we propose to use them to solve the alignment problem. It can be shown that the rigid transformation  $\mathbf{T}$  can be computed from 2 points plus the normal at one of these points. Thus, instead of searching for quadrilaterals, we look just for pairs of points using their normals to decide if they are a plausible match. This dramatically reduces the combinatorics of the search leading to a simpler, substantially less complex algorithm, as shown in Figure 3b that presents the scheme of a complete RANSAC iteration of our method. The modified and removed steps w.r.t. Super4PCS are highlighted in green and red, respectively in Figure 3a.

#### A. Computing normals

PC normal estimation has been a well-studied problem due to its important applications in areas such as object detection, segmentation, surface fitting and registration. According to a recent work that compares several approaches for surface normal estimation in PCs, the most accurate method is PlanePCA [11], an optimal total least squares solution, being the one we use in the experiments.

Since the direction of the normal is not recovered - its symmetric is also a solution - there are always two solutions for the unitary normal vector. This is important because normals are used in our proposed method in the estimation of the rotation  $\mathbf{R}$  and symmetric vectors lead to

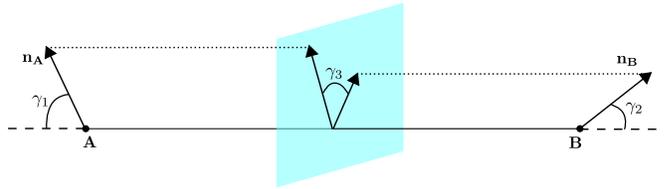


Fig. 2: Angles used in the 2PNS pair extraction stage.

different solutions, as shown next. A scheme for forcing coherent directions of all normals in the two PCs being registered could be devised, for instance by computing each PC's centroid and making the normal vectors point in that direction. However, since we are dealing with cases of very small overlap, this could introduce errors and thus we opted to work with the two solutions for the normal vectors.

2PNS fails whenever normals are not properly estimated. This may occur if the PC is too sparse or strongly dominated by sharp edges and corners. However, both cases rarely occur, either in real or synthetic scenarios, with the PCs being often sufficiently dense and containing many smooth regions.

#### B. The 2PNS search to obtain putative matches

At each RANSAC iteration, our method starts by extracting a random pair of points and the corresponding normals from the source PC  $\mathcal{P}$ . These entities are shown in Figure 4, where  $\mathcal{P}$  is the pink PC,  $(\mathbf{A}, \mathbf{B})$  is the pair and  $\mathbf{n}_A, \mathbf{n}_B$  the two respective normals. Since we establish alignment hypotheses from 2 corresponding points and their normals, it is important that the pairs extracted from the target PC  $\mathcal{Q}$ , shown in blue in Figure 4, are congruent with the base pair extracted from  $\mathcal{P}$ , *i.e.* can be aligned by a rigid transformation.

Super4PCS extracts pairs that have a distance  $d = \|\mathbf{A} - \mathbf{B}\|$  and an angle  $\theta = \angle(\mathbf{n}_A, \mathbf{n}_B)$ . However, it is known that this leads to a set of pairs that may contain instances that can never be aligned by a rigid transformation. Thus, we propose to perform the search for pairs that verify not only the conditions for  $d$  and  $\theta$ , but also for the 3 extra angles shown in Figure 2: the angle of the first normal with the line segment joining the two points ( $\gamma_1$ ), the angle of the second normal with the line segment ( $\gamma_2$ ), and the angle between the two normals projected onto the plane orthogonal to the line segment ( $\gamma_3$ ). Since angles are preserved under rigid transformations, these 5 invariants are used for extracting congruent pairs in the target PC  $\mathcal{Q}$ , using the method from Section II-B. Due to the presence of noise and outliers in the data, a tolerance  $\delta$  is considered, which can vary between  $5^\circ$  and  $20^\circ$ , depending on the quality of the input data.

