Machine Vision and Applications (2024) 35:41
https://doi.org/10.1007/s00138-024-01510-w

RESEARCH

Addressing the generalization of 3D registration methods with a featureless baseline and an unbiased benchmark

David Bojanić[1] · Kristijan Bartol[2] · Josep Forest[3] · Tomislav Petković[1] · Tomislav Pribanić[1]

Received: 22 August 2023 / Revised: 12 December 2023 / Accepted: 16 January 2024

Abstract
Recent 3D registration methods are mostly learning-based that either find correspondences in feature space and match them, or directly estimate the registration transformation from the given point cloud features. Therefore, these feature-based methods have difficulties with generalizing onto point clouds that differ substantially from their training data. This issue is not so apparent because of the problematic benchmark definitions that cannot provide any in-depth analysis and contain a bias toward similar data. Therefore, we propose a methodology to create a 3D registration benchmark, given a point cloud dataset, that provides a more informative evaluation of a method w.r.t. other benchmarks. Using this methodology, we create a novel FAUST-partial (FP) benchmark, based on the FAUST dataset, with several difficulty levels. The FP benchmark addresses the limitations of the current benchmarks: lack of data and parameter range variability, and allows to evaluate the strengths and weaknesses of a 3D registration method w.r.t. a single registration parameter. Using the new FP benchmark, we provide a thorough analysis of the current state-of-the-art methods and observe that the current method still struggle to generalize onto severely different out-of-sample data. Therefore, we propose a simple featureless traditional 3D registration baseline method based on the weighted cross-correlation between two given point clouds. Our method achieves strong results on current benchmarking datasets, outperforming most deep learning methods. Our source code is available on github.com/DavidBoja/exhaustive-grid-search.

Keywords Featureless 3D registration · Baseline · Benchmark creation methodology · Benchmark difficulty assessment

1 Introduction

3D point cloud registration is the task of finding the rotation and translation that aligns the *source* point cloud to the partially overlapping *target* point cloud. It arises as a subtask in many different computer vision applications such as: 3D reconstruction [1], object recognition and categorization [2], shape retrieval [3], robot navigation [4] and is still an active research area [5, 6].

The typical registration pipeline consists of several steps: detecting features by finding salient points or patches of the point clouds, extracting features by describing those detected points or patches, matching features by finding the correspondences between the features of the point clouds, removing outlier correspondences by satisfying a specific criteria, and estimating the transformation by using only confident correspondences to find the alignment with the highest inlier ratio. These steps can be learning-based or handcrafted as in the traditional approaches.

The most recent advances have been inspired by the successes of deep learning and the development of novel architectures convenient for point cloud processing, such as PointNet [7] and KPConv [8]. Most of the learning-based approaches follow the typical pipeline by first extracting point cloud features [9–12] and then either applying RANSAC for creating feature-based matches [13–15] and

✉ David Bojanić
david.bojanic@fer.hr

Kristijan Bartol
kristijan.bartol@tu-dresden.de

Josep Forest
josep.forest@udg.edu

Tomislav Petković
tomislav.petkovic.jr@fer.hr

Tomislav Pribanić
tomislav.pribanic@fer.hr

1 Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia

2 TU Dresden, 01062 Dresden, Germany

3 Computer Vision and Robotics Research Institute, University of Girona, Plaça de Sant Domènec 3, 17004 Girona, Spain

Published online: 23 March 2024

filtering out the bad matches [16–18] or learning the whole registration pipeline end-to-end [19–21]. These methods achieve remarkable performance on public benchmarks [13, 22–24], even on very difficult examples with an overlap smaller than 30% [14].

A big limitation, however, of the state-of-the-art methods, which is typical for deep-learning-based methods [25–27], is that the model performance drops on benchmark data that differ from their training data. Most recent methods answer the generalizability question by training on the 3DMatch dataset [13] and evaluating their generalization capabilities on the KITTI [22] or ETH [23] benchmarks. As we argue in Sect. 4.1, however, these benchmark datasets lack data variability, where the datasets are biased toward similar data onto which the feature extraction pipeline can focus on. Additionally, as we show in Sect. 4.1, the current benchmarks have a restricted range of the registration parameters (rotation, translation and overlap), therefore providing less information about the actual quality of a method. Moreover, none of the benchmarks provide the option to assess the quality and robustness of a 3D registration method w.r.t. a single registration parameter. Therefore, none of the current benchmarks can provide an adequate in-depth analysis on the performance and generalization of a 3D registration method.

To address these limitations, we propose a methodology for creating a 3D registration benchmark, starting from a point cloud dataset. The methodology improves on the current benchmarks by allowing an in-depth analysis toward concrete registration parameters and providing a bigger range of variability in the registration parameters. We provide the methodology steps by creating a new version of the FAUST-partial (FP) benchmark [28] based on the FAUST [29] dataset, but the process can be extended to any point cloud dataset, including the already mentioned 3DMatch, KITTI and ETH benchmark datasets. By using the human body point clouds from the FAUST dataset, however, we address the bias in the current benchmarks which are mostly comprised of similar objects, providing a substantially different point cloud distribution than the current datasets. We start by creating 3 different settings for the FP benchmark, where each setting changes the difficulty (easy, medium or hard) for one of the 3 following parameters: rotation range, translation range, or overlap range; whilst fixing the remaining two. By fixing two out of three registration parameters, we can isolate the analysis of the quality of a particular 3D registration method to a single parameter, which is not possible with the current benchmarks. The three difficulty levels of the newly created benchmark datasets provide a bigger variety of parameter ranges for all the three registration parameters, which allows for determining the robustness of a method toward that parameter. We compare in detail the newly created benchmark with the existing benchmarks and conclude that the FP benchmark provides a much more detailed analysis, allowing to answer questions related to the generalization onto different point cloud distributions and different registration parameters. This comparison additionally provides us with a general methodology for assessing the difficulty of a 3D registration benchmark based on the registration parameter range.

Using the newly created FP benchmark, we carry out a thorough analysis of the state-of-the-art methods in Sect. 4 and address the research gap for a more thorough and recent survey (evaluation) of 3D registration methods. Our analysis suggests that most methods are very sensitive to an overlap decrease, somewhat sensitive to a larger rotation range, and not sensitive at all to a larger translation range.

To address the generalization downside of the current feature-based methods, we further propose a straightforward featureless traditional 3D registration method to use as a baseline for comparing with the state-of-the-art methods. We extend the work from [28], that is based on a grid search of the quantized rotation SO(3) and translation $\mathbb{R}^3$ spaces. The best transformation candidate is selected as the solution with the maximum cross-correlation between the voxelized source and target point clouds. Thus, we name the method *exhaustive grid search* (EGS). The EGS shows competitive performance, outperforming most traditional and deep learning methods, as well as achieving state-of-the-art results on the ETH benchmark and several FP benchmarks. The results suggest that the learning-based methods, although remarkable on many public benchmarks, are still not robust enough to be applied to any 3D data. On the other hand, our EGS method performs consistently, regardless of the data distribution and regardless of its parameter choices, providing a robust method with higher applicability (see Sect. 5.6).
In summary, we:

- Propose a new 3D registration method which performs an exhaustive search of the rotation and translation spaces and selects the transformation candidate based on the maximum weighted cross-correlation between the voxelized point clouds;
- Provide a methodology to create a 3D registration benchmark, starting from an existing 3D point cloud dataset, that provides a more informative evaluation w.r.t. existing benchmarks and allows to asses the difficulty of existing 3D registration benchmarks;
- Using the newly proposed methodology, generate a novel FAUST-partial (FP) 3D registration benchmark, which addresses the bias toward similar data in the current benchmarks, provides greater parameter range variability than observed in the current benchmarks, and allows to evaluate the strengths and weaknesses of a 3D registration method w.r.t. a single registration parameter;
- Thoroughly evaluate the generalization performance of a great number of state-of-the-art methods under a com-

mon set of 3D registration metrics and benchmarks and analyze in detail the influence of three registration parameters;

2 Related work

We divide the related works into traditional and deep learning methods. Along with the standard optimization-based and handcrafted feature-based traditional methods, we additionally overview the cross-correlation and Fourier-based methods, since the EGS uses the cross-correlation computed in the Fourier domain as a guiding signal for the registration. Between the deep learning methods, we distinguish the feature learning, robust estimation learning and end-to-end learning methods.

2.1 Traditional methods

Optimization-based The most popular traditional registration method is the iterative closest point (ICP) algorithm. The algorithm selects a subset of points as correspondences, calculates the optimal transformation between the clouds using SVD, and iterates until convergence. The original implementations used point-to-point [30] and point-to-plane [31] distances for finding the tentative correspondences, but many other strategies have been proposed [32–35]. GO-ICP [33] proposes a branch-and-bound scheme and claims the global optimality of the algorithm. The 4-point congruent sets (4PCS) algorithm [36] and its variants [37, 38] are based on the idea that there exist sets of four coplanar points whose alignment corresponds to the alignment of the point clouds. To select the correspondences, RANSAC is used, and ICP is applied for refinement.

Cross-correlation based We single out related methods that use the cross-correlation to find the 3D alignment. [39] determines the 3D rotation using a sensor (accelerometer or magnetometer) attached to the 3D scanner, followed by a 3D cross-correlation between the voxelized point clouds to determine the translation. A group of works [40–42] uses the 2D cross-correlation between the reprojected 3D data to a 2D space to either determine the correspondences or the final registration.

Differently, our approach uses the 3D cross-correlation to determine both the 3D rotation and translation in order to align the point clouds. Additionally, our method does not mix 2D and 3D information, but rather uses only the 3D information of the given point clouds to align them. To the best of our knowledge, there are no works that explicitly use the 3D cross-correlation to determine both the rotation and translation to register two point clouds.

Fourier-based We overview related works that use the Fourier domain to compute the 3D registration between two point clouds. [43] uses the Fourier domain to align slices (2D images) of 3D brain MRI volumes by searching for only a single rotation angle along the Z-axis that obtains the biggest cross-correlation. A set of works [44–49] find the 3D alignment completely in the frequency domain by leveraging the fact that the magnitude of the Fourier transform of the displaced voxelized point cloud decouples the rotational from the translational component of the 3D alignment. The translational component is then found using the phase correlation or phase matching.

Differently, our method is 3D-based (not 2D) and does not find the rotation in the Fourier domain. Instead, we find the rotation by sampling the SO(3) space, which increases the rotation estimation robustness since the Fourier rotation theorems used in these works are only valid in the continuous case; introducing numerical issues if discretized. Moreover, these methods perform significantly worse when the point clouds do not (nearly) completely overlap.

Handcrafted feature-based These methods first extract potential correspondences between the point clouds using the computed features and then find the transformation using RANSAC. Similar to the image keypoint-based methods such as SIFT [50], 3D feature-based methods focus on keypoint detection [51–53] and their distinctive description [54–62]. Fast global registration [63] refines the initial correspondences computed using the FPFH [55] descriptor and optimizes the Black-Rangarajan duality between robust estimation and line processes to estimate the 3D alignment.

Differently, our method does not require any features, keypoints, or their description to estimate the transformation. Instead, it exhaustively searches the rotation and translation spaces, avoiding the common pitfalls of the feature-based methods, and increasing its generalization capabilities.

2.2 Deep learning methods

Feature learning Instead of handcrafting distinctive features, keypoint detection and description can be learned. 3DMatch [13] transforms patches into volumetric voxel grids of truncated distance function (TDF) values and processes them through a 3D convolutional network [64, 65] to output local descriptors. Followed by 3DMatch and the popularity of deep learning, many other works propose to learn keypoint detection [14, 15, 66, 67] and description [27, 68–75]. Most of these works are learned by optimizing some version of the contrastive loss [76, 77] between the descriptors of matching and non-matching points and then by applying RANSAC to select the final correspondences and find the transformation.

Robust estimation learning Instead of creating even better features for the process of registration, these methods focus on removing outliers from a given set of features or correspondences, prior to estimating the rotation with RANSAC, GC-RANSAC [78] or CG-SAC [79]. [80] classifies the given

correspondences into inliers or outliers and computes the transformation from the given inliers. [17] uses triplets of correspondences to cast a vote in the 6D Hough space to vote for a particular transformation. [16] selects the transformation with the most inliers from a list of transformation estimations computed by using the confidence of each given correspondence. [81] removes outliers from given correspondences by using a two-stage branch-and-bound algorithm to find a simpler (1+2) and (2+1) degrees of freedom for the rotation and translation, respectively. [82] finds consistent correspondences between two sets of features by building the adjacency matrix of a graph whose nodes represent the potential correspondences and weights on the links the pairwise agreements between the potential correspondences. [18] uses non-local channel spatial attention layers to obtain more reliable contextual information and uses the work from [82] to find consistent correspondences. [83] proposes a decoupled approach to solve in cascade for the scale, rotation and translation of a truncated least squares registration formulation using given correspondences. [84] considers a second-order spatial compatibility measure to compute the similarity between correspondences. From these, they find reliable initial correspondences that form consensus sets, based on which a rigid transformation can be found. [85] jointly learns the FCGF [68] features along with the outlier removal. [86] learns a matching matrix to match DGCNN [87] rectified virtual point features, after which they use Procrustes to solve for the transformation matrix.

End-to-end registration learning There are many recent approaches that learn not only feature description, but also the subsequent matching step, thus learning end-to-end. The first group of these methods [9–12, 88–91], pioneered by the deep closest point [9], follow the ICP idea by (iteratively) establishing soft correspondences and then applying weighted SVD to obtain the transformations. The second group of methods [19–21, 92, 93], represented by PointNetLK [19], use the PointNet architecture [7] or similar global description strategy to iteratively regress the transformation based on the global feature vectors. The third group of methods [94–97] use mechanisms of self-attention and cross-attention to densely back-propagate the encoded superpoint features and choose the final transformation from candidates of superpoint matches.

2.3 Generalization to other datasets

Several recent methods [16, 27, 71, 72] attempt to generalize to datasets other than training. All of these methods demonstrate a significant performance retention on novel datasets when, for example, evaluating 3DMatch-trained models on KITTI [16, 27, 71]. However, most of the results [15, 27, 71, 72] only show that the computed descriptors have a high registration recall by presenting the feature-matching-recall metric, never actually evaluating the quality of the 3D registration. As will be seen, many methods still struggle to generalize when encountered with completely unseen data.

2.4 3D registration surveys

Recent survey papers on 3D point cloud registration [98–102] provide a grouping of the traditional and learning methods, a detailed overview of the key elements of each method, the current benchmarks used in the literature and the different evaluation metrics used. Additionally these papers also present the results for some of the methods. However, most of the results have been gathered from previous papers. Since there are multiple benchmarks with multiple metrics, the gathered results are mostly comprised of only a few methods. Therefore, the current literature is lacking of an in-depth analysis on the results of the current state-of-the-art 3D registration methods. In order to work toward the goal of a fully robust and generalizable method, a thorough comparison is necessary. We provide a detailed analysis of 33 of the current state-of-the-art methods on three established benchmarks (3DMatch, KITTI and ETH) and our newly created FP benchmark. For comparison, the survey papers mention only 10 (or less depending on the paper) out of the 33 methods we compare.

3 Method description

Let $\mathcal{X} \in \mathbb{R}^{N\times 3}$ be the *source* point cloud and $\mathcal{Y} \in \mathbb{R}^{M\times 3}$ the *target* point cloud. The goal of rigid 3D registration is to find the homogeneous transformation $\mathbf{T} \in \mathrm{SE}(3)$ that best aligns $\mathcal{X}$ to $\mathcal{Y}$. The rigid transformation $\mathbf{T}$ is composed of a rotation component $\mathbf{R} \in \mathrm{SO}(3)$ and a translation component $\mathbf{t} \in \mathbb{R}^3$.

To find the correct rotation and translation, we perform an exhaustive search over the parametrization of the rotation and translation spaces (also called the search space). We divide our method into 3 consecutive steps: *pre-processing*, *cross-correlation* and *estimation*, as shown in Fig. 1. Optionally, an additional *refinement* step can be added to further refine the results. In this section, we first introduce the search space parametrization and the general pipeline, whereas in Sect. 5 we discuss the results of the different tested strategies for each of the 4 steps. The final estimation of the rigid transformation is provided in Eq. 12.

Search space parametrization To parametrize the SO(3) space, we first create a geodesic polyhedron $\{3, 5+\}_{4,0}$ comprised of 162 vertices [103], each lying on a unit 2-sphere equidistant with its neighbors. These vertices are used as a uniform sample of $\mathbb{S}^2$. Next, for each point on the 2-sphere, we uniformly sample $\mathbb{S}^1$ using an angle step S. Each combination of a point on $\mathbb{S}^2$ (denoted as axis) and point on $\mathbb{S}^1$ (denoted as angle) creates an angle-axis representation

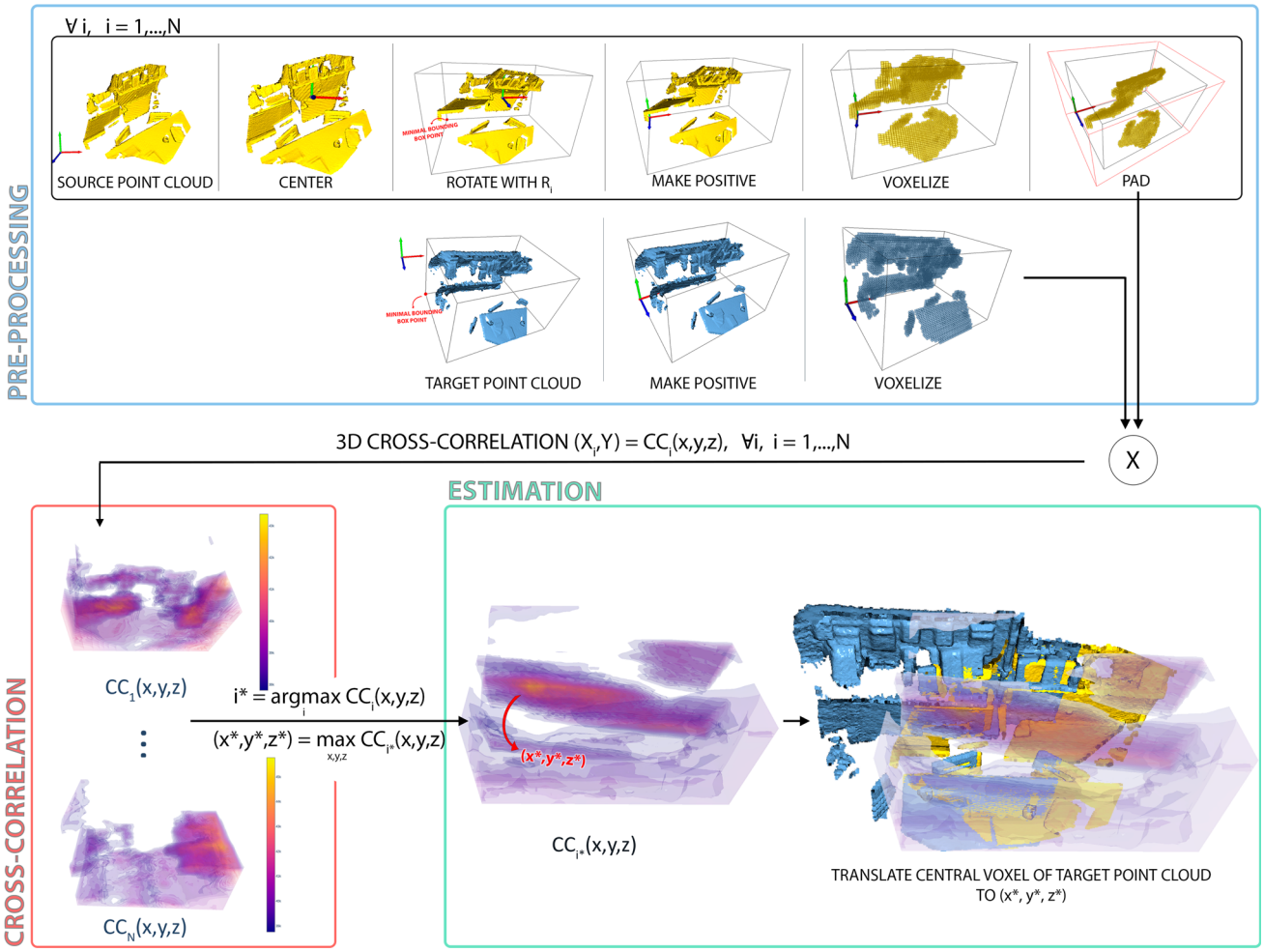

Fig. 1 The proposed pipeline. The method is divided into 3 consecutive steps: *pre-processing*, *cross-correlation* and *estimation*, after which an optional *refinement* step can be added. The *pre-processing* step prepares the initial data and outputs N voxelized source volumes and one target volume. The *cross-correlation* step performs the 3D cross-correlation over each source volume and the target volume. The cross-correlation volumes $CC_i(x, y, z)$ are heatmaps that should indicate higher (indicated in yellow on the volumes) or lower (indicated with purple on the volumes) matching between the source $\mathbf{X}_i$ and target $\mathbf{Y}$ volumes at the corresponding voxels. White spaces are present because the cross-correlations values are clipped so only the upper cross-correlation range is visible. Finally, the *estimation* step finds the solution from the output volumes by using the maximal cross-correlation from all the given volumes

of a rotation. This results in $N = 162 \times (360/S)$ non-unique rotations that can be converted to rotation matrices $R_i, i = 1, \ldots, N$. The non-uniqueness of the rotations follows from having opposite axes present in the sampling of $\mathbb{S}^2$. We remove these duplicate rotation matrices by iteratively rejecting the ones where the norm of their difference equals 0. Note that this step is only computed once, prior to any registration.

The translation space is inherently parameterized by the voxelization process of the given point clouds. The possible translations hence correspond to the centers of the source point cloud voxels and are therefore dependent on the voxelization resolution (VR). More details are provided in the next few sections.

Pre-processing First, we center and rotate the source point cloud $\mathcal{X}$ around the origin using the N precomputed rotation matrices R_i:

$$\mathcal{X}_i = R_i(\mathcal{X} - t_{\mathcal{X}}^{\text{CENTER}}) \tag{1}$$

$$t_{\mathcal{X}}^{\text{CENTER}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{X}[i, :] \in \mathbb{R}^3 \tag{2}$$

obtaining $\mathcal{X}_i$, $i = 1, \ldots, N$, where $[:, :]$ indicates the row and column-wise indexing.

Next, we make all the point clouds coordinates positive by translating their minimal bounding box point into the origin:

$$\mathcal{X}_i = R_i(\mathcal{X} - t_{\mathcal{X}}^{\text{CENTER}}) + t_{\mathcal{X}}^{\text{POSIT}} \tag{3}$$

$$\mathcal{Y} = \mathcal{Y} + t_{\mathcal{Y}}^{\text{POSIT}} \tag{4}$$

where

$$t_{\mathcal{X}}^{\text{POSIT}} = -\begin{bmatrix} \min \mathcal{X}[:,1] \\ \min \mathcal{X}[:,2] \\ \min \mathcal{X}[:,3] \end{bmatrix} \quad t_{\mathcal{Y}}^{\text{POSIT}} = -\begin{bmatrix} \min \mathcal{Y}[:,1] \\ \min \mathcal{Y}[:,2] \\ \min \mathcal{Y}[:,3] \end{bmatrix} \in \mathbb{R}^3 \tag{5}$$

and *min* indicates the minimal element of an array. This step is performed to facilitate the voxelization process.

We then voxelize each source $\mathcal{X}_i$ and target $\mathcal{Y}$ point clouds with a voxel resolution of VR cm. We experiment with different voxelization resolutions and strategies and discuss them in more depth in Sect. 5. Generally, voxelizing a point cloud results in a 3D grid volume where a value of 1 represents that a point from the point cloud is present in that specific grid box (voxel), whereas a value of 0 represents that there are no points from the point cloud present in that specific grid box (voxel). Instead of having a 3D grid with ones and zeros, we set a value of PV (positive voxel) for the filled voxels and a value of NV (negative voxel) for the empty ones. This results in N voxelized source volumes $\mathbf{X}_i$ and one voxelized target volume $\mathbf{Y}$:

$$\mathbf{X}_i(x,y,z),\ \mathbf{Y}(x,y,z) = \begin{cases} PV, \text{if voxel } (x,y,z) \text{filled} \\ NV, \text{ if voxel } (x,y,z) \text{empty} \end{cases} \tag{6}$$

Cross-correlation For each source volume $\mathbf{X}_i$, we perform a 3D cross-correlation with the target volume $\mathbf{Y}$. Essentially, the central voxel of the target volume is translated over each voxel of the source volume where the cross-correlation can be computed by multiplying the overlying voxel values of the two volumes and summing them together. This results in N cross-correlation volumes $CC_i(x,y,z)$ with the same 3 dimensions as the source volume. The volumes can be thought of as discrete heatmaps where higher values should represent higher degrees of matching between the voxelized point clouds. Prior to the cross-correlation, each source volume is padded in order for the target volume to *slide* all over the source volume. We mark with $\mathbf{P} = [n_{\text{left}}, n_{\text{right}}, n_{\text{bottom}}, n_{\text{top}}, n_{\text{front}}, n_{\text{back}}] \in \mathbb{R}^6$ the padding applied to each source volume $\mathbf{X}_i$, where the values represent the number of voxels padded to the left, right, bottom, top, front and back of the volume, respectively. We experiment with different padding sizes and choices in Sect. 5. We make use of the Fourier domain to accelerate the computation of the cross-correlation. Both volumes are first transformed into the Fourier space using the FFT algorithm [104], after which the cross-correlation simplifies to a matrix multiplication [105]. The output is then transformed back with an inverse FFT. More details are given in Sect. 5.6.

Estimation We estimate the rotation matrix $\hat{R}$ that aligns (rotation-wise) $\mathcal{X}$ to $\mathcal{Y}$ using one of the N precomputed rotation matrices R_i. We select the matrix R_i that corresponds to $\mathbf{X}_i$ with the maximal cross-correlation value from the $CC_i(x,y,z)$ volumes. More concretely, we use the index

$$i^* = \underset{i}{\text{argmax}}\, CC_i(x,y,z) \tag{7}$$

to select the estimated rotation matrix $\hat{R} = R_{i^*}$. To estimate the translation, we find the voxel with the maximal cross-correlation value from CC_{i^*}. Then, we translate the central voxel of the target volume $\mathbf{Y}$ to the just found voxel of the CC_{i^*} volume. Since the CC_{i^*} volume corresponds to the source $\mathbf{X}_{i^*}$ volume, we essentially translate the central voxel of the target volume to the voxel of the source volume with the maximal cross-correlation. More concretely, we find the index of the voxel with the maximal cross-correlation value with

$$(x^*, y^*, z^*) = \underset{x,y,z}{\text{argmax}}\, CC_{i^*}(x,y,z). \tag{8}$$

Then, to translate the central voxel of the target volume to it, we use the translation:

$$t_{\text{est}} = \Bigg(\underbrace{\underbrace{-\underbrace{\mathbf{Y}_{\text{CV}}}_{\substack{\text{target}\\ \text{volume}\\ \text{central}\\ \text{voxel}}}}_{\text{move to origin}} \underbrace{-\begin{bmatrix} \mathbf{P}[0] \\ \mathbf{P}[2] \\ \mathbf{P}[4] \end{bmatrix}}_{\substack{\text{padding}\\ \text{displacement}}} + \underbrace{\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix}}_{\substack{\text{max cc}\\ \text{voxel}}} + \underbrace{\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}}_{\substack{\text{move to}\\ \text{center of}\\ \text{voxel}}}}_{\text{move to } (x^*,y^*,z^*)} \Bigg) \times \text{VR} \tag{9}$$

where each value is multiplied by the voxel resolution VR to transform from voxel indices to Euclidean coordinates. The central voxel of the target volume can be computed as:

$$\mathbf{Y}_{\text{CV}} = \begin{bmatrix} \lceil V_x/2 \rceil \\ \lceil V_y/2 \rceil \\ \lceil V_z/2 \rceil \end{bmatrix} \tag{10}$$

where V_x, V_y, V_z are the number of voxels of $\mathbf{Y}$ along the 3 dimensions. Intuitively, the central voxel along a dimension is the middle voxel if the number of voxels is odd, and one on the left of the "middle point" if it's even.

Following all of the steps above, the rigid registration can be summarized as:

$$\left(\hat{R}\left(\mathcal{X} - t_{\mathcal{X}}^{\text{CENTER}}\right)\right) + t_{\mathcal{X}}^{\text{POSIT}} \sim \left(\mathcal{Y} + t_{\mathcal{Y}}^{\text{POSIT}}\right) - t_{\text{est}} \tag{11}$$

where $\sim$ indicates that the left and right parts are aligned.

Since the final rigid transformation needs to align $\mathcal{X}$ to $\mathcal{Y}$, Equation (11) can be rewritten as:

$$\left(\hat{R}\left(\mathcal{X} - t_{\mathcal{X}}^{\text{CENTER}}\right)\right) + t_{\mathcal{X}}^{\text{POSIT}} + t_{\text{est}} - t_{\mathcal{Y}}^{\text{POSIT}} \sim \mathcal{Y} \tag{12}$$

Therefore, the final rotation and translation estimations are:

$$\hat{R} = R_{i*}, \qquad \hat{t} = -\hat{R}\, t_{\mathcal{X}}^{\text{CENTER}} + t_{\mathcal{X}}^{\text{POSIT}} + t_{\text{est}} - t_{\mathcal{Y}}^{\text{POSIT}} \tag{13}$$

Refinement Since the rotation and translation spaces are discretized, the initial alignment can be further refined. We derive the numerical and analytical rotation and translation upper bounds RB and T B in Appendix A. It can be concluded from these bounds that the rough initial alignment provides a good initialization for a fine registration algorithm. We experiment with different refining strategies in Sect. 5.4. In the final case, we use generalized ICP [106] to refine the initial solution since it provided slightly better results. We run $i = 500$ iterations with an adaptive distance threshold based on the q-th quantile of the nearest neighbor distances between the two point clouds. Using an adaptive threshold provides more robustness to the method, since the point clouds from different benchmarks have very different resolutions. As we show in Table 8, however, the final results vary only slightly for different threshold values of i and q, which make the EGS independent of the refinement strategy.

4 Experiments

We evaluate the traditional and deep learning state-of-the-art methods trained on 3DMatch [13] and compare them to the EGS method. We use three established benchmarks: 3DMatch [13], ETH [23] and KITTI [22]; and create a novel FAUST-partial benchmark based on the FAUST dataset [29]. These benchmarks test the generalization abilities in terms of different sensor modalities (RGB-D, laser scanner), different environments (indoor, outdoor), resolution (6mm to 5cm) and completely different structure (from indoor objects to human bodies). Implementation details for all the compared methods are listed in Sect. 6.

4.1 Benchmarks

3DMatch The 3DMatch [13] benchmark dataset contains 46 training, 8 validation and 8 test indoor scenes. Each scene is fragmented into multiple 3D scans that need to be aligned. The scenes represent indoor scans of various rooms such as offices, hotel rooms, kitchens, laboratories, etc. The benchmark has been created by joining several existing benchmarks into one. Hence, it shows the biggest variability in terms of the rotation, translation and overlap parameters. Following standard practice [12–14, 94, 95], we evaluate our EGS method on the 8 test scenes and align all fragments with a minimum overlap of 30%, including the neighboring benchmark pairs that the original benchmark excluded.

KITTI The KITTI [22] benchmark dataset is comprised of 11 sequences of outdoor driving scenarios obtained by a lidar scanner. Compared to 3DMatch, the fragments are much larger, have lower resolution and a different structure. Following common practice [14, 14, 15, 68, 71, 95], we evaluate our EGS method on scenes 8 to 10 using pairs which are at least 10 m away from each other. The ground-truth transformation matrices are refined using ICP [14, 15, 27, 95] since the ground-truth alignment parameters are obtained using the imperfect GPS coordinates of the moving vehicle.

ETH The ETH [23] benchmark dataset consists of 4 scenes mostly comprised of outdoor vegetation. Compared to 3DMatch, the fragments are larger, have lower resolution and have more complex geometries. Following common practice [71, 72, 108], we use only point clouds with an overlap greater than 30%.

FAUST-partial The state-of-the-art learning methods that train on the 3DMatch dataset usually test their generalization capabilities [12, 16, 71, 72, 95, 108] on the KITTI or ETH benchmarks. We argue that the data from these benchmarks are comprised of many similar flat objects, such as walls, tables or floors that make the generalization process easier for a method. When encountered with completely unseen data however (such as 3D human scans), the methods have difficulty generalizing. Additionally, as we show in Figs. 3, 4 and 5, the existing benchmarks lack parameter range variability and do not provide any insights into the robustness of a method w.r.t. a single registration parameter.

To improve the generalization testing of a 3D registration method, we propose a methodology for creating a novel 3D registration benchmark, starting from a point cloud dataset. The methodology improves on the current benchmarks by providing larger 3D registration parameter variability, and more informative evaluations. We indicate the methodology steps by creating a novel FAUST-partial benchmark based on the FAUST [29] dataset, but the process can be extended to any point cloud dataset, including the already seen 3DMatch, KITTI and ETH benchmark datasets.