#### C. Estimation of $\mathbf{R}, \mathbf{t}$

Let  $(\mathbf{A}', \mathbf{B}')$ , with normals  $\mathbf{n}_{A'}$  and  $\mathbf{n}_{B'}$ , be a pair of points in the target PC  $\mathcal{Q}$  congruent with the selected pair in the source  $\mathcal{P}$ , as shown in Figure 4. The rigid transformation is estimated by first centring the two pairs in the origin (using their centroids) and then computing the rotation  $\mathbf{R}$ . This is

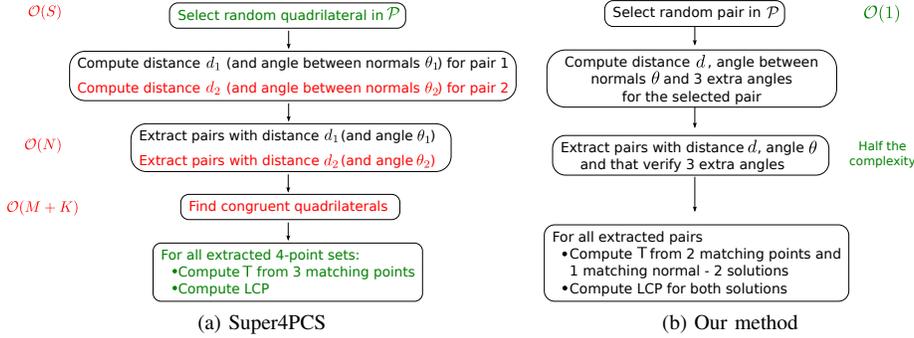


Fig. 3: Sequence of steps of a complete RANSAC iteration of (a) Super4PCS and (b) our method. The stages that were modified are identified in green and the ones that were removed are shown in red.

done in two steps: first the rotation  $R_\alpha$  that aligns the vectors  $\mathbf{v}_1 = \mathbf{B} - \mathbf{A}$  and  $\mathbf{v}_2 = \mathbf{B}' - \mathbf{A}'$  is estimated; then the rotation  $R_\beta$  that aligns the normal vectors is determined. The final rotation  $R$  comes from  $R = R_\beta R_\alpha$ .

$R_\alpha$  is a simple rotation to align two vectors, defined by the rotation axis  $\omega_\alpha = \mathbf{v}_1 \times \mathbf{v}_2$  and the rotation angle  $\alpha = \cos^{-1}(\mathbf{v}_1 \cdot \mathbf{v}_2)$ . Let  $\mathbf{n}_P^* = R_\alpha \mathbf{n}_P$ ,  $\mathbf{P} = \mathbf{A}, \mathbf{B}$  be the rotated normal vectors.  $R_\beta$  can be found by rotating an angle  $\beta$  around axis  $\mathbf{v}_2$ . In order to find  $\beta$ , we project the normal vectors  $\mathbf{n}_A^*$  and  $\mathbf{n}_{A'}$  onto the plane defined by  $\mathbf{v}_2$  and compute the angle between the projected vectors.  $R_\beta$  has only one unknown parameter, so one normal suffices to fully determine the rotation (it could be done using  $\mathbf{n}_B^*, \mathbf{n}_{B'}$ ).

The solution for  $R_\beta$  corresponding to the other solution for the normal vector (the symmetric one) is computed from the angle  $\beta' = \pi - \beta$  since the projected vector also becomes symmetric. A new solution for  $R$  is estimated, and thus this method always provides 2 solutions for the rigid transformation. After knowing  $R$ , the translation  $\mathbf{t}$  is estimated from  $R$  and the translation vectors used for centring the two pairs in the origin.

#### D. Comparison of complexity with Super4PCS

The improvements of our approach w.r.t. Super4PCS are in avoiding extra computation while selecting a random base in  $\mathcal{P}$ , reducing in half the pair search and not requiring a congruent set extraction stage. On the down side comes the fact that for each pair there are two motion solutions that must be verified, whereas Super4PCS generates only one, but this overhead is largely compensated by the improvements above, as shown in the experimental section.