The FAUST [29] dataset is comprised of 100 human body scans. We divide each scan into multiple overlapping fragments that need to be aligned. The steps to create the fragments are illustrated in Fig. 2. We begin by moving each scan so the xz-plane acts as the floor. We do this by moving the minimal bounding box point of the scan to the origin. Next, we create a regular icosahedron centered at the center of mass of each scan. A regular icosahedron contains 12 points that lie on a unit sphere around its center, each equidistant from its neighbors. We scale the icosahedron to a 1.5-m-radius

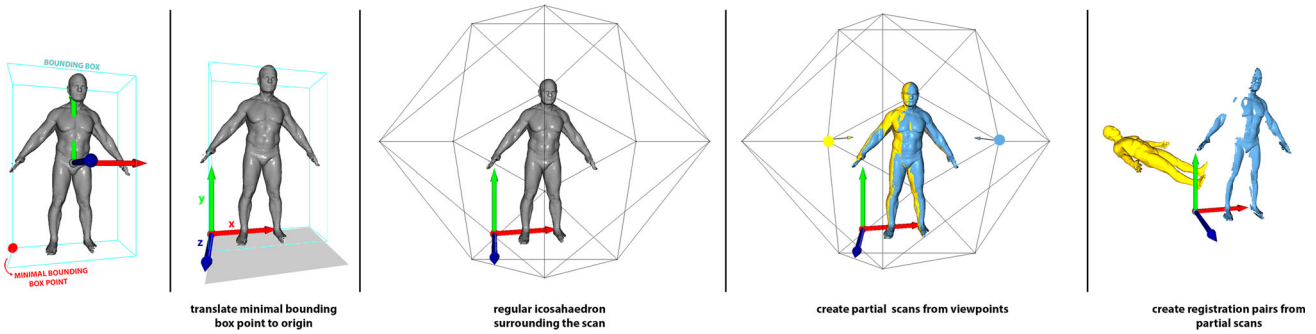

Fig. 2 FAUST-partial benchmark generation. For a given scan from the FAUST [29] dataset, we translate its minimal bounding box point to the origin. Next, we surround the scan with a regular icosahedron. Each point of the icosahedron acts as a viewpoint used to create a partial scan using the hidden point removal algorithm [107]. For two partial scans with a desired overlap, we use a random rotation and translation from the desired ranges to obtain a registration pair for the FAUST-partial benchmark

Table 1 The 9 FAUST-partial benchmark dataset versions

		Easy	Medium	Hard
vary rotation FP–R	rotation	$[-15°, 15°]$	$[-45°, -15°\rangle \cup \langle 15°, 45°]$	$[-180°, -45°\rangle \cup \langle 45°, 180°]$
	translation	$[0\text{m}, 1\text{m}]$		
	overlap	$[60\%, 100\%]$		
	dataset name	FP–R–E	FP–R–M	FP–R–H
vary translation FP–T	rotation	$[-15°, 15°]$		
	translation	$[0\text{m}, 1\text{m}]$	$\langle 1\text{m}, 5\text{m}]$	$\langle 5\text{m}, 10\text{m}]$
	overlap	$[60\%, 100\%]$		
	dataset name	FP–T–E	FP–T–M	FP–T–H
vary overlap FP–O	rotation	$[-15°, 15°]$		
	translation	$[0\text{m}, 1\text{m}]$		
	overlap	$[60\%, 100\%]$	$[30\%, 60\%\rangle$	$[10\%, 30\%\rangle$
	dataset name	FP–O–E	FP–O–M	FP–O–H

From left to right, we increase the difficulty of the parameter we vary, and keep the other two fixed to the easy difficulty

sphere so each scan fits inside it. The icosahedron points are used as the viewpoints for creating the partial views (fragments). For each viewpoint, we use the hidden point removal [107] algorithm to create a partial point cloud. Finally, for each two pairs of viewpoints (i, j) that satisfy a desired overlap criteria (discussed below), we sample a random rotation using 3 Euler angles and a random translation for the x, y and z axes, respectively. We rotate and translate the partial point cloud obtained from viewpoint i to finally get the benchmark registration pair (i, j).

To achieve the variability w.r.t. the registration parameters and have a clear distinction between datasets, we create 3 different benchmark settings. Each setting changes one of the 3 following parameters: rotation range, translation range or overlap range, whilst fixing the remaining two. For the changing parameter, we distinguish 3 levels of difficulty, namely an easy, medium and hard difficulty. Therefore, we obtain 9 different benchmarks; 3 for each setting. Table 1 summarizes the different settings and the attributed acronyms to each of the benchmark datasets for easier reference. As can be seen, the datasets are denominated as FP–$\mathcal{S}$–$\mathcal{D}$, where $\mathcal{S} \in \{R, T, O\}$ indicates the setting (R for rotation, T for translation and O for overlap), whilst $\mathcal{D} \in \{E, M, H\}$ indicates the difficulty level (E for easy, M for medium and H for hard). For every benchmark instance (for example FP–R–E), the difficulty of the setting $\mathcal{S}$ (the rotation R) is determined by the chosen difficulty level $\mathcal{D}$ (the difficulty E) whilst the difficulty of the other two parameters (translation T and overlap O in this case) are kept at their easy difficulty. Therefore, the triplet of datasets FP–R–E, FP–R–M and FP–R–H, for example, indicate an increasing difficulty in the rotation range and a fixed easy difficulty for the remaining translation T and overlap O ranges. By fixing two out of three registration parameters, we can isolate the analysis of the quality of a particular 3D registration method to a single parameter and determine its robustness regarding that parameter. Even though the datasets FP–R–E, FP–T–E and FP–O–E have all the same ranges for all the three parameters (since the difficulty for all the parameters is easy), the datasets are not equal since we sample the rotations and translations for each dataset independently. We

choose to keep all the three seemingly equal datasets in order to showcase the results for different samplings of the same dataset.

To create the three difficulties for each registration parameter, we start by determining the theoretical bounds for each registration parameter: $-180°$ to $180°$ for the rotation, 0 m for the lower translation bound, and 0% to 100% for the overlap. The goal is to find three non-overlapping parameter ranges within those bounds, with increasing difficulty of alignment. To create sensible bounds for each difficulty, we observe the parameter ranges from existing benchmarks. We plot the kernel density plot (KDE) [109, 110], along with its carpet plot, for each parameter in Figs. 3, 4 and 5. The KDE approximates the underlying continuous probability density function generated by the data, by smoothing the binned observation frequencies with a Gaussian kernel. The carpet plot addresses the smoothing pitfalls of the KDE (continuity of the curve where there is no data for example) and shows a histogram-like plot of observed data values. For easier comparison between benchmarks, we additionally normalize each KDE using its maximal binned frequency so that each curve displays a maximal probability density of 1.

To determine the easy, medium and hard rotation bounds, therefore, we observe Fig. 3, where we plot the Euler angle ranges that rotate around the x, y and z axes, respectively. As can be seen, the angles around the x and y axes for the KITTI and ETH benchmarks are around $[-10°, 10°]$. Additionally, the angles around the z axis for the KITTI benchmark are around $[-20°, 20°]$. Since these are small rotations, we choose a similar range, $[-15°, 15°]$, for our easy rotation range. Since we do not want the parameter ranges for the different difficulties to overlap, the lower bound for the medium rotation range therefore needs to be $15°$. By looking at the 3DMatch Euler angle distributions for the three axes, we notice that the range increases to $40°$. Therefore, we create the medium range from $[-45°, -15°\rangle$ and $\langle 15°, 45°]$. Again, since we do not overlap the parameter ranges for the different difficulties, the lower bound for the hard range needs to be $45°$. This range is mostly covered by the z axis in the ETH benchmark, peaking at around $75°$. Therefore, we use the whole remaining angles as the hard parameter ranges $[-180°, -45°\rangle$ and $\langle 45°, 180°]$.

To determine translation bounds, we observe Fig. 4, where we plot the translation distances in meters for each benchmark. As can be seen, the 3DMatch benchmark translations peak at 0.5m and decrease rapidly after 1m. Therefore, we choose the range [0m, 1m] as the easy translation range. Next, we see that for the ETH benchmark, the translations are mostly between 1m and 3m. Therefore, we chose the medium difficulty translation range as $\langle 1\text{m}, 3\text{m}]$. Finally, since the KITTI registration pairs are sampled at a 10m distance, we create the hard difficulty translation range as $\langle 5\text{m}, 10\text{m}]$, in order to include the upper bound of the KITTI translations.

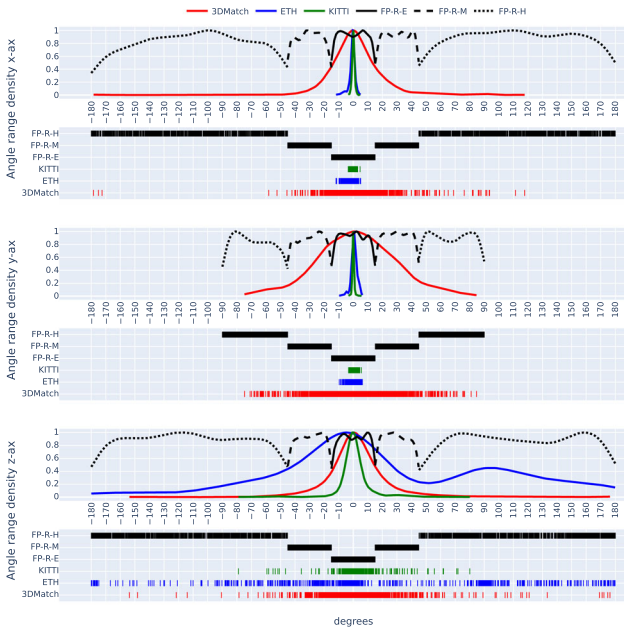

Fig. 3 Euler angle ranges for the x, y and z axes. For each axis, we plot the KDE plot and the carpet plot. To facilitate comparison, we normalize each KDE using the maximal binned frequency

To determine the overlap bounds, we observe Fig. 5. All the benchmarks peak at around 40% overlap, with 3DMatch and ETH having only examples with overlap larger than 30%. Therefore, we determine that the hardest difficulty should have overlaps lower than 30%. On the other hand, anything below 10% overlap is not enough to register, so we deem that the hard overlap range should be $[10\%, 30\%\rangle$. As we can see from Fig. 5, the KITTI and ETH overlaps begin to drop more drastically after 60%. Therefore, we use the $[30\%, 60\%\rangle$ as the medium overlap range. Finally, the remaining $[60\%, 100\%]$ is used for the easy overlap difficulty range. Note that the actual overlap in the FP benchmark is never 100%, since it makes little sense to register two fully overlapping point clouds.

Benchmark comparison To motivate the creation of the FAUST-partial benchmark, we provide a detailed overview of the three established benchmarks, namely 3DMatch, ETH and KITTI, and compare them to the newly generated versions of the FAUST-partial benchmark. Using the analysis, we explain why are the current benchmarks insufficient to provide an in-depth analysis of a registration method.

As already mentioned, Fig. 3 plots the KDE and carpet plot for the Euler angle ranges that rotate around the x, y and z axes, respectively. As can be seen, the 3DMatch benchmark has the largest variability in angle ranges. Since the benchmark is actually a combination of existing 3D registration benchmarks in itself, this intuitively does make sense. The ETH and KITTI benchmarks observe a very small angle variation around the x and y axes and a big angle variation

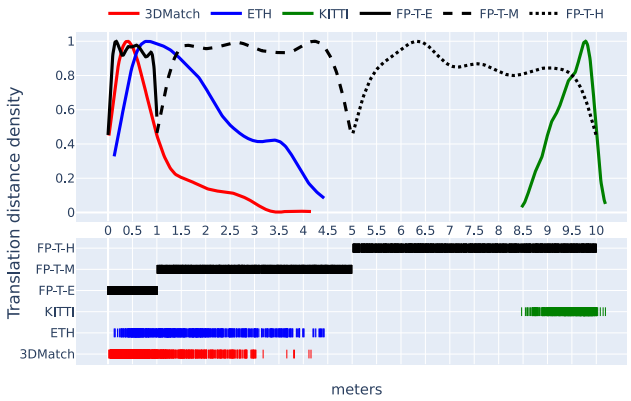

Fig. 4 Translation ranges for each dataset. We plot the KDE and carpet plot using the translation vector norms of each example. To facilitate the comparison, we normalize each KDE using the maximal binned frequency

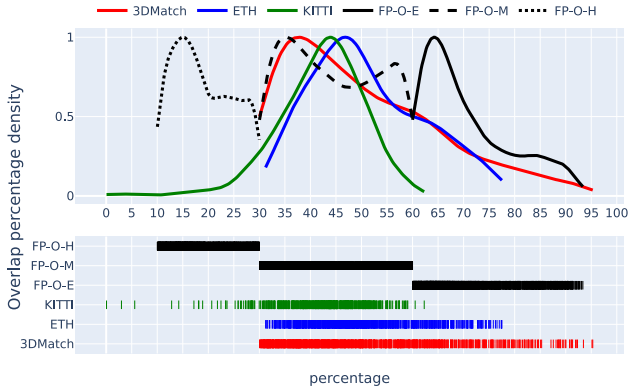

Fig. 5 Overlap ranges for each dataset. We plot the KDE and carpet plot using the overlap between the registration examples. To facilitate the comparison, we normalize each KDE using the maximal binned frequency

around the z ax. This, again intuitively does make sense, since these benchmark datasets are obtained with a 3D scanner fixed to the floor; either on a car (KITTI) or on a tripod (ETH). The newly created FP$-R-\{E, M, H\}$ benchmarks, on the other hand, provide a big variety of angle ranges for all the three axes, covering the whole spectrum of possible values. Additionally, the benchmarks provide a clear difficulty increase of the sampled rotation range without any overlap of the ranges between them. Hence, the benchmark can be used to thoroughly analyze the rotation robustness of a method.

Figure 4 overviews the translation ranges for each benchmark. As can be seen, the translation ranges are clearly connected to the type of scene that is being scanned. The 3DMatch benchmark dataset is comprised of indoor scenes that limit the scanner movement range. Hence, they observe the smallest translation ranges, peaking at 0.5m. The ETH benchmark dataset is comprised of outdoor scenes and allow for greater scanner movement range; hence, the translation ranges increase, ranging mostly from 1m to 3m. The KITTI benchmark dataset is comprised of lidar scans from a moving vehicle sampled at approximately every 10m, as clearly indicated in the Figure. The newly created FP$-T-\{E, M, H\}$ benchmarks cover the whole range between $0-10$m for each difficulty level, without any overlap. Therefore, they can be used to determine the robustness of a 3D registration method toward an increasing level of translation between point clouds.

Figure 5 overviews the overlap range for each benchmark. To compute the overlap percentage, we compute the inlier ratio between the two registration examples. We use an adaptive distance threshold of 3 times the median resolution of the source point cloud. As can be seen, all the datasets have most examples with around 40% of overlap. A clear cut at 30% can be seen for the 3DMatch and ETH benchmarks, which explicitly only take examples with overlap greater than 30%.

Table 2 Benchmark statistics for the 3DMatch, KITTI, ETH and FP benchmarks

	Number point clouds	Number benchmark pairs	Average number points	Average resolution (cm)	Average size (meters)
3DMatch	433	1523	337,258.21	0.6	$2.5 \times 2.0 \times 2.2$
KITTI	6863	555	123,589.28	5.2	$152.7 \times 95.4 \times 11.3$
ETH	132	713	100,625.28	2.7	$33.5 \times 30.4 \times 15.2$
FP-R FP-T FP-O-E	1184	1686	63,917.72	0.3	$0.7 \times 1.7 \times 0.5$
FP-O-M	1173	1935	63,528.61	0.3	$0.7 \times 1.7 \times 0.5$
FP-O-H	1183	1781	64,018.49	0.3	$0.7 \times 1.7 \times 0.5$

The FP–R, FP–T and FP–O–E benchmarks use the same examples with the easy overlap and, therefore, have the same statistics

The newly created FP$-O-\{E, M, H\}$ benchmarks observe three non-overlapping ranges covering an overlap from 10% to 100%. By differentiating the difficulty levels, the benchmarks allow to evaluate the robustness of a method w.r.t. a decreasing overlap between registration pairs.

Even though the current benchmarks are not sufficient to provide an in-depth analysis, we emphasize that they are not rendered obsolete by introducing the FP benchmark. On the contrary, the FP benchmark is complementary and should be used in addition to the current benchmarks when evaluating a 3D registration method. The existing benchmarks still provide a different point cloud distribution in their datasets (indoor and outdoor scans, different resolution, different number of points, etc.) when comparing to the FP benchmark (human bodies). We provide some of those dataset statistics in Table 2.

As can be seen from Table 2, the point clouds in the different benchmarks can still provide an answer to the generalization question regarding different statistics, such as point cloud resolution, number of points, and point cloud size. Therefore, to get a complete picture of a 3D registration method, it should be evaluated using all the provided benchmarks.

Benchmark difficulty The benchmark comparison from the previous section provides a clear indicator of the benchmark difficulty. Having a small rotation, small translation, and a large overlap range decreases the difficulty of a benchmark. Oppositely, having a large rotation, large translation, and a small overlap range increases the difficulty of a benchmark. This intuition is clearly backed by the evaluation results from Sect. 4.4, where the increasing parameter difficulty decreases the number of successful registrations. Therefore, we propose that the analysis given in the previous section be used as an indicator of the difficulty of any 3D registration benchmark.

4.2 Metrics

Following [12, 15, 71, 85, 95] we evaluate the results using the relative rotation error (RRE), the relative translation error (RTE) and the registration recall (RR) measures. The relative rotation error measures the relative angle in degrees between the ground truth R^* and estimated $\hat{R}$ rotation matrices:

$$RRE = \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}^T\mathbf{R}^*) - 1}{2}\right)\frac{180}{\pi} \tag{14}$$

The relative translation error measures the distance from the ground truth $\mathbf{t}^*$ and the estimated $\hat{\mathbf{t}}$ translation vectors:

$$\text{RTE} = \|\mathbf{t}^* - \hat{\mathbf{t}}\|_2 \tag{15}$$

The registration recall measures the fraction of successfully registered pairs of point clouds. A registration is deemed successful (or a true positive in terms of the recall measure) if its RRE and RTE are below predefined thresholds τ_r and τ_t:

$$RR = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} \mathbb{1}_{\{\text{RRE}(i,j)<\tau_r \ \wedge \ \text{RTE}(i,j)<\tau_t\}} \tag{16}$$

where Ω is the set of all the point cloud registration pairs (i, j) in the dataset, $\mathbb{1}$ is an indicator function and $\text{RRE}(i, j)$, $\text{RTE}(i, j)$ indicate the RRE and RTE for registration pairs (i, j). Following standard practice, the final RRE and RTE measurements are averaged only over the successfully registered pairs (i, j) obtained from the RR.

4.3 Parameters

To fully define the proposed registration method, we need to set the parameters of the angle step S, voxelization values PV and NV, and voxelization resolution VR. We use an angle step of $S = 10°$, which determines the number of rotation matrices $N = 162\times(360/S) = 5832$. As already mentioned, we remove duplicate rotations and obtain $N = 3536$ rotation matrices R_i. We use a voxel value of $PV = 5$ for the positive voxel and $NV = -1$ for the negative one. Intuitively, these values promote high cross-correlations in regions with good overlap where filled and empty voxel spaces in both the source and target volumes coincide. More details are provided in Sect. 5.

The only parameter we vary for each benchmark is the voxel resolution VR since the datasets vary greatly in their dimensions ranging from volumes of 152.7m × 95.4m × 11.3m for KITTI to 0.7m × 1.7m × 0.5m for FP on average. We use a voxel resolution VR of 7cm, 75cm, 60cm and 6cm for the 3DMatch, KITTI, ETH and FP benchmarks respectively. For the refinement strategy, we use $i = 500$ iterations of the generalized ICP with $q = 0.25$ for 3DMatch and FP, and $q = 0.80$ for ETH and KITTI.

4.4 Results

3DMatch. Following standard practice [12–14, 94], we evaluate our EGS method on the 8 test scenes and align all fragments with a minimum overlap of 30%. We use the common thresholds $\tau_r = 15°$ and $\tau_t = 30$cm. For a fair comparison, we use the overlaps computed in [14] that slightly differ from ours computed in the previous section. Instead of finding the overlap between complete point clouds, [14] first voxel-downsamples the point clouds and then computes the overlap. The difference between their overlaps and ours is 14 percentage points (pp) on average.

The results presented in Table 3 are divided into two parts: *traditional* and *deep learning* methods. Naturally, the deep learning methods clearly dominate over the traditional methods since they are evaluated on the same dataset they were trained on.

Between the traditional methods, the handcrafted feature-based traditional methods perform better than the optimization-based methods. The reasons are the large initial displacements of the point clouds and the noise in the scans, which are known to influence the optimization-based methods. The best traditional methods, however, are those that focus on filtering the outliers and finding the good correspondences. As can be seen, SC2-PCR achieves the best result between the current traditional methods. It shows that using a second-order spatial compatibility measure as guidance for sampling inliers facilitates an outlier-free set.

The deep learning methods are further divided into feature learning methods, robust estimation methods and end-to-end learning-based methods. Here, the differences of the best methods of each category are much smaller, favoring more the end-to-end learning-based methods. Feature learning methods achieve the lowest recall measures with GeDi at the forefront. Robust estimation methods build on top of those learned features and try to filter out bad correspondences. The best method, CSCE-Net, combines the second-order

Table 3 Results for $\tau_r = 15°$ and $\tau_t = 30$cm on the 3DMatch benchmark

	Method	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
TRADITIONAL	ICP (P2Point)† [30]	6.04	8.25	18.10
	ICP (P2Plane)† [31]	6.59	6.61	15.2
	Super4PCS† [111]	21.6	5.25	14.10
	GO-ICP† [33]	22.9	5.38	14.7
	FPFH+FGR† [63]	42.91	4.96	10.25
	FPFH+SM† [82]	55.88	2.94	8.15
	FPFH+RANSAC-1M† [55]	64.20	4.05	11.35
	FPFH+RANSAC-2M† [55]	65.25	4.07	11.56
	FPFH+RANSAC-4M† [55]	66.10	3.95	11.03
	FPFH+GC-RANSAC† [78]	67.65	2.33	6.87
	FPFH+TEASER† [83]	75.48	2.48	7.31
	FPFH+CG-SAC† [79]	78.00	2.40	6.89
	FPFH+SC2-PCR† [84]	83.98	2.18	6.56
DEEP LEARNING	3DMatch [13]	73.40	2.49	7.69
	FCGF+FGR† [63]	78.93	2.90	8.41
	GeDi [27]	92.97	1.86	7.13
	FPFH+3DRegNet† [80]	26.31	3.75	9.60
	FPFH+DGR† [85]	32.84	2.45	7.53
	FPFH+DHVR† [17]	67.10	2.78	7.84
	VRNet† [86]	72.90	3.12	8.64
	FPFH+PointDSC† [16]	77.57	2.03	6.38
	FCGF+3DRegNet† [80]	77.76	2.74	8.13
	FPFH+CSCE-Net† [18]	83.61	2.11	6.73
	FCGF+TEASER† [83]	85.77	2.73	8.66
	FCGF+SM† [82]	86.57	2.29	7.07
	FCGF+CG-SAC† [79]	87.52	2.42	7.66
	FCGF+RANSAC-1M† [68]	88.42	3.05	9.42
	FCGF+RANSAC-2M† [68]	90.88	2.71	8.31
	FCGF+RANSAC-4M† [68]	91.44	2.69	8.38
	FCGF+DGR† [85]	88.85	2.28	7.02
	FCGF+DHVR† [17]	91.40	2.08	6.61
	FCGF+GC-RANSAC† [78]	92.05	2.33	7.11
	FCGF+PointDSC† [16]	92.85	2.08	6.51
	FCGF+SC2-PCR† [84]	93.28	2.08	6.55
	FCGF+CSCE-Net† [18]	<u>93.47</u>	2.06	6.53
	PointNetLK† [19]	1.61	8.04	21.3
	DCP† [9]	3.22	8.42	21.4
	PCAM† [12]	92.4	2.16	∼ 7
	RegTR† [94]	93.0	<u>1.92</u>	<u>5.92</u>
	GeoTransformer† [95]	**95.0**	1.98	**5.69**
	EGS	84.11	**1.69**	6.46

Methods indicated with † are taken from [18, 84–86, 95]. PCAM results are originally given in meters. All the deep learning methods are trained on the 3DMatch dataset. The results are shown in ascending order w.r.t. the RR measure for each category. We bold and underline the best and the second best results, respectively, for each metric

spatial compatibility from SC2-PCR and spectral matching from SM [82] with a novel channel-spatial contextual layer that uses self-attention mechanisms to aggregate information in the channel and spatial dimensions. The state-of-the-art results are achieved by the end-to-end learning methods. These methods use self-attention and cross-attention [12, 95] mechanism to exchange contextual information between the features of the two point clouds to register. This allows for information sharing between the point clouds that traditional and robust estimation methods cannot achieve.

The EGS achieves the best results between the traditional methods and even comparable with the best deep learning methods by successfully registering 84.11% of the 3DMatch pairs. Additionally, the EGS achieves the lowest rotation error, which indicates that the cross-correlation measure does give an indication into the fitness of two point clouds. Contrary to the deep learning methods however, the EGS is simple and explainable. The effects of different strategies (discussed in Sect. 5) are clear and intuitive, providing insights into the registration process.

Interestingly, most methods seem to average around 2° for the RRE measure, having a much larger RTE measure of $7cm$. This could indicate noisy point clouds, since the dataset contains many flat surfaces that make the rotation *click* into place, but leave the translation handling the noise.

KITTI Following common practice [14, 14, 15, 68, 71, 95], we test our EGS method on scenes 8 to 10 using pairs which are at least 10 m away from each other. We use the common thresholds $\tau_r = 5°$ and $\tau_t = 2$m. The stricter rotation threshold, compared to 3DMatch, reflects the fact that the rotation ranges are much smaller. The more lenient translation threshold reflects the fact that the scenes are large and have a big translation distance between them. We evaluate the generalization from the 3DMatch benchmark dataset to the KITTI benchmark dataset, which poses several challenges: change in scanning modality (from time-of-flight scanner to lidar scanner) and point cloud size (from 2.2m^3 to 86.4m^3 by averaging all three axes).

As can be seen from Table 4, some methods are somewhat able to generalize onto the KITTI dataset. The registration pairs from the dataset are gravity aligned, which is reflected in lower RRE errors compared to the 3DMatch benchmark, since most of the ground-truth rotation comes from rotating around one axis. The fragments are also much bigger than those from 3DMatch (152.7m × 95.4m × 11.3m on average in size compared to 2.5m × 2.0m × 2.2m in 3DMatch) which is reflected in higher RTE errors.

Generally, the learning methods outperform the traditional methods, which intuitively make sense, since the registration pairs showcase a smaller overlap with more noise.

Between the traditional methods, both ICPs do not register a single example. Intuitively, the point-to-point ICP fails since it is a fine registration method, expecting an already close initial alignment of the point clouds. GO-ICP should address this downside and find the global solution to the registration. However, as noted in previous works [33, 83], the registration method is very sensitive to its parameters. We unsuccessfully test for several different parameters noted in Sect. 6. The exception between the traditional methods is SC2-PCR, which achieves good results using the same FPFH features as the RANSAC methods. This could indicate that the FPFH features are not unique enough for different points in the scene or that the scene contains a lot of similar

Table 4 Results for $\tau_r = 5°$ and $\tau_t = 2$m on the KITTI benchmark

	Method	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
TRADITIONAL	ICP (P2Point) [31]	0.00	-	-
	GO-ICP [33]	0.00	-	-
	FPFH+RANSAC-8M [55]	9.91	2.40	36.79
	FPFH+RANSAC-2M [55]	10.81	2.24	33.57
	FPFH+SC2-PCR [84]	**98.74**	0.38	8.87
DEEP LEARNING	D3Feat-rand† [15]	18.47	1.58	37.80
	FCGF+RANSAC† [68]	24.19	1.61	27.10
	D3Feat-pred† [15]	36.76	1.44	31.60
	DIP [72]	51.71	1.02	13.43
	SpinNet† [71]	81.44	0.98	15.60
	GeDi [27]	82.88	0.65	10.99
	FPFH+PointDSC [16]	94.05	0.33	7.44
	FCGF+PointDSC [16]	96.76	0.37	9.45
	FCGF+SC2-PCR [84]	97.66	0.38	9.61
	Predator† [14]	41.20	-	-
	GeoTransformer [95]	67.93	0.51	103.03
	GLORN† [96]	74.30	-	-
	YOHO-O [73]	81.44	1.99	54.25
	YOHO-C [73]	82.16	1.38	39.30
	EGS	94.95	**0.11**	**3.90**

All the methods are trained on the 3DMatch dataset. Results marked with † are taken from [71, 96]. The results are shown in ascending order w.r.t. the RR measure in each category. We bold and underline the best and the second best results, respectively, for each metric

Table 5 Results for $\tau_r = 5°$ and $\tau_t = 60$cm on the KITTI dataset

	Method	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
TRADITIONAL	ICP (P2Point) [31]	0.00	-	-
	GO-ICP [33]	0.00	-	-
	FPFH+FGR† [63]	5.23	0.86	43.84
	FPFH+CG-SAC† [79]	74.23	0.73	14.02
	FPFH+RANSAC† [55]	74.41	1.55	30.20
	FPFH + SC2-PCR [84]	98.38	0.38	8.66
DEEP LEARNING	FCGF + RANSAC† [68]	80.36	0.73	26.79
	GeDi [27]	82.34	0.64	10.54
	FCGF + FGR† [63]	89.54	0.46	25.72
	FPFH+DGR† [85]	77.12	1.64	33.10
	FCGF+CG-SAC [79]	83.24	0.56	22.96
	FPFH+PointDSC [16]	93.51	0.33	7.08
	FCGF+PointDSC [16]	96.40	0.37	9.20
	FCGF+DGR† [85]	96.90	0.34	21.70
	FCGF+SC2-PCR [84]	97.48	0.38	9.47
	FCGF+CSCE-Net† [18]	97.84	0.32	20.89
	FPFH+CSCE-Net† [18]	98.74	0.33	7.05
	FCGF+DHVR† [17]	**99.10**	0.29	19.80
	GeoTransformer [95]	15.14	0.55	38.75
	YOHO-O [73]	54.41	1.70	33.19
	YOHO-C [73]	70.09	1.26	30.37
	EGS	94.59	**0.11**	**3.61**

All the methods are trained on the 3DMatch dataset. Results marked with † are taken from [18, 84]. The results are shown in ascending order w.r.t. the RR measure for each category. We bold and underline the best and the second best results, respectively, for each metric

structures. Only by filtering out bad correspondences with geometric properties, does the SC2-PCR remove the non-unique matches and achieves good results.

The same trend is also present in the learning methods, where the best performing methods are in the category of robust estimation methods which filter out the bad correspondences. We can also can notice a gap between these methods and the remaining feature-based and end-to-end learning methods. Intuitively, since the KITTI benchmark data is much noisier than the 3DMatch benchmark data, robust estimation methods have the advantage of addressing that noise by eliminating bad correspondences.

The interval for the translation error (RTE) goes from around 7cm to around 50cm with the exception of GeoTransformer having RTE of 103.03cm. Even though it displays the highest RTE measurement, it still achieves a recall of 67.93% which indicates that using the common practice threshold of 2m can be misleading. Hence, we additionally show the results for a stricter $\tau_t = 60$cm threshold in Table 5.

As we immediately notice, when comparing Table 4 with Table 5, the biggest drop in performance can be seen by the end-to-end deep learning methods GeoTransformer and YOHO, with an average drop of 30.59 recall percentage points. The remaining methods showcase less than 1 percentage points drop in recall (less than 55 registration pairs), which indicates that the translation error was already low for those methods. The performance drop from the end-to-end learning methods comes from their large memory footprint [95]. Because these methods have very large models that need to fit onto a GPU, they need to compromise by subsampling and scaling down the point clouds. Since larger point clouds, such as the ones in the KITTI dataset, mean a greater number of voxels, these methods need to use a much coarser voxelization in order to be able to register the examples, which in turn affects the results.

The EGS outperforms most deep learning methods and achieves a 94.95% and 94.59% recall (for the two thresholds, respectively), which makes it competitive with the best methods; lagging behind only 4 percentage points on average from the best result. The EGS, however, achieves the best RRE and RTE measures, outperforming by 3.49cm on average the second best RTE result.

Compared to the 3DMatch results, we can notice that the robust estimation learning methods, along with our EGS, follow an upwards trend, achieving better results on the KITTI benchmark. The reason is that KITTI is an *easier* (on average) benchmark to register compared to the 3DMatch benchmark, w.r.t. the registration parameters. However, it is a *harder* benchmark to register w.r.t. the noise in the point clouds, but, because these methods are robust to noise, the results improve. On the other hand, the end-to-end and feature-based methods are affected by the noise and, therefore, follow a downward trend.

Table 6 Results for $\tau_r = 5°$ and $\tau_t = 30$cm on the ETH dataset

	Method	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
TRADITIONAL	GO-ICP [33]	1.54	1.32	21.75
	ICP (P2Point) [30]	2.10	0.57	4.20
	FPFH-2M [55]	66.20	1.13	6.76
	FPFH-8M [55]	66.34	1.09	6.48
	FPFH+FGR [63]	66.34	1.09	6.48
	FPFH+SC2-PCR [84]	85.41	0.91	6.08
DEEP LEARNING	GeDi [27]	86.54	1.18	5.09
	FPFH+PointDSC [16]	41.94	0.90	6.78
	FCGF+PointDSC [16]	77.42	0.94	4.05
	FCGF+SC2-PCR [84]	92.85	0.81	4.05
	GeoTransformer [95]	4.91	0.69	21.08
	DIP [72]	62.41	1.94	14.71
	SpinNet [71]	73.07	1.20	5.35
	YOHO-O [73]	79.94	2.16	16.11
	YOHO-C [73]	84.85	1.95	16.17
	EGS	**94.25**	**0.42**	**1.83**

All the methods are trained on the 3DMatch dataset. The results are shown in ascending order w.r.t. the RR measure in each category. We bold and underline the best and the second best results, respectively, for each metric

ETH Following common practice [71, 72, 108] we register only point clouds with an overlap greater than 30%. We set the thresholds to $\tau_r = 5°$ and $\tau_t = 30$cm. The stricter rotation threshold, compared to 3DMatch, reflects the fact that the rotation ranges are much smaller. Similarly to the 3DMatch benchmark, we use the overlaps computed from [108] for a fair comparison. The average difference between their overlaps and ours is 8 percentage points. We evaluate the generalization from the 3DMatch benchmark dataset, which poses several challenges: change in scanning modality (from time-of-flight scanner to laser scanner), point cloud size (from 2.2m^3 to 26.3m^3 by averaging the three axes) and point cloud structure (from *flat* tables, floors and walls to *scattered* vegetation).

As can be seen in Table 6, the learning methods outperform again the traditional methods. The general performance of the traditional methods is worse than on the KITTI benchmark. The slight increase in performance from the optimization-based methods can be attributed to having a few closer initial alignments of the registration pairs, since the recall increases for only 1.82 percentage points. The loss in performance from the feature-based traditional methods, on the other hand, can be attributed to the difference in point cloud distribution; whereas the KITTI point clouds contain much more structure inferred from the roads, the ETH point clouds contain more *fuzzy* vegetation.

The learning methods tend to struggle with generalizing from the 3DMatch dataset. The best learning method is SC2-PCR, with a recall of 92.85%. Differently than the results on the KITTI benchmark, however, the remaining robust estimation method PointDSC does not achieve great results. The key difference is that PointDSC learns to embed the input correspondences into a higher-dimensional space using the 3DMatch dataset, whereas SC2-PCR does not. This, in turn, makes the process of generalization harder for PointDSC. Surprisingly, GeoTransformer achieves the lowest recall results of 4%. Similarly to the KITTI benchmark, this result can be partly explained by the scaling of the input point clouds. As the authors point out [95], the method suffers from a big memory footprint, which in turn means that the downsampling rate needs to be balanced for performance and efficiency. For bigger and denser point clouds, the solution is to scale them in order to simulate the density and inlier rate of the training 3DMatch dataset, which does not always work.

Our EGS method achieves the best result on the ETH benchmark, outperforming all the traditional and learning methods. Additionally, the EGS achieves the lowest RRE and RTE measures. Since the ETH benchmark has less big flat surfaces than the previous two benchmarks, the EGS suffers less from its main limitation, and therefore achieves better results. More details are provided in Sect. 5.5.