1) *Selection of the base from the source PC*: The Super4PCS's procedure of extracting a coplanar quadrilateral from  $\mathcal{P}$  is described in Section II-A. It runs in  $\mathcal{O}(S)$  time because the algorithm goes through all the points in  $\mathcal{P}$  to find the fourth point in the base. On the other hand, our method simply selects a random pair of points by testing a set of pairs and choosing the widest one. This distance has an upper bound to account for the overlap between the PCs. Since we are not going through the whole PC, this stage is

very fast and runs in  $\mathcal{O}(1)$  time, being independent of the size of  $\mathcal{P}$ . Remark that since our base is a line, instead of a quadrilateral, we can choose a wider segment than the length of the quadrilateral sides and still find an overlapping area. Wider bases lead to smaller numbers of congruent sets, and thus the algorithm runs faster and more robustly [6].

2) *Congruent set extraction*: The modification to the pair extraction stage consists in performing it only once, instead of twice as in the Super4PCS algorithm, and by using 5 rigid invariants (a distance plus 4 angles), instead of 2, to constrain the search. This happens because we are working with a 2-point base, instead of a 4-point base that is decomposed into two pairs of points. It is important to notice that, while it is not required to include the normals of the points as input to the Super4PCS algorithm, when this information is available, the method uses it to further constrain the search. This is indicated with parentheses in Figure 3a.

In Super4PCS, the pair extraction stage runs in  $\mathcal{O}(N)$  time, where  $N$  is the number of points in the target PC  $\mathcal{Q}$ . Since our method performs the pair extraction process only once, it has half the computational complexity of Super4PCS, running in half the time.

Besides extracting pairs in the target  $\mathcal{Q}$ , Super4PCS has a subsequent stage for finding congruent sets. This is necessary because generating candidate quadrilaterals based solely on the distance (and possibly the angle between normals) invariant would produce sets of congruent bases with many candidates not related by rigid transformations. This congruent set extraction stage has a runtime complexity of  $\mathcal{O}(M + K)$  which is not included in our algorithm.

3) *Estimation of R*: The final modification performed to Super4PCS is the process of estimating the rigid rotation  $R$ . In the original algorithm, this is performed with Horn's method [5] using 3 out of the 4 points of the quadrilateral and provides a unique solution. In our case there are two solutions as explained in Section III-C. For the same number  $K$  of congruent bases, this would lead to twice the computational complexity. However, due to the significant decrease in complexity in the other stages of the pipeline, the final runtime is still significantly lower. Also, since we

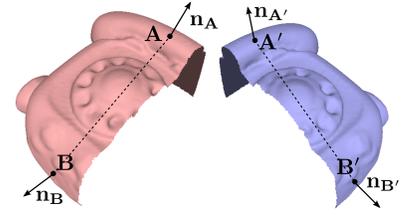


Fig. 4: A source (pink) and a target (blue) PCs are shown with the corresponding pairs of points and normals. These two pairs are the only information used for computing the rigid transformation between the two PCs.

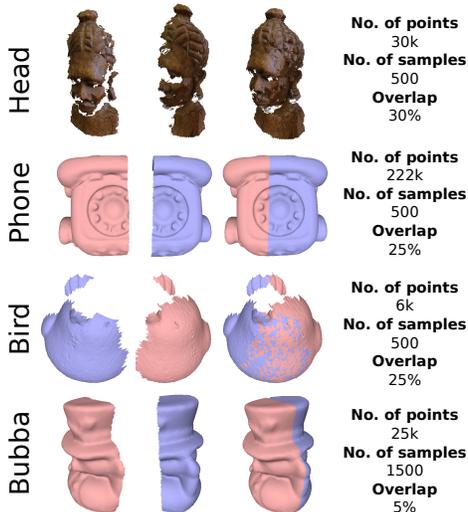


Fig. 5: Different models with small overlap ( $< 30\%$ ) used for testing our proposed method and the Super4PCS algorithm. All datasets were downloaded from [12] and contain ground truth.

are working with 2-point bases, wider bases can be used, yielding less candidates. Experiments in the next section show how our method is able to achieve similar results to Super4PCS while being approximately  $3.6\times - 5.2\times$  faster when using Kinect scans and being two orders of magnitude faster in noise-free datasets.