Compared to the 3DMatch results, we can notice a downward trend in the results of the learning methods, which affirms our hypothesis that learning methods have difficulties generalizing onto different datasets. The larger noise factor in the ETH benchmark influences all the learning methods, that seem to prefer more rigid structures, as those that they have been trained on. Compared to the KITTI results, the feature-based and end-to-end methods seem to improve. This indicates that the learned features seem to transfer better onto the ETH dataset than onto the KITTI dataset. The robust estimation methods, on the other hand, perform worse because they try to eliminate the outliers using geometric properties in a dataset with much a much noisier point cloud distribution coming from the vegetation.

FAUST-partial We set the thresholds to $\tau_r = 10°$ and $\tau_t = 3$cm. The stricter thresholds reflect the fact that the fragments from FAUST-partial are much smaller in volume than all the other datasets. As such, misalignments on smaller fragments are much more noticeable and erroneous. We evaluate the generalization from the 3DMatch dataset, which poses several challenges: change in the scanning modality (from time-of-flight scanner to multi-view stereo), the point cloud size (from 2.2m^3 to 0.9m^3 by averaging the three axes) and the point cloud structure (from *flat* tables, floors and walls to *curvy* human bodies).

In Table 7, we evaluate GO-ICP [33], SC2-PCR [84], GeDi [27] and GeoTransformer [95] on the FAUST-partial benchmark. The chosen methods represent the state-of-the-art methods from each category, namely the traditional optimization, traditional feature, deep learning feature, robust estimation and end-to-end learning-based categories from Table 3. The FAUST-partial benchmark datasets allows to

Table 7 Faust-partial results for different dataset difficulties

	Method	Easy			Medium			Hard		
		RR (%) ↑	RRE (°) ↓	RTE (cm) ↓	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
FP–R	GO-ICP [33]	0.00	–	–	0.06	1.87	2.71	0.00	–	–
	FPFH+SC2-PCR [84]	99.64	0.95	0.43	94.54	1.43	0.60	75.21	1.76	0.78
	GeDi [27]	**99.76**	1.629	1.162	**99.94**	1.66	1.14	**99.41**	1.70	1.63
	FCGF+SC2-PCR [84]	98.46	1.21	0.82	91.93	1.67	1.02	85.77	1.77	1.08
	GeoTransformer [95]	64.12	2.29	1.57	55.93	2.26	1.63	47.75	0.47	1.69
	EGS	99.64	**0.005**	**0.002**	97.92	**0.003**	**0.003**	78.00	**0.01**	**0.007**
FP–T	GO-ICP [33]	0.00	–	–	0.00	–	–	0.00	–	–
	FPFH+SC2-PCR [84]	**99.76**	0.93	0.43	99.53	0.92	0.43	99.58	0.93	0.43
	GeDi [27]	99.47	1.68	1.16	99.70	1.65	1.15	**99.70**	1.63	1.14
	FCGF+SC2-PCR [84]	98.34	1.25	0.83	98.34	1.24	0.82	98.22	1.25	0.82
	GeoTransformer [95]	66.25	2.32	1.59	64.29	2.29	1.58	64.18	2.27	1.57
	EGS	99.70	**0.005**	**0.002**	**99.82**	**0.005**	**0.002**	98.81	**0.005**	**0.002**
FP–O	GO-ICP [33]	0.00	–	–	0.00	–	–	0.00	–	–
	FPFH+SC2-PCR [84]	**99.88**	0.91	0.43	84.70	1.68	0.80	**38.85**	2.42	1.23
	GeDi [27]	99.64	1.69	1.16	75.40	2.14	1.45	8.70	2.56	1.76
	FCGF+SC2-PCR [84]	98.52	1.26	0.83	63.00	1.98	1.27	17.80	2.68	1.72
	GeoTransformer [95]	63.94	2.30	1.57	22.07	2.45	1.94	2.64	2.57	2.22
	EGS	99.47	**0.067**	**0.035**	**88.06**	**0.030**	**0.017**	37.06	**0.477**	**0.234**

FP–R indicates the dataset where the rotation is varied from an easy-to-hard setting, whereas the translation and overlap are kept at the easy difficulty. Similarly, the FP–T and FP–O alter the translation and overlap setting, respectively, and keep the remaining parameters fixed

analyze each method regarding the three primary registration parameters: the rotation range, the translation range and the overlap range.

As can be seen from FP–R in Table 7, GeDi is very robust to the increasing rotation range, achieving an almost perfect recall of 99% on all the three difficulties. All the other methods observe a small dip in the recall with the medium difficulty, and a larger dip in the recall with the hard difficulty w.r.t. the easy difficulty. The RRE and RTE metrics naturally follow the inverse relationship and increase with the increasing difficulty. The exception to this is GO-ICP, which only registers a few examples for the FP–R–M dataset. Following the author guidelines, we center and scale the point clouds and run several different parameter choices to improve the results. However, we observe that the method still does not register any examples. Analyzing the results, it would seem that the method suffers mostly from wrongly estimated translations, with the rotations just above the RRE threshold of 10°. More details are provided in Sect. 6. The SC2-PCR method seems more robust to the rotation changes when using the FCGF features compared to the FPFH ones, which indicates that the FCGF training produces rotation-invariant features. The EGS achieves the second best results on two out of three benchmarks, namely the FP–R–E and FP–R–M datasets. Additionally, it achieves the best RRE and RTE measures for all the three datasets. These results indicate, however, that the SO(3) parametrization is still lacking in uniformity since bigger rotation ranges should not, in theory, affect the method.

Analyzing FP–T in Table 7, we can clearly see that the change in translation does not seem to affect the registration recall. All the methods seem to be invariant to the translation. Again, the exception to this is the GO-ICP method that does not register a single example. The EGS outperforms all the methods on the FP–T–M and achieves the second best result on FP–T–E. Again, it achieves the best RRE and RTE measurements.

Analyzing FP–O in Table 7, we can clearly see that the overlap greatly affects the registration results. GeDi and GeoTransformer seem to take the biggest hit from a lower overlap where the recall drops to 8.70% and 2.64%, respectively. As noted by the authors [27], the low overlap between the point clouds contains partial structures with little geometric information which leads to registration failure. Interestingly, contrary to the FP–R scenario, the FPFH features seem to be more robust than the FCGF ones w.r.t. the smaller overlap. This would indicate that, even though the FCGF features are more distinctive, they ignore the overlap with little geometric information. In contrast, the EGS achieves the best recall on FP–O–M and the second best result on FP–O–H. This would

indicate that the cross-correlation is a good indicator of the overlap region between two partial point clouds.

5 Ablation study

In Table 8, we evaluate the different strategies tested on the 3DMatch benchmark. Each strategy is then analyzed to get key insights into the registration process. We mark with *Ref* the reference result (under column #) already presented in Table 3. Each of the remaining results varies a single part of the method pipeline, such as the voxelization, the filling, the rotation or the refinement strategy of the reference result *Ref*.

5.1 Voxelization strategy

We test several different voxelization resolutions VR ranging from 3 to 9 cm. As can be seen from results #1 - #5, the voxelization resolution does not affect significantly the results. Contrary to intuition however, we notice that a lower voxel resolution does not automatically improve the results. The reason is that a coarser resolution registers the *global* shape of the scene instead of being affected by the noise more present at the lower resolutions.

5.2 Filling strategy

As already noted, voxelizing a point cloud results in a 3D grid volume where a value of $PV = 1$ represents that a point from the point cloud is present in that specific grid box (voxel). Contrary, a value of $NV = 0$ represents that there are no points from the point cloud present in that specific grid box (voxel). We test this simplest voxelizing strategy under result #6 in Table 8. As can be seen, this strategy does not perform well compared to the others. To improve the registration results, we experiment with $PV = 5$ and $NV = -1$. Intuitively, the idea is to promote high cross-correlations in regions with good overlap where filled and empty voxel spaces in both the volumes coincide. More concretely, we encourage the filled voxels to coincide by giving them a higher value in the cross-correlation ($5 \times 5 = 25$ in the cross-correlation operation); encourage less (but still do not penalize) empty voxels to coincide by giving them a positive value in the cross-correlation ($-1 \times -1 = 1$); and discourage any of the leftover cases ($5 \times -1 = -5$). As can be seen from result #7 in Table 8, this strategy slightly degrades the RR measure, whilst improving the RTE.

Along with the PV and NV values, we additionally experiment with the value of the padded voxels using the parameter PDV. As seen from the reference result *Ref*, using the padding value of -1 drastically improves the results w.r.t. results #6 and #7 that use a value of 0. This strategy allows the method to register point clouds with smaller overlap because it does not penalize the motion of the source point cloud beyond the volume of the target point cloud.

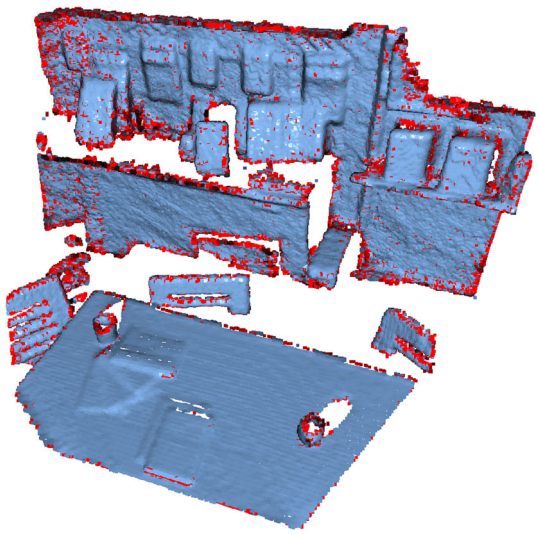

Fig. 6 Removing points from flat surfaces using the difference of normals (don) [112]. The blue points represent the whole point cloud. The red points represent the retained points after the don filtering. As can be seen, points from flat surfaces are mostly rejected

We experiment with additional voxelizing strategies that are based on the same idea of promoting higher cross-correlations in salient regions: *importance* and *layering*. The importance strategy tries to emphasize salient points on the point cloud that should be prioritized in the registration process. The voxels containing these *important* points receive a higher value than the remaining filled voxels. Therefore, this strategy uses the already seen $PV = 5$ and $NV = -1$ with the addition of an intermediary voxel value $IV = 2$. Voxels with salient points are given a value of PV, whereas non-salient points are given a value of IV. The reasoning for this strategy is discussed more in the Limitations section. To find non-salient points, we use the difference of normals (don) [112] to find points on flat surfaces. First, we compute the normals at each point using two neighborhood radius distances, $r_1 = 1$cm and $r_2 = 3$cm. The neighborhood radius distances determine the points used for finding the spanning plane from which the normal is computed. Intuitively, if a point in the point cloud is located on a flat surface, its two computed normals should have similar directions. Hence, we compute the difference of the two normal vectors and find its L^2 norm. Finally, we threshold these norms to retain only salient points not located on flat surfaces. An example of the don filtering can be seen in Fig. 6. As we see from result #8 in Table 8, this strategy does not improve the recall results, but improves on the rotation and translation errors. Hence, we do not use it as a method of choice since the voxelizaton process is more complex than the one presented in the reference *Ref* strategy.

Table 8 The different strategies tested for the EGS method

#	VR (cm)	FILLING $[PV, NV, PDV]$	ROTATIONS	S	N	REFINEMENT	ADDITIONS	RR (%) ↑	RRE (°) ↓	RTE (cm) ↓
	7	$[5, -1, -1]$	AA, $\{3, 5+\}_{4,0}$	10	3536	x	x	75.97	4.90	13.67
Ref	7	$[5, -1, -1]$	AA, $\{3, 5+\}_{4,0}$	10	3536	gen ICP, q=0.25, i=500	x	84.11	1.69	6.46
1	3	$[5, -1, -1]$	AA, $\{3, 5+\}_{4,0}$	10	3536	gen ICP, q=0.25, i=500	x	76.10	1.71	6.48
2	5							82.40	1.65	6.42
3	6							82.86	1.69	6.43
4	8							83.26	1.67	6.40
5	9							83.98	1.66	6.48
6	7	$[1, 0, 0]$	AA, $\{3, 5+\}_{4,0}$	10	3536	gen ICP, q=0.25, i=500	x	76.49	1.67	6.44
7		$[5, -1, 0]$						70.52	1.67	6.42
8		Importance						83.06	1.66	6.35
9		Layering						83.06	1.67	6.53
10	7	$[5, -1, -1]$	Euler	15	6364	gen ICP, q=0.25, i=500	x	80.56	1.69	6.49
11			Euler, Limited	15	1886			82.27	1.72	6.47
12			Euler, Limited	10	6177			**86.41**	1.72	6.54
13			HOPF	x	4608			79.51	1.68	6.35
14			Super-Fibonacci	x	3536			75.71	1.61	6.21
15			AA, $\{3, 5+\}_{2,0}$	15	281			74.05	1.70	6.19
16			AA, $\{3, 5+\}_{2,0}$	10	913			75.25	1.65	6.29
17			AA, $\{3, 5+\}_{4,0}$	15	2289			82.14	1.73	6.42
18			AA, $\{3, 5+\}_{4,0}$, +	24	4531			79.45	1.81	6.75
19			AA, $\{3, 5+\}_{8,0}$	30	4368			74.20	1.80	6.74
20	7	$[5, -1, -1]$	AA, $\{3, 5+\}_{4,0}$	10	3536	gen ICP, q=15, i=500	x	83.45	1.83	6.93
21						gen ICP, q=35, i=500		83.85	1.65	6.19
22						gen ICP, q=45, i=500		81.81	**1.57**	5.91
23						gen ICP, q=55, i=500		78.79	**1.57**	**5.88**
24						gen ICP, q=65, i=500		74.52	1.76	6.26
25						gen ICP, q=75, i=500		67.83	2.03	6.87
26						gen ICP, q=25, i=100		83.91	1.73	6.63
27						gen ICP, q=25, i=30		83.39	1.81	6.87
28						ICP (P2Point)		82.27	1.94	7.38
29						ICP (P2Plane)		83.85	1.73	6.67
30	7	$[5, -1, -1]$	AA, $\{3, 5+\}_{4,0}$	10	3536	gen ICP, q=0.25, i=500	fps subsample	83.39	1.66	6.35
31							don subsample	79.25	1.66	6.27
32							fps+don subsample	76.63	1.63	6.22

The columns in order refer to the voxelization strategy, filling strategy, rotation strategy, angle step, number of precomputed rotation matrices, refinement strategy, additional parameters, and the evaluation metrics RR, RRE and RTE defined in Sect. 4.2. All the results are a variation of the reference result marked with *Ref* under column # and are divided into several categories depending on the parameter that is being varied; results from #2-#5 vary the voxelization strategy, results from #6-#9 vary the filling strategy, results from #10-#18 vary the rotation strategy, results from #19-#28 vary the refinement strategy and finally the results from #29-#31 describe some additional changes in the reference result. For clarity, we only mark the varied parameters in the table, whilst the remaining ones are left equal to the reference *Ref* result. For the rotation strategy, we denote with *AA* the angle-axis representation, with Euler Limited the limited range for the Euler angles. For the refinement strategy, we denote with q the quantile for the distance threshold and with i the number of iterations used. For the Additions column, we denote with don the difference of normal method, and with fps the farthest point sampling method. Please refer to the text for more details

The *layering* strategy adds a layer of $LV = -2$ voxel values around flat surfaces beside the classical $PV = 5$ and $NV = -1$. The reasoning behind this is that sometimes the registration is very close to the solution, but needs a further push to *snap* into place. By providing negative values on borders of flat surfaces, we promote exactly that. As can be seen from result #9 in Table 8, however, the recall is very similar to the *importance* and reference *Ref* results and does not additionally improve the result.

In Fig. 7, we show the cross-correlation volumes for different voxelization strategies under the ground-truth rotation matrix. As can be seen, most methods tend to suffer from wrong or local maximas except our strategy of choice. Since in this registration example, the correct translation voxel is located on the bounds of the volume, the methods using $PDV = 0$ fail and push the translation more toward a bigger overlap. The *importance* strategy prioritizes the translation onto falsely important points not located on the floor and the *layering* strategy moves the translation further from the bounds since the LV values push them further along the floor plane. The strategy of choice, with $PV = 5$ and $NV = -1$, suppresses these local maximas and pushes the translation into the correct voxel location, as can be seen from the image.