#### IV. EXPERIMENTS

This section reports a set of experiments performed on several PCs with different sizes, percentages of overlap and levels of noise. The performance of our proposed method is assessed in terms of speed and robustness, and compared with the state-of-the-art method Super4PCS [7] that served as a starting point for our algorithm. The first set of experiments was performed on the models shown in Figure 5 and we report both the alignments obtained for a search limited in time, which is relevant for applications with real-time requirements, and the best possible results when the execution time is not constrained. Since there is ground truth, we show a quantitative evaluation. The second set of experiments consisted in aligning several scans acquired with a Kinect camera, with both methods, and the qualitative results are shown. All these datasets were downloaded from [12].

We used the C++ source code for the Super4PCS algorithm available in [12] and made the necessary modifications to implement our method. For all PCs, the normals to each point were computed using the PlanePCA algorithm [11] with a neighbourhood of 20 points. All tests were performed on a AMD Quad-Core Processor A6-3400M with a speed of 2.30 GHz and 6GB of RAM.

##### A. Quantitative evaluation

The first set of experiments consisted in performing the alignment of the 4 models in Figure 5 with both our method and Super4PCS, for 10 random initial positions of the PCs.

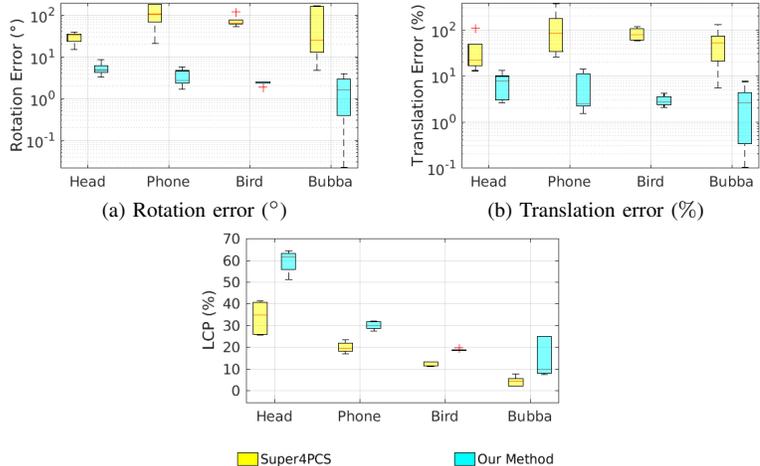


Fig. 6: Results obtained with our and the Super4PCS methods for the 4 models in Figure 5 by limiting the maximum execution time as follows: Head - 1 s, Phone - 2 s, Bird - 1 s, Bubba - 3 s.

As explained in [6], since the LCP measure is being used as the metric for selecting the best alignment hypothesis, Super4PCS and 2PNS allow a very heavy sampling of the PCs. The approximate size of the PCs and the number of samples used are shown in Figure 5. We show quantitative results as the angular magnitude of the residual rotation between the ground truth and the estimated rotations,  $e_R$ , in degrees, and the norm of the difference between the ground truth and the obtained translation vectors,  $e_t$ , in percentage. We started by analysing how well both methods perform when the maximum execution time is limited. We used a threshold of 1 s for the head and bird datasets, 2 s for the phone and 3 s for the bubba point set, as they have decreasing percentages of overlap.

Figure 6 shows the distributions of the best LCPs achieved by RANSAC in each run, and the rotation and translation errors for all models, without any ICP refinement. The first observation is that, in the same amount of time, our method significantly outperforms Super4PCS, providing much smaller rotation and translation errors and larger LCPs. More importantly, unlike Super4PCS, our method was able to provide acceptable results as can be seen in Figures 6a and 6b that the maximum median rotation and translation errors are below  $5^\circ$  and  $5\%$ , respectively. Also, it never diverged, with the maximum errors being  $e_R = 8.4^\circ$  and  $e_t = 13.8\%$ .