5.3 Rotation strategy

We test several different rotation parametrizations to cover the SO(3) space. We start with the simplest case of uniformly sampling 3 Euler angles α, β and γ that rotate around the x, y and z axes, respectively. Each angle is sampled with an angle

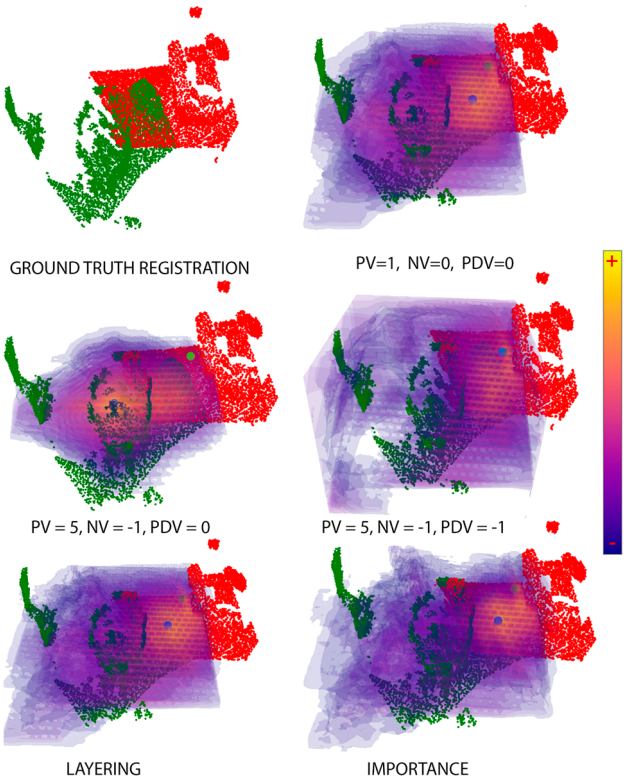

Fig. 7 The different voxelization strategies tested. PV and NV are the voxelization values for the filled and empty voxels. PDV is the padding value. Yellow regions indicate higher cross-correlation values, purple regions indicate lower cross-correlation regions. There are white spaces present in the cross-correlation volume because the values are clipped so only the top 10% of values are shown. The light green dot indicates the ground-truth location where the central voxel of the red point cloud should be located. The blue dot indicates the estimated location where the central voxel of the red point cloud should be located according to the cross-correlation from the EGS method. For more details, refer to Sect. 5

step S, from the range $\langle -\pi, \pi \rangle$ for α and γ and the range $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$ for β. Starting with result #10 from Table 8 we use an angle step of $S = 15$ and obtain good results of 80.56% recall. This strategy, however, is comprised of $N = 6364$ rotation matrices, unnecessarily increasing the runtime of the algorithm. We therefore limit the range for all the three Euler angles to $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$ in result #11 to combat the computation time. As we can notice, the recall further increases to 82.27%. The reason behind this slight increase in results is that by decreasing the range of the rotation, we eliminate big rotations as an option for the EGS to chose from. Since the 3DMatch dataset is comprised of indoor scans, the rotations from the registration are always in the lower ranges. Consequentially, we remove the possibility for the EGS to make obscure registrations between floors and walls, or wrong walls (discussed in more detail in Sect. 5.5). We further reduce the angle step to $S = 10$ in result #12 to receive a finer discretization of the rotation space. This parametrization achieves the best recall results of 86.41%. However, it increases the number of rotations again to $N = 6177$ and does not behave consistently on the other benchmarking datasets. Hence, we do not use it as the final solution.

Uniformly sampling the Euler angles does not result in uniformly sampling the rotation space SO(3). Intuitively however, a more uniform parametrization could result in a better representation of the rotation space, and potentially have a smaller need on the number of rotations N. In the search for a more uniform discretization of the rotation space, we experiment with the Hopf [113] and Super-Fibonacci [114] parametrizations. For the Hopf parametrization, we use the rotations computed in [115], originally proposed in [113]. They start by sampling the sphere $\mathbb{S}^2$ with the HEALPix [116] representation, and sampling the circle $\mathbb{S}^1$ uniformly. Next, the sphere sampling is converted into spherical coordinates, which, together with the $\mathbb{S}^1$ sampling, can be used as a sampling of the Hopf coordinates. The Hopf coordinates are finally converted into $N = 4608$ quaternions. The Super-Fibonacci [114] parametrization uses the Fibonacci sampling twice to generate points in a cylinder, which are then mapped to a 3-sphere using a volume preserving mapping. As can be seen from #13 and #14, however, the results do not seem to reflect our intuition, since the recalls slightly drop to 79.51% and 75.71% from the *Ref* result.

Inspired by the uniformity of SO(3), we test a similar method of sampling that results in being our method of choice. The main idea is to sample the $\mathbb{S}^2$ sphere using a geodesic polyhedron and sample the $\mathbb{S}^1$ circle with a simple uniform interval sampling. Then, the points on the sphere act as the axis and the points on the circle act as the angles for the angle-axis rotation representation. We experiment with several samplings of the $\mathbb{S}^2$ sphere and several angle steps S for the $\mathbb{S}^1$ sampling. In the results #15 and #16, we start with the geodesic polyhedron $\{3, 5+\}_{2,0}$, a regular shape with 42 vertices on the unit sphere, all with equidistant neighbors, to sample the $\mathbb{S}^2$ sphere. In #15, we use an angle step of $S = 15$ for sampling the circle $\mathbb{S}^1$. The results for this option are pretty low, which intuitively does make sense. This parametrization is coarse and is not a representative cover of SO(3). Increasing the sampling angle step $S = 10$ in result #16, helps the EGS achieve better results, however still lower than the reference result. In results #17 and #18, as well as the reference result *Ref*, we increase the number of vertices of the polyhedron by splitting the edges of each face, resulting in $\{3, 5+\}_{4,0}$, with 162 vertices. This in turn increases the number of rotation matrices N, since each vertex represents the axis of rotation. All the results seem to benefit from this increase, expect result #18, in which, we increase the number of vertices, keeping only those in the positive hemisphere of the sphere. The reasoning for this experiment stems from the fact that an axis, and its opposite negative axis, represent the same rotation if we sample the angle from $\mathbb{S}^1$

uniformly. By removing the *negative* axes, and adding twice the *positive* axes, we could, in theory, have a better and finer representation of the SO(3). Unfortunately, the intuition is not reflected in the results, achieving a 79.45% recall. Result #17, on the other hand, does achieve better results than its counterpart with polyhedron $\{3, 5+\}_{2,0}$. Improving on that, the reference result *Ref*, decreases the angle step $S = 10$ to achieve a finer resolution, and better results. Increasing the number of rotations even further, with the polyhedron $\{3, 5+\}_{8,0}$, does not improve the results further, as can be seen from result #19, indicating that the discretization of the rotation space is unnecessarily saturated which introduces noise into the rotation choosing.

Comparing the number of rotation matrices N with the recall RR, we can somewhat see a pattern. The highest and lowest recalls are achieved for the biggest and smallest number of rotations $N = 6177$ and $N = 281$. Whilst a higher number of rotations allow the method to have more rotations to choose from, it is also important to have a good representation of the SO(3) space, as seen from results #11, where using only $N = 1886$ rotations achieves a recall of 82.27%; just under 2 percentage points lower than the reference *Ref* result.

Comparing the angle step S with the recall RR, we can see a clear pattern of linearity: as the angle step decreases, the recall improves. Comparing results #11 - #12, #15 - #16, #17 - *Ref*, we see an average increase of 2.43 percentage points when going from an angle step $S = 15$ to $S = 10$.

5.4 Refinement strategy

We experiment with a few refinement strategies, namely point-to-point ICP (P2Point) [30], point-to-plane ICP (P2Plane) [31], and generalized ICP [106]. As can be seen from results #20 to #29, the different version of ICP perform similarly, with generalized ICP taking a slight advantage. Therefore, we choose it as our refinement strategy. We further test several choices for the hyperparameters of the generalized ICP algorithm. In results #20 - #25, we test different options for the ICP inlier ratio hyperparameter. To find the inlier ratio, we use a quantile threshold q of the nearest neighbor distance for all the points in the source point cloud. As we can see, staying in the reasonable inlier threshold range ($< 50\%$) for the 3DMatch data does not affect the results gravely. In results #26 and #27, as well as the reference result *Ref*, we test the maximum allowable number of iterations generalized ICP can perform. As can be seen, the best recall is achieved using 500 iterations. However, the results for 100 and 30 iterations achieve very similar performance of 83.91% and 83.39% recall.

All the experiments for the refinement strategy indicate that the EGS is not dependent on the performance of the refinement algorithm and is not susceptible to its hyperparameter optimization.

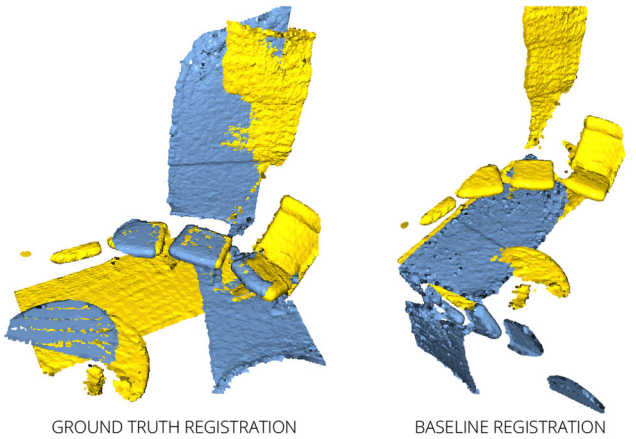

Fig. 8 EGS registration failure case. The wall of one point cloud is registered onto the floor of the other point cloud

5.5 Limitations

The main limitation of the EGS stems from aligning wrong big flat surfaces. This can be reflected in wrong translations, where walls or floors are slightly offset, wrong rotations, where one point cloud is rotated so their floors coincide, or simply wrong matching of flat surfaces, such as aligning wrong walls, floors and tables. One such extreme example can be seen in Fig. 8 where the wall of one point cloud is registered onto the floor of the other point cloud. Each point cloud in this example has more than 80% of points located on a flat surface; either on the floor, wall or table.

In order to fully alleviate this issue, further knowledge should be introduced into the model. Since our method is featureless, it does not have the information necessary to distinguish walls from floors or tables; in short, big flat surfaces are dealt with in the same manner. Introducing further knowledge, however, opposes our approach of having a simple method, along with our initial hypothesis that learning feature-based methods are unable to generalize onto different benchmarks. We experiment, therefore, by using geometric *features* which can be deterministically extracted from each point cloud and are pose invariant, such as the difference of normals (don), farthest point sampling (fps), the *importance* strategy, etc. We overview of the strategies that we experiment with, in order to address the problem of aligning flat surfaces.

We start by purposefully voxelizing the point clouds using $PV = 5$ and $NV = -1$ to discourage such registration and promote those in which both the positive voxels (those with values PV) and negative voxels (those with values NV) are aligned onto each other. However, when big flat surfaces are present in the point clouds, there are cases where the

alignment of the positive voxels from these surfaces trump other, more meaningful alignments. In other words, aligning flat surfaces results in big cross-correlation values because of numerous alignments of positive voxels; and consequentially numerous 5×5 summands in the cross-correlation.

To address this issue even further, we try to remove points from flat surfaces with the *subsampling* strategy. In result #31, we use the difference of normals (don) filtering mentioned in the Filling strategy section, to remove points from flat surfaces. In result #32 we reduce that subsampled set of points even further, using the fps algorithm to pick a set of 5000 points with the farthest distances. As can be seen from the results, these strategies lower the recall. The reason for this is that in many cases, the flat surfaces are also a guidance for the method to align the surfaces together. Therefore, to leave the flat surfaces in the point clouds, we experiment with the *importance* strategy, which is halfway between the *subsampling* and reference *Ref* strategy regarding their filling strategy; whereas the subsampling strategy removes points on flat surfaces, the importance strategy assigns lower weights to the flat surfaces. As already discussed in the filling strategy section, we emphasize salient points using the don sampling. Even though this strategy restores the original recall, it does not improve onto the reference *Ref* result.

By using a *subsampling* strategy that does not discriminate toward flat surface points, such as the farthest point sampling, the results do not change at all, compared to using all the points. As can be seen from result #30, subsampling to 5000 points from an average of 338000 points per cloud from the 3DMatch dataset, the results lower for only 0.72 percentage points. This means that the EGS is robust to the number of points in the point cloud and does not require dense point clouds to be able to register them.

Even though the alignment of flat surfaces presents a problem for our method, analyzing the results on the 3DMatch, KITTI and ETH benchmarks, however, we notice that it does not pose a severe problem. Despite the fact that the point clouds from these datasets mostly contain big flat surfaces (walls, floors, roads, pavements, etc.), we can see from Tables 3, 4, 5 and 6, that our method still obtains very high results on all the three benchmarks, even achieving the best results on the ETH benchmark. This is because the point clouds also contain objects with distinguishable shape, such as indoor appliances or outdoor vegetation, which steers our method onto focusing onto these shapes instead of prioritizing the flat surfaces.

5.6 Computational complexity

We discuss the computational complexity of the EGS method in terms of the average runtime and resources needed for registering two point clouds from the 3DMatch benchmark. These metrics directly correlate to the applicability of a method, so we compare them to the latest learning methods. Because of their large memory footprint, the learning methods only use 5000 points from each point cloud. Therefore, to have a fair comparison, we use the farthest point sampling to get 5000 points from each point cloud and re-run the EGS on the 3DMatch benchmark.

Before comparing the runtime and resources, however, we provide the results for the EGS on the 3DMatch, KITTI and ETH benchmarks using only 5000 points from each point cloud.

As can be seen from Table 9, the results stay almost the same, compared to using all the points from the point clouds. Therefore, we can fairly compare the runtime and resources with the learning methods knowing that the final results of

Table 9 Results for the EGS method on the 3DMatch, KITTI and ETH benchmarks using only 5000 points from each point cloud obtained with the farthest point sampling algorithm

	RR (%)	RRE (°)	RTE (cm)
3DMatch	83.39	1.66	6.35
KITTI	90.63	0.18	5.98
ETH	94.67	0.41	2.17

We use the stricter threshold $\tau_t = 60$cm to evaluate on the KITTI benchmark

Table 10 Average runtime in seconds for registering two pairs from the 3DMatch benchmark

	Mean	Std
GeDi	41.51	7.91
SC2-PCR	0.14	0.10
GeoTransformer	0.24	0.04
EGS	12.11	1.99

Comparison made by using only 5000 points from each point cloud

Table 11 Average runtime breakdown in seconds for a registration pair from the 3DMatch benchmark

			#	CPU / GPU	Cum. time
pre-processing	target point cloud		1	CPU	0.02 s
	source point cloud	rotate	N	CPU	0.24 s
		make positive	N	CPU	0.22 s
		voxelize	N	CPU	0.79 s
		pad	N	CPU	0.55 s
		move to gpu	N	both	1.07 s
cross-correlation			N	GPU	10.28 s
estimation			1	GPU	0.002 s
refinement			1	CPU	0.10 s

For each pipeline part we denote its number of repetitions under column #, if the part is executed on the CPU or GPU and the cumulative time of execution. For parts that are repeated N times, the cumulative time is the sum of its N repetitions. For the final version of the EGS method, $N = 3536$. Note that, in order to obtain these results, the GPU parallelization has been omitted

the EGS are not impaired. In the final version of the EGS, however, we still use all the points available, in order to eliminate information loss and the additional overhead of running the farthest point sampling algorithm.

As can be seen from Table 10, the learning methods are very fast. With the exception of GeDi, SC2-PCR and GeoTransformer operate in under a second. GeDi, on the other hand, runs for 41.51 seconds on average. The longer runtime comes from the computation of the local reference frame at each extracted patch. Our non-learning featureless EGS method, on the other hand, runs for 12.11 seconds on average with a deviation of 1.99 seconds. Compared to GeoTransformer and SC2-PCR, the EGS has a significantly higher runtime. However, as we show in the following discussion, these methods also require a lot more computing resources compared to the EGS, without having the benefit of improving the registration results. Therefore, the lower runtime might not justify the usage of such methods, since their applicability is impaired. We note here that Table 10 does not measure the time necessary to load the model onto the GPU or the time for obtaining the farthest point sampling indices which, differently from the EGS, are a necessity for the learning methods.

We break down the average runtime for each part of our pipeline in Table 11 and present the cumulative runtime results. In order to measure the cumulative runtimes, we omit using the GPU parallelization, obtaining, therefore, higher results than the presented 12.11 seconds. Nevertheless, these results can still provide an intuition into the various parts of the EGS method. To recap, the EGS pipeline is comprised of parts that are repeated N times (such as rotating the source point cloud or performing the cross-correlation) and parts that are repeated only once (such as estimating the final rotation and translation). Therefore, for the pipeline parts that are repeated N times, we sum up their N runtimes in order to obtain the cumulative running time.

As can be seen from Table 11, the largest portion of time is being used on the cross-correlation part. Even though this runtime seems large, we note that the average size of the voxelized point clouds from the 3DMatch dataset is $35.7 \times 28.5 \times 31.4$ for the voxel resolution of $VR = 7$cm. This, in addition to repeating the cross-correlation $N = 3536$ times for each precomputed rotation R_i, makes the 10.28 seconds more plausible. In order to accelerate the cross-correlation, we make use of the FFT algorithm. Using the cross-correlation without the FFT implementation in the background (like the one in PyTorch [117] for example) runs on average around 17 times slower than our method. The problem arises because these implementations are not built for having both the source and target volumes of such great size. The FFT, in turn, reduces the complexity from $O(K^6)$ to $O(K^3 \log(K))$ [118], where K indicates the size of the voxelized volumes. Since K can reach very high values (reaching even sizes of $203.65 \times 127.24 \times 15.08$ on average for the larger KITTI dataset), using the FFT algorithm greatly reduces the runtime.

Table 12 Average GPU occupancy in GBs for registering the 3DMatch benchmark using only 5000 points from each point cloud

	Average GB
GeDi	8.24
SC2-PCR	6.02
GeoTransformer	7.23
EGS	1.29

In contrast to the learning methods, the EGS does not pose a limitation on the number of points it can use during the registration process. To register, therefore, two point clouds from the 3DMatch dataset with 338000 points on average each, the EGS takes 27.80 seconds. The rise in running time compared to using only 5000 points, comes from the pre-processing of the source point cloud and the refinement parts. Since the point clouds are much bigger (67.6 times more points on average) and these parts are computed on the CPU, their runtime gets inflated. The runtimes of the remaining parts, namely the pre-processing of the target point cloud, the cross-correlation and the estimation parts, remain approximately the same, since the number of voxels do not drastically change. Therefore, to increase the efficiency of the EGS when using a great amount of points, the pre-processing step should be parallelized on the GPU.