In Figure 7, we show examples of the alignments obtained with Super4PCS and our method in the experiment with limited time, for all models. The PCs, in arbitrary initial positions, are accurately aligned when using our method, which significantly facilitates a subsequent step of ICP refinement. On the other hand, the alignment provided by Super4PCS is very poor, as initially shown in the boxplots of Figure 6, not being sufficiently accurate for a subsequent refinement to be applied. This experiment demonstrates that

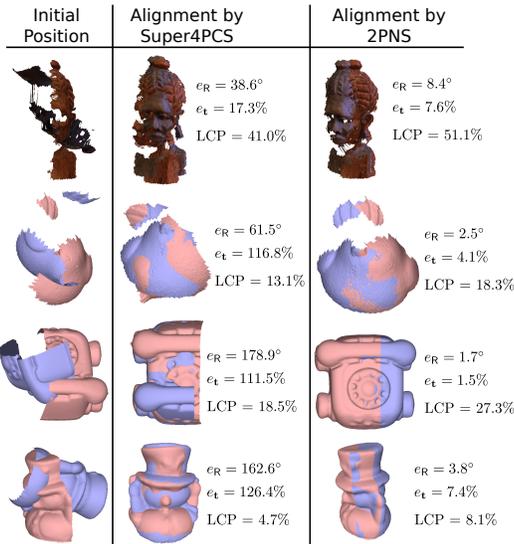


Fig. 7: Alignment results and corresponding LCP and rotation and translation errors (without ICP) obtained with Super4PCS and our method 2PNS in the experiment with limited time.

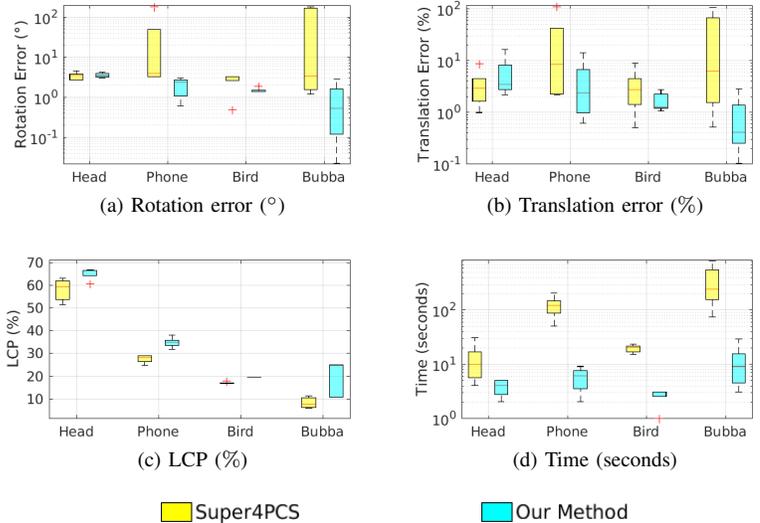


Fig. 8: Best possible results obtained with our and the Super4PCS methods using the same configuration parameters as the experiment corresponding to Figure 6, but without setting a maximum execution time threshold.

in applications that require fast processing, our method can be used as it is very fast in providing sufficiently good results for a refinement step to converge to the global minimum.

In order to evaluate the best performance in terms of accuracy of alignment that both our and the Super4PCS methods are able to achieve, we removed the time limit and tested the methods with the same parameters as in the first part of this experiment. Results are shown in Figure 8 and include not only rotation and translation errors and the best LCP, but also the distribution of the execution times.

As expected, the accuracy of Super4PCS dramatically increased, reaching median errors of less than  $4^\circ$  and 10% in rotation and translation, respectively, as opposed to errors over  $100^\circ$  and 200% that were obtained in some cases of the first part of this experiment. The increase in the LCP is coherent with this decrease in the registration errors. However, despite running for over 15 minutes, Super4PCS still did not manage to converge to a good solution in a few initial poses of the phone and bubba datasets due to the very small overlap. As an example, the divergence cases for the bubba dataset corresponded to alignments similar to the one shown in Figure 7, due to the symmetric nature of the model that yields a local minimum. On the other hand, our method was able to achieve good solutions in all cases in less than 30 seconds for the bubba dataset and less than 10 seconds for the other ones. For the less problematic datasets head and bird, both methods performed well, with our method being able to find slightly better solutions (with larger LCPs) in about 30% and 13% of the time, respectively.

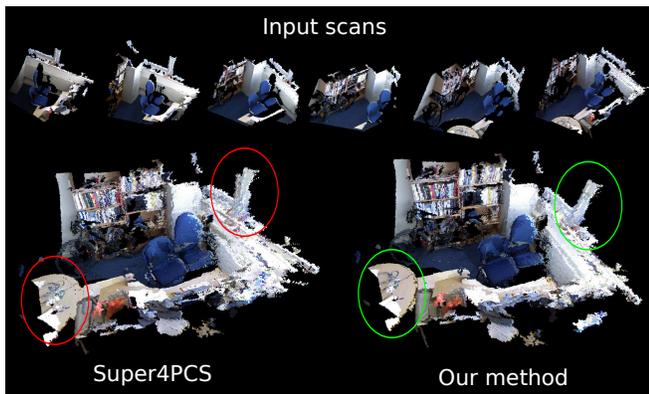
In order to estimate the speed up achieved by our method, we measured the time it requires to reach solutions as good as Super4PCS's, *i.e.* to achieve values of LCP equal or higher than the ones obtained by Super4PCS. For the bubba

and phone datasets, we were able to achieve speed-ups of  $117\times$  and  $58\times$ , respectively. The reason for these very high speed-ups is that these are synthetic noise-free datasets, and thus we can be very restrictive in selecting the threshold for extracting congruent pairs, leading to very few high-quality candidates for providing alignment hypotheses, and thus significantly decreasing computational time. For the bird and head datasets, our algorithm runs  $19.2\times$  and  $7.8\times$  faster, respectively, which is significantly better than the speed up reported for the Super Generalized 4PCS method [8] that is between  $1.3\times$  and  $6.5\times$ . Remark that the results provided in [8] are only for datasets with more than 30% of overlap, suggesting that the method is not able to perform well for smaller overlaps. Also, for some models shown in [8], Super4PCS outperforms Super Generalized 4PCS, whereas our method is always superior. This indirect comparison to Super Generalized 4PCS shows the superiority of our approach w.r.t. both state-of-the-art methods.

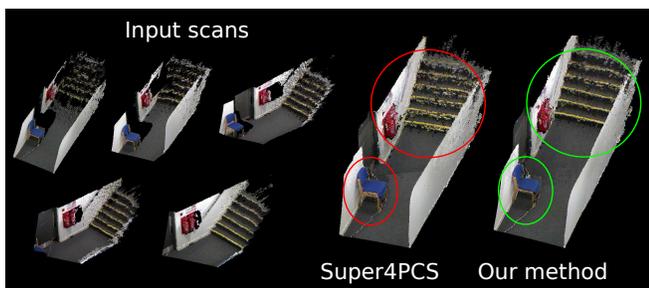
To conclude, this experiment demonstrates the importance of our new method as it shows that it is able to achieve similar or even better results than Super4PCS in a fraction of the time. Also, by limiting the execution time, while Super4PCS clearly fails, our method still performs well, with only a slight decrease in accuracy. This means that our method is very fast in finding a proper solution and increasing the execution time simply leads to more accuracy, which may not be crucial since the method performs coarse alignment and should be followed by a refinement step.