To further improve our sub-optimal implementation we can make use of the hierarchical property of the rotation sampling. Since the polyhedron $\{3, 5+\}_{2,0}$ vertices are a subset of the polyhedron $\{3, 5+\}_{4,0}$ vertices, we can hierarchically query the rotations of polyhedron $\{3, 5+\}_{4,0}$ using only the candidate solutions of polyhedron $\{3, 5+\}_{2,0}$. Since the runtime of the EGS amounts to around 2 seconds when using $\{3, 5+\}_{2,0}$ as the sampling for $\mathbb{S}^2$, the total runtime would benefit from such a strategy. We leave the pre-processing parallelization, and the hierarchical implementations as future work.

We compute the average resources needed to run a method by comparing the average occupancy of the GPU memory necessary for registering a pair of point clouds. An increasing trend in recent works is to tackle the registration problem by adding more compute power, which in turn translates to a bigger learning model. Therefore, all of the recent methods have a very large memory footprint, which makes them inefficient for many practical use-cases.

As can be seen from Table 12, the most memory efficient learning method uses at least 6 GB of GPU RAM. In contrast, the EGS method uses only 1.29 GB, which makes it much more efficient and applicable in scenarios with limited resources, such as robots or mobile phones. Additionally,

the advantage of the EGS over the learning methods is that the parameters can be changed to address the needs of the user, without any need for re-training. For example, to lower the resources even more, the user can use a different rotation parametrization, as the one show in Table 8 under result #11, without any trade-offs to the results; where the GPU occupancy drops to only 0.89 GB. Therefore, our method provides a broader applicability compared to the learning methods, without any trade-offs to the quality of the results.

6 Implementation details

All the experiments are done on a single desktop computer, with an Intel Core i7-9700 CPU (3GHz, 8-core), 16GB RAM, and NVIDIA TITAN Xp, under Ubuntu 18.04 LTS. We use PyTorch [117] to implement the EGS and Open3D [119] to implement the generalized ICP algorithm. The cross-correlation is a wrap around of the PyTorch FFT implementation which is made to mimic a standard PyTorch 3D convolution (which by definition performs a cross-correlation). Therefore, we realize a fast cross-correlation that can handle two big voxelized volumes, which a *standard* Pytorch 3D convolution implementation could not do. The padding discussed in the paper, therefore, refers to the cross-correlation volume padding, and not to the FFT padding.

All the other results are reproduced using the respective authors code and data pre-processing guidelines. We provide the implementation details in Appendix B.

7 Visualizations

We visualize the registration results on the two most challenging datasets from the novel FAUST-partial benchmark: FP–R–H and FP–O–H. In Figs. 9 and 10, we show the worst registrations for each method on the respective datasets. More concretely, each row represents the worst performing registration pair for a given method based on the average distance (AD) between the points of the transformed source point clouds:

$$AD = \frac{1}{K} \sum_{\substack{i=1 \\ x_i \in \text{src}}}^{K} \|(R_{gt} x_i + t_{gt}) - (R_{\text{est}} x_i + t_{\text{est}})\|_2 \tag{17}$$

where K is the number of points, R_{gt} and t_{gt} represent the ground-truth rotation and translation, and R_{est} and t_{est} represent the estimated ones. Whereas the RRE and RTE metrics are useful to distinguish between the rotational and translational errors, the AD metric allows us to aggregate this information into one metric and discriminate the *worst* examples over a single metric.

The Figures reflect well the results already presented in Table 7. The percentage of successfully registered examples (the recall measure RR) from the Table follows the successfully registered worst case examples from the Figures.

8 Conclusion

The proposed traditional approach provides exceptionally strong 3D registration results. Even though the method is simple and featureless, it is still very effective, demonstrating great results on public benchmarks, and even achieving the best results on several of them, when compared to the generalization performance of the current state-of-the-art methods. Following a thorough analysis, we see that the EGS is robust to the change of its parameters and is not dependent on the refinement strategy choice. To further advance the analysis of a 3D registration method, we provide a methodology for creating better 3D registration benchmarks and assessing their difficulty. Using this methodology, we propose a novel FAUST-partial benchmark that addresses the lack of registration parameter range variability in the current benchmarks, as well as the bias toward similar data. The benchmark provides the option to isolate the analysis of the quality of a particular 3D registration method to a single registration parameter and determine its robustness regrading that parameter. Comparing the state of the art on the novel benchmark, we observe: a clear drop in performance for lower overlapping point clouds, almost no influence of the translation parameter to the difficulty of the registration, and some influence of the rotation range to the difficulty of the registration. The EGS baseline achieves competitive results and outperforms all the methods on the medium translation FP–T–M and medium overlap FP–O–M benchmarks.

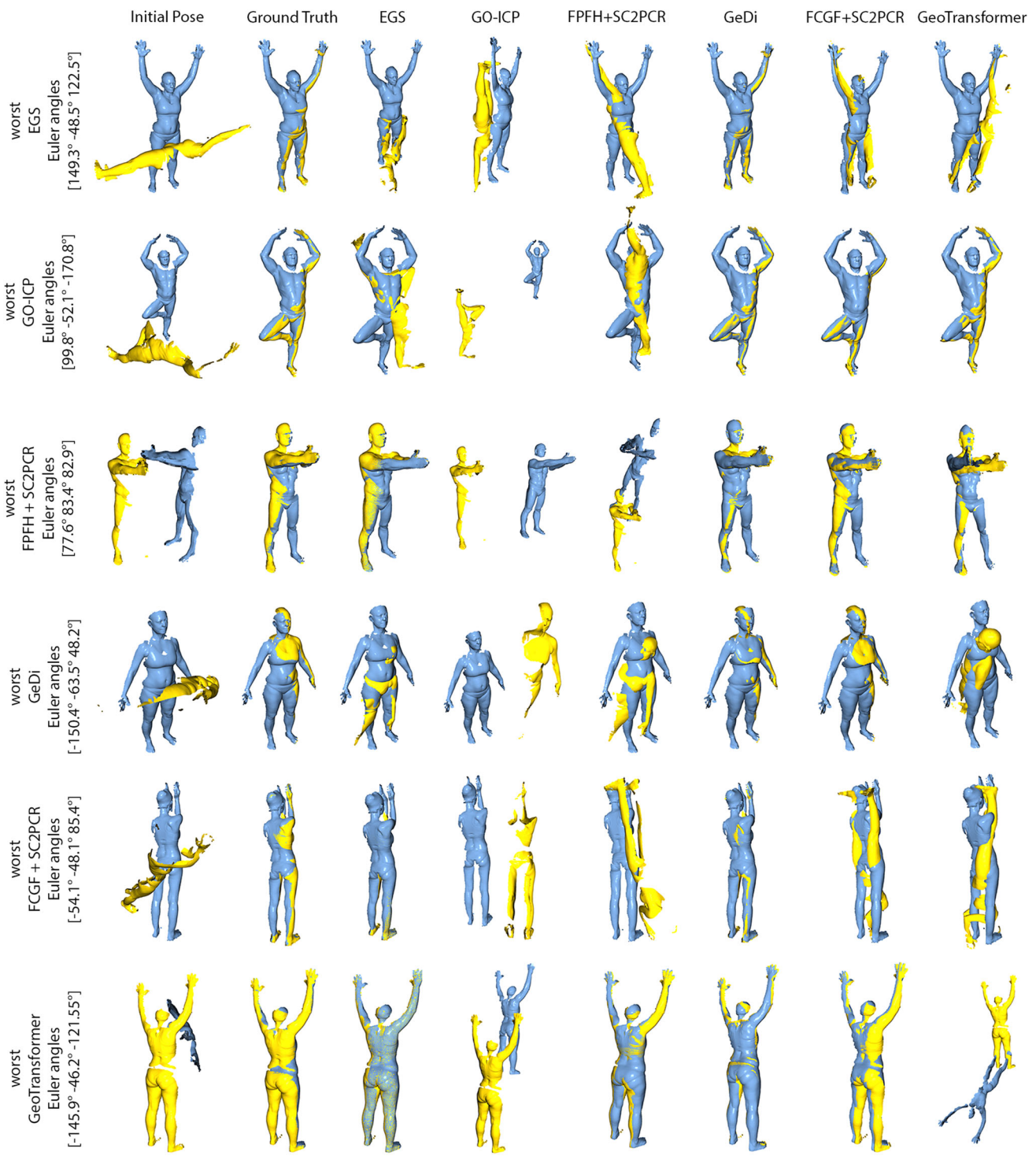

Fig. 9 Visualization of the worst registration results of the GO-ICP, FPFH+SC2PCR, GeDi, FCGF+SC2PCR, GeoTransformer and our proposed EGS method on the FP–R–H dataset. Each row represents the registration example that obtained the worst AD metric result for a particular method. Each column represents the registration result for a particular method. We additionally indicate the ground-truth rotation Euler angles in degrees for each example

Fig. 10 Visualization of the worst registration results of the GO-ICP, FPFH+SC2PCR, GeDi, FCGF+SC2PCR, GeoTransformer and our proposed EGS method on the FP–O–H dataset. Each row represents the registration example that obtained the worst AD metric result for a particular method. Each column represents the registration result for a particular method. We additionally indicate the ground-truth overlap for each example

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by David Bojanić and Kristijan Bartol. The first draft of the manuscript was written by David Bojanić, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work has been supported by the Croatian Science Foundation under the projects IP-2018-01-8118, DOK-2020-01 and IP-2019-04-9157, and has been partially funded by the EU under project iToBoS (SC1-BHC-06-2020-965221).

Availability of data and materials Not applicable.

Declarations

Conflict of interest The authors have no conflict of interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Code availability The implementation code is available at github.com/DavidBoja/exhaustive-grid-search.

Appendix A Error bounds derivation

In this section, we derive a numerical bound RB on the rotation error and an analytical bound TB on the translation error.

To create the numerical bound RB for the rotation error, we generate a set of $K = 1,000,000$ random rotation matrices $R^{\text{rand}} = \{R_k \mid k = 1, \dots, K\}$ using [120]. Next we find the upper bound for the minimal RRE between each generated matrix R_k and our precomputed set of N rotation matrices $R^{\text{precomp}} = \{R_i \mid i = 1, \dots, N\}$ from the main paper. We use R^{precomp} generated with the polyhedra $\{3, 5+\}_{4,0}$ and angle step $S = 10$. The numerical approximation for the upper rotation bound RB can therefore be derived as:

$$RB = \max_{R_k \in R^{\text{rand}}} \min_{R_i \in R^{\text{precomp}}} RRE(R_i, R_k) = 21.79^\circ \tag{A1}$$

The numerical bound amounts to 21.79 degrees.

The translation error depends on the voxelization resolution VR used. Let t_{GT} be the ground-truth translation vector for a registration example, and let t_* be the estimation computed by the EGS method. If the ground-truth translation t_{GT} is located in the discretized location represented with t_*, we can derive the following:

$$RTE(t_*, t_{\text{GT}}) = \|t_* - t_{\text{GT}}\|_2 \leq \frac{1}{2}\sqrt{3}VR \tag{A2}$$

where Eq. (A2) follows by knowing that the maximal distance from the central point of a voxel is half of the space diagonal of that voxel. For a voxel with side VR, the space diagonal amounts to $\sqrt{3}VR$. Therefore, the upper bound on the translation error is:

$$TB = \frac{\text{VR}\sqrt{3}}{2}. \tag{A3}$$

For a coarse voxel resolution of $VR = 7cm$, the upper translation error bound would amount to $TB = 6.06\text{cm}$.

Appendix B Implementation details

We list the parameters used for the various implemented registration methods.

To implement GO-ICP [33], we first translate each point cloud to the origin and use 5cm, 30cm, 10cm and 1cm voxel grid to downsample the points from the 3DMatch, KITTI, ETH and FP benchmarks, respectively. We tried to register the examples without downsampling as well, but the results did not differ. Next, we scale the point clouds of a registration pair using the $L2$ norm of the farthest point from both point clouds. We present the results using the default parameters noted in the paper: 0.001 mean squared error convergence threshold, $-\pi$ for the smallest rotation values along the x,y and z dimensions of the rotation cube, 2π radians for the side length of each dimension of the rotation cube, -0.5 for the smallest translation value along the x,y and z dimensions of the translation cube, 1.0 for the side length of each dimensions of the translation cube, no trimming, $300 \times 300 \times 300$ distance transforms (DT) with 2.0 expand factor. As noted by several works [33, 83], the method is sensitive to the trimming parameter that is correlated to the overlap parameter. Therefore, we tried out several trimming factors: 0%, 30% and 60%. The results do change for the different parameters.

To implement SC2-PCR [84], we first voxel-downsample the point clouds and then find the FPFH or FCGF features with the same parameters used by the authors. Refer to Table 13 for the parameters used.

To implement DIP [72] we randomly sample 25k points from each point cloud to compute their descriptors since

Table 13 Parameters used for the SC2-PCR method

Dataset	Number Iterations	Inlier Threshold	d_thre	Voxel Downsample	nms Radius
3DMatch	10	0.1	0.1	0.05	0.1
KITTI	20	0.6	0.1	0.3	0.6
ETH	10	0.3	0.2	0.1	0.3
FP	10	0.03	0.04	0.02	0.03

Table 14 Parameters used for the GeDi method

Dataset	Samples Per batch	Samples lrf	Radius lrf	Patches Per pair
3DMatch	500	4000	0.6	5000
KITTI	250	2000	2.5	50000
ETH	500	4000	1.5	5000
FP	500	4000	0.3	5000

Table 15 Parameters used for the PointDSC method

Dataset	Inlier Threshold	Sigma_d	Voxel Size
KITTI	0.6	1.2	0.30
ETH	0.3	1.2	0.20

results with 5k points were performing poorly. We tested different thresholds for the LRF radius, and present the results for the best performing one: $0.6 \times \sqrt{3}$cm. For the ETH dataset we use the pre-processed data from the authors and register the descriptors using RANSAC.

To implement GeDi [27], we use the parameters provided in Table 14. The remaining parameters: descriptor dimension, samples per patch and voxel size are kept fixed for all the datasets and are set to the values 32, 1024 and 0.01, respectively.

For SpinNet [71], we use the FCGF backbone to create descriptors for 5k keypoints chosen randomly.

To implement PointDSC [16], we use the parameters provided by the authors for the 3DMatch dataset and change several of them accordingly, as noted in Table 15.

To implement GeoTransformer [95], we use the provided parameters by the authors. However, to deal with the large memory footprint (noted by the authors themselves), we follow the authors recommendation to scale down and voxel downsample the point clouds so they either match the inlier threshold (0.1m) or point cloud density (0.006m) of 3DMatch. We choose the latter, since it provides better results. To match the point cloud density for a registration pair, we scale down the point cloud with:

$$s = \frac{\frac{0.006}{\text{src res}} + \frac{0.006}{\text{tgt res}}}{2} \tag{B4}$$

where *src res* and *tgt res* are the original source and target point cloud resolutions. Intuitively, we use the average scale needed to match the original resolutions of the source and target point clouds and the 3DMatch dataset. This scaling, however, is still too large to satisfy the large memory footprint. Therefore, we multiply the scaling factor with 2.3 for the KITTI dataset, 1.5 for the ETH dataset. We do not scale the FP datasets. After that, we voxel-downsample the point clouds using the same voxel size as used for the 3DMatch dataset: 0.025m.

To implement YOHO [73] we use the author guidelines to voxel-downsample the point clouds by multiplying the resolution of 0.025 with the scale difference between the evaluation dataset and the 3DMatch dataset. Hence, for the KITTI dataset the voxel-downsample resolution is 0.75cm and for ETH is 0.20cm.

References

1. Yang, J., Xiao, Y., Cao, Z.: Aligning 2.5d scene fragments with distinctive local geometric features and voting-based correspondences. IEEE Transactions on Circuits and Systems for Video Technology **29**(3), 714–729 (2019)
2. Tombari, F., Salti, S., Stefano, L.D.: Unique signatures of histograms for local surface description. Proc. ECCV **6313**, 356–369 (2010). https://doi.org/10.1007/978-3-642-15558-1_26
3. Li, Y., Dai, A., Guibas, L., Niessner, M.: Database-assisted object retrieval for real-time 3d reconstruction. Comput. Graph. Forum **34**, 435–446 (2015)
4. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3D-NDT. J. Field Robot. **24**, 803–827 (2007). https://doi.org/10.1002/rob.20204
5. Huang, X., Mei, G., Zhang, J., Abbas, R.: A comprehensive survey on point cloud registration. arXiv:2103.02690 (2021)
6. Bojanić, D., Bartol, K., Petković, T., Pribanić, T.: A review of rigid 3d registration methods. In: Proceedings of 13th International Scientific - Professional Symposium Textile Science & Economy (2020)
7. Qi, C., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85 (2017)
8. Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.: Kpconv: Flexible and deformable convolution for point clouds. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6410–6419 (2019). https://doi.org/10.1109/ICCV.2019.00651
9. Wang, Y., Solomon, J.M.: Deep closest point: learning representations for point cloud registration. In: 2019 IEEE/CVF

International Conference on Computer Vision (ICCV), pp. 3522–3531 (2019)
10. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. In: NeurIPS (2019)
11. Fu, K., Liu, S., Luo, X., Wang, M.: Robust point cloud registration framework based on deep graph matching. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8889–8898 (2021)
12. Cao, A.-Q., Puy, G., Boulch, A., Marlet, R.: PCAM: Product of cross-attention matrices for rigid registration of point clouds. In: International Conference on Computer Vision (ICCV) (2021)
13. Zeng, A., Song, S., Niessner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from RGB-D reconstructions. In: CVPR (2017)
14. Huang, S., Gojcic, Z., Usvyatsov, M.M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4265–4274 (2021)
15. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.-L.: D3feat: joint learning of dense detection and description of 3d local features. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6358–6366 (2020)
16. Bai, X., Luo, Z., Zhou, L., Chen, H., Zeyu Hu, L.L., Fu, H., Tai, C.-L.: PointDSC: robust point cloud registration using deep spatial consistency. In: CVPR (2021)
17. Lee, J., Kim, S., Cho, M., Park, J.: Deep hough voting for robust global registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
18. Wang, J., Yang, C., Wei, L., Chen, R.: Csce-net: channel-spatial contextual enhancement network for robust point cloud registration. Remote Sens. (2022). https://doi.org/10.3390/rs14225751
19. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: robust & efficient point cloud registration using pointnet. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7156–7165 (2019)
20. Li, X., Pontes, J.K., Lucey, S.: Pointnetlk revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12763–12772 (2021)
21. Xu, H., Liu, S., Wang, G., Liu, G., Zeng, B.: Omnet: learning overlapping mask for partial-to-partial point cloud registration. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3112–3121 (2021)
22. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074
23. Pomerleau, F., Liu, M., Colas, F., Siegwart, R.: Challenging data sets for point cloud registration algorithms. Int. J. Robot. Res. **31**, 1705–1711 (2012). https://doi.org/10.1177/0278364912458814
24. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: a deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, pp. 1912–1920 (2015). https://doi.org/10.1109/CVPR.2015.7298801
25. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: NIPS (2017)
26. Kawaguchi, K., Kaelbling, L.P., Bengio, Y.: Generalization in deep learning. arXiv:1710.05468 (2017)
27. Poiesi, F., Boscaini, D.: Learning general and distinctive 3d local deep descriptors for point cloud registration. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
28. Bojanić, D., Bartol, K., Forest, J., Gumhold, S., Petković, T., Pribanić, T.: Challenging the universal representation of deep models for 3d point cloud registration. In: BMVC 2022 Workshop Universal Representations for Computer Vision (2022)
29. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3D mesh registration. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway, NJ, USA (2014)
30. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**, 239–256 (1992)
31. Chen, Y., Medioni, G.G.: Object modelling by registration of multiple range images. Image Vis. Comput. **10**, 145–155 (1992)
32. Pavlov, A., Ovchinnikov, G., Derbyshev, D., Tsetserukou, D., Oseledets, I.: Aa-icp: Iterative closest point with anderson acceleration, pp. 1–6 (2018). https://doi.org/10.1109/ICRA.2018.8461063
33. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-icp: a globally optimal solution to 3d icp point-set registration. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2241–2254 (2016). https://doi.org/10.1109/TPAMI.2015.2513405
34. Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P.: The trimmed iterative closest point algorithm. In: 2002 International Conference on Pattern Recognition, vol. 3, pp. 545–5483 (2002). https://doi.org/10.1109/ICPR.2002.1047997
35. Yue, P., Bisheng, Y., Fuxun, L., Zhen, D.: Iterative global similarity points: a robust coarse-to-fine integration solution for pairwise 3d point cloud registration. In: 2018 International Conference on 3D Vision (3DV) (2018)
36. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graph. **27**(3), 1–10 (2008). https://doi.org/10.1145/1360612.1360684
37. Huang, J., Kwok, T.-H., Zhou, C.: V4PCS: volumetric 4PCS Algorithm for Global Registration. J. Mech. Des. (2017). https://doi.org/10.1115/1.4037477.111403
38. Mohamad, M., Rappaport, D., Greenspan, M.: Generalized 4-points congruent sets for 3d registration. In: 2014 2nd International Conference on 3D Vision, vol. 1, pp. 83–90 (2014). https://doi.org/10.1109/3DV.2014.21
39. Pribanić, T., Petković, T., Donlić, M.: 3d registration based on the direction sensor measurements. Pattern Recognit. **88**, 532–546 (2019). https://doi.org/10.1016/j.patcog.2018.12.008
40. Huang, Y., Da, F.: Registration algorithm for point cloud based on normalized cross-correlation. IEEE Access **7**, 137136–137146 (2019). https://doi.org/10.1109/ACCESS.2019.2942127
41. Liu, M., Li, L.: Cross-correlation based binary image registration for 3D palmprint recognition. In: 2012 IEEE 11th International Conference on Signal Processing, vol. 3, pp. 1597–1600 (2012). https://doi.org/10.1109/ICoSP.2012.6491885
42. Liu, S., Yang, B., Wang, Y., Tian, J., Yin, L., Zheng, W.: 2d/3d multimode medical image registration based on normalized cross-correlation. Appl. Sci. (2022) https://doi.org/10.3390/app12062828
43. Wang, C., Jing, X., Zhao, C.: Local upsampling fourier transform for accurate 2d/3d image registration. Comput. Electr. Eng. **38**(5), 1346–1357 (2012). https://doi.org/10.1016/j.compeleceng.2012.04.005
44. Lucchese, L., Doretto, G., Cortelazzo, G.M.: A frequency domain technique for range data registration. IEEE Trans. Pattern Anal. Mach. Intell. **24**(11), 1468–1484 (2002). https://doi.org/10.1109/TPAMI.2002.1046160
45. Keller, Y., Shkolnisky, Y., Averbuch, A.: Volume registration using the 3-d pseudopolar fourier transform. IEEE Trans. Signal Process. **54**(11), 4323–4331 (2006). https://doi.org/10.1109/TSP.2006.881217
46. Curtis, P., Payeur, P.: A frequency domain approach to registration estimation in three-dimensional space. IEEE Trans. Instrum. Meas. **57**(1), 110–120 (2008). https://doi.org/10.1109/TIM.2007.909499

47. Bülow, H., Birk, A.: Spectral 6dof registration of noisy 3d range data with partial overlap. IEEE Trans. Pattern Anal. Mach. Intell. **35**(4), 954–969 (2013). https://doi.org/10.1109/TPAMI.2012.173
48. Bülow, H., Birk, A.: Scale-free registrations in 3d: 7 degrees of freedom with fourier mellin SOFT transforms. Int. J. Comput. Vis. **126**(7), 731–750 (2018). https://doi.org/10.1007/s11263-018-1067-5
49. Tong, X., Ye, Z., Xu, Y., Gao, S., Xie, H., Du, Q., Liu, S., Xu, X., Liu, S., Luan, K., Stilla, U.: Image registration with fourier-based image correlation: a comprehensive review of developments and applications. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 12, 4062–4081 (2019) https://doi.org/10.1109/JSTARS.2019.2937690
50. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999). IEEE
51. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Proceedings of the Symposium on Geometry Processing. SGP '09, pp. 1383–1392. Eurographics Association, Goslar, DEU (2009)
52. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. Vis. Comput. **27**, 963–976 (2011) https://doi.org/10.1007/s00371-011-0610-y
53. Salti, S., Tombari, F., Spezialetti, R., Stefano, L.D.: Learning a descriptor-specific 3d keypoint detector. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2318–2326 (2015). https://doi.org/10.1109/ICCV.2015.267
54. Steder, B., Rusu, R.B., KurtKonolige, Burgard, W.: NARF: 3D Range Image Features for Object Recognition
55. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217 (2009). https://doi.org/10.1109/ROBOT.2009.5152473
56. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3d data description. In: Proceedings of the ACM Workshop on 3D Object Retrieval. 3DOR '10, pp. 57–62. Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1877808.1877821
57. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pp. 689–696 (2009). https://doi.org/10.1109/ICCVW.2009.5457637
58. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Anal. Mach. Intell. **21**(5), 433–449 (1999). https://doi.org/10.1109/34.765655
59. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors **3**, 224–237 (2004). https://doi.org/10.1007/978-3-540-24672-5_18
60. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III. ECCV'10, pp. 356–369. Springer, Berlin, Heidelberg (2010)
61. Guo, Y., Bennamoun, M., Sohel, F.A., Wan, J., Lu, M.: 3d free form object recognition using rotational projection statistics. In: 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 1–8 (2013). https://doi.org/10.1109/WACV.2013.6474992
62. Theiler, P.W., Wegner, J.D., Schindler, K.: Keypoint-based 4-points congruent sets - automated marker-less registration of laser scans. ISPRS J. Photogram. Remote Sens. **96**, 149–163 (2014). https://doi.org/10.1016/j.isprsjprs.2014.06.015
63. Zhou, Q.-Y., Park, J., Koltun, V.: Fast global registration. In: ECCV **2016**, 766–782 (2016)
64. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3d: Generic features for video analysis. arXiv:1412.0767 (2014)
65. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013). https://doi.org/10.1109/TPAMI.2012.59
66. Li, J., Lee, G.H.: Usip: Unsupervised stable interest point detection from 3d point clouds. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 361–370 (2019)D
67. Yew, Z.J., Lee, G.H.: 3dfeat-net: Weakly supervised local 3d features for point cloud registration. arXiv:1807.09413 (2018)
68. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8957–8965 (2019). https://doi.org/10.1109/ICCV.2019.00905
69. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching (2018). https://doi.org/10.1109/CVPR.2018.00028
70. Khoury, M., Zhou, Q.-Y., Koltun, V.: Learning compact geometric features. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 153–161 (2017)
71. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
72. Poiesi, F., Boscaini, D.: Distinctive 3D local deep descriptors. In: IEEE Proceedings of of Int'l Conference on Pattern Recognition, Milan, Italy (2021)
73. Wang, H., Liu, Y., Dong, Z., Wang, W., Yang, B.: You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. ACM Multimedia 2022 (2022)
74. Li, L., Zhu, S., Fu, H., Tan, P., Tai, C.-L.: End-to-end learning local multi-view descriptors for 3d point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
75. Huang, X., Qu, W., Zuo, Y., Fang, Y., Zhao, X.: Imfnet: interpretable multimodal fusion for point cloud registration. IEEE Robot. Autom. Lett. **7**(4), 12323–12330 (2022). https://doi.org/10.1109/LRA.2022.3214789
76. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 539–5461 (2005). https://doi.org/10.1109/CVPR.2005.202
77. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
78. Barath, D., Matas, J.: Graph-cut RANSAC. In: Conference on Computer Vision and Pattern Recognition (2018)
79. Quan, S., Yang, J.: Compatibility-guided sampling consensus for 3-d point cloud registration. IEEE Trans. Geosci. Remote Sens. **58**(10), 7380–7392 (2020). https://doi.org/10.1109/TGRS.2020.2982221
80. Pais, G.D., Miraldo, P., Ramalingam, S., Nascimento, J.C., Govindu, V.M., Chellappa, R.: 3DRegNet: a deep neural network for 3D point registration, 7193–7203 (2019)
81. Chen, W., Li, H., Nie, Q., Liu, Y.-H.: Deterministic point cloud registration via novel transformation decomposition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6338–6346 (2022). https://doi.org/10.1109/CVPR52688.2022.00624
82. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume

1, vol. 2, pp. 1482–14892 (2005). https://doi.org/10.1109/ICCV.2005.20
83. Yang, H., Shi, J., Carlone, L.: Teaser: Fast and certifiable point cloud registration. IEEE Trans. Robot. **37**(2), 314–333 (2021). https://doi.org/10.1109/TRO.2020.3033695
84. Chen, Z., Sun, K., Yang, F., Tao, W.: Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13221–13231 (2022)
85. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: CVPR (2020)
86. Zhang, Z., Sun, J., Dai, Y., Fan, B., He, M.: Vrnet: learning the rectified virtual corresponding points for 3d point cloud registration. IEEE Trans. Circuits Syst. Video Technol. **32**, 4997–5010 (2022)
87. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. (TOG) (2019)
88. Yew, Z.J., Lee, G.H.: RPM-Net: robust point matching using learned features. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11821–11830 (2020)
89. Simon, M., Fischer, K., Milz, S., Witt, C., Groß, H.-M.: Stickypillars: robust feature matching on point clouds using graph neural networks. arXiv:2002.03983 (2020)
90. Li, J., Zhang, C., Xu, Z., Zhou, H., Zhang, C.: Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In: European Conference on Computer Vision (ECCV) (2020)
91. Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., Kautz, J.: DeepGMR: learning latent gaussian mixture models for registration. In: ECCV (2020)
92. Huang, X., Mei, G., Zhang, J.: Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11363–11371 (2020)
93. Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R.A., Lucey, S., Choset, H.: PCRNet: point cloud registration network using pointnet encoding. arXiv:1908.07906 (2019)
94. Yew, Z.J., Lee, G.h.: Regtr: End-to-end point cloud correspondences with transformers. In: CVPR (2022)
95. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. arXiv:2202.06688 (2022)
96. Xu, J., Huang, Y., Wan, Z., Wei, J.: Glorn: Strong generalization fully convolutional network for low-overlap point cloud registration. IEEE Trans. Geosci. Remote Sens. 60, 1–14 (2022) https://doi.org/10.1109/TGRS.2022.3208380
97. Li, Y., Harada, T.: Lepard: Learning partial point cloud matching in rigid and deformable scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
98. Huang, X., Mei, G., Zhang, J., Abbas, R.: A comprehensive survey on point cloud registration. arXiv:2103.02690 (2021)
99. Zhang, Z., Dai, Y., Sun, J.: Deep learning based point cloud registration: an overview. Virtual Real. Intell. Hardw. **2**(3), 222–246 (2020). https://doi.org/10.1016/j.vrih.2020.05.002
100. Xie, D., Zhu, W., Rong, F., Xia, X., Shang, H.: Registration of point clouds: A survey. In: 2021 International Conference on Networking Systems of AI (INSAI), pp. 136–142 (2021). https://doi.org/10.1109/INSAI54028.2021.00034
101. Marek, J., Chmelař, P.: Survey of point cloud registration methods and new statistical approach. Mathematics **11**(16) (2023) https://doi.org/10.3390/math11163564
102. Xu, N., Qin, R., Song, S.: Point cloud registration for lidar and photogrammetric data: A critical synthesis and performance analysis on classic and deep learning algorithms. ISPRS Open Journal of Photogrammetry and Remote Sensing 8, 100032 (2023) https://doi.org/10.1016/j.ophoto.2023.100032
103. Pugh, A.: Polyhedra: A Visual Approach. University of California Press, Oakland, CA (1976)
104. Brigham, E.O., Morrow, R.E.: The fast fourier transform. IEEE Spectr. **4**(12), 63–70 (1967). https://doi.org/10.1109/MSPEC.1967.5217220
105. Rao, K.R., Kim, D.N., Hwang, J.-J.: Fast Fourier Transform - Algorithms and Applications, 1st edn. Springer, Dordrecht, the Netherlands (2010)
106. Segal, A., Hähnel, D., Thrun, S.: Generalized-icp. (2009). https://doi.org/10.15607/RSS.2009.V.021
107. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets, vol. 26 (2007). https://doi.org/10.1145/1275808.1276407
108. Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A.: The perfect match: 3d point cloud matching with smoothed densities. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5540–5549 (2019)
109. Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. Ann. Math. Stat. **27**(3), 832–837 (1956). https://doi.org/10.1214/aoms/1177728190
110. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**(3), 1065–1076 (1962). https://doi.org/10.1214/aoms/1177704472
111. Mellado, N., Aiger, D., Mitra, N.J.: Super 4pcs fast global pointcloud registration via smart indexing. Comput. Graph. Forum **33**(5), 205–215 (2014). https://doi.org/10.1111/cgf.12446
112. Taati, B., Ioannou, Y., Harrap, R., Greenspan, M.: Difference of normals as a multi-scale operator in unorganized point clouds. In: 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 501–508. IEEE Computer Society, Los Alamitos, CA, USA (2012). https://doi.org/10.1109/3DIMPVT.2012.12
113. Yershova, A., Jain, S., LaValle, S.M., Mitchell, J.C.: Generating uniform incremental grids on SO(3) using the hopf fibration. Int. J. Robot. Res. **29**(7), 801–812 (2009). https://doi.org/10.1177/0278364909352700
114. Alexa, M.: Super-fibonacci spirals: Fast, low-discrepancy sampling of so(3). In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8291–8300 (2022)
115. Murphy, K.A., Esteves, C., Jampani, V., Ramalingam, S., Makadia, A.: Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. In: Proceedings of the 38th International Conference on Machine Learning, pp. 7882–7893 (2021)
116. Gorski, K.M., Hivon, E., Banday, A.J., Wandelt, B.D., Hansen, F.K., Reinecke, M., Bartelmann, M.: HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. Astrophys J. **622**(2), 759–771 (2005). https://doi.org/10.1086/427976
117. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates Inc, New York (2019)
118. Frigo, M., Johnson, S.G.: The design and implementation of fftw3. Proc. IEEE **93**(2), 216–231 (2005)
119. Zhou, Q.-Y., Park, J., Koltun, V.: Open3D: A Modern Library for 3D Data Processing (2018). arXiv:1801.09847

120. Stewart, G.W.: The efficient generation of random orthogonal matrices with an application to condition estimators. SIAM J. Numer. Anal. **17**(3), 403–409 (1980)

David Bojanić received his MSc degree in mathematics from the University of Zagreb, Croatia, in 2019. He is currently pursuing a PhD degree in computer science with the Faculty of Electrical Engineering and Computing. During his PhD, he has been a Visiting Researcher at the INRIA Grenoble Rhône-Alpes center in France and the TU Dresden faculty in Germany. His current research interest is shape estimation from 3D human body scans, focusing on removing constraints from 3D body measurement estimation methods. He is a Graduate Student Member of IEEE.

Kristijan Bartol received a BSc, MSc, and PhD degrees, all in computer science at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia (2016, 2019, and 2023, respectively). He is currently a postdoctoral research associate at the Technical University of Dresden, Germany. His research interests include human pose and shape estimation, accurate virtual garment tailoring