### B. Experiments on Kinect scans

The second set of experiments consisted in registering several scans acquired by a Kinect of an office and a hall. Since there is no ground truth, we show the alignments obtained with Super4PCS and our method in Figure 9. For



(a) Registration of 6 Kinect scans acquired in an office



(b) Registration of 5 Kinect scans acquired in a hall

Fig. 9: Registration results obtained using Super4PCS and our method on two Kinect datasets. The total execution times were (a) Super4PCS - 36s, our method - 10s, and (b) Super4PCS - 57s, our method - 11s. Registration is performed for every pair of consecutive scans. The sequences are sorted from left to right and top to bottom.

the sake of fairness, we used the same degree of sampling in both methods and did not limit the execution time. Also, all results are shown without any ICP refinement.

Overall, both methods were able to correctly align all the Kinect scans in the two examples of Figure 9, with our method being slightly more accurate, especially in the hall sequence (Figure 9b), as shown in the areas identified with ellipses. Regarding the office dataset in Figure 9a, while Super4PCS took approximately 36 s to align all 6 scans, our method was 3.6 times faster, requiring only 10 s. A better alignment can be observed in the table and window areas in the output provided by our algorithm. In the 5-scan hall example (Figure 9b), our method also provides an overall more accurate alignment not only in the stairs and the fire extinguisher areas but also in the other end of the PC, near the chair and the wire on the floor. Super4PCS and our method took 57 s and 11 s, respectively, to align the 5 scans, corresponding to a speed up of  $5.2\times$ . The state-of-the-art algorithm Super Generalized 4PCS [8] reports a speed up of  $4\times$  w.r.t. Super4PCS in a sequence of scans acquired with the Kinect. Again, our method is superior since we are able to perform  $5.2\times$  faster.

Although our method still performs significantly faster than Super4PCS on these real datasets, it can be noticed that

the decrease in computational time is not as evident as in the experiments from Section IV-A. This is due to the fact that, unlike most of the models used in the first set of experiments, these Kinect scans are noisy and possibly contain outliers, providing a less accurate estimation of the normals. Thus, the angular threshold used for extracting congruent sets has to be relaxed, leading to more alignment hypotheses and hence a higher computational time. By achieving speed ups of up to  $5.2\times$ , this experiment confirms that including normals in the estimation of rigid transformations is very beneficial since very accurate results are obtained while significantly decreasing computational time. Since there are established algorithms for normal estimation, as long as the PC is sufficiently dense (as happens in the case of scans acquired with depth cameras), we believe there is no obvious reason for performing the registration using only points.

## V. CONCLUSIONS

We propose 2PNS, a new method that significantly advances the state-of-the-art in terms of global 3D registration of PCs in arbitrary initial poses. Experiments show that our method performs better than the state-of-the-art Super4PCS, providing more accurate alignments, in approximately  $1/5$  of the time when working with scans acquired by depth cameras that are contaminated by noise and outliers.

As future work, we intend to study until which extent working simultaneously with multiple bases randomly extracted from the source PC is beneficial. The idea is that, since there would be multiple hypotheses for the base, computational time would not be wasted in testing all the sets congruent to a base that is noisy or does not correspond to an overlapping area. We believe that this can provide a significant increase in computational efficiency.

## REFERENCES

- [1] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence*, 1992.
- [2] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Symposium on Geometry Processing*, 2004, pp. 23–31.
- [3] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Computer Graphics Forum (Symposium on Geometry Processing)*, vol. 32, no. 5, pp. 1–11, 2013.
- [4] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE TPAMI*, 2016.
- [5] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, no. 4, Apr 1987.
- [6] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," in *ACM SIGGRAPH*, 2008.
- [7] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global point-cloud registration via smart indexing," *Computer Graphics Forum*, vol. 33, no. 5, pp. 205–215, 2014.
- [8] M. Mohamad, M. T. Ahmed, D. Rappaport, and M. Greenspan, "Super generalized 4pcs for 3d registration," in *3D Vision (3DV)*, 2015.
- [9] H. Badino, D. F. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *ICRA*, 2011.
- [10] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, 1981.
- [11] K. Jordan and P. Mordohai, "A quantitative evaluation of surface normal estimation in point clouds," in *IROS*, 2014.
- [12] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," <http://geometry.cs.ucl.ac.uk/projects/2014/super4PCS/>, 2014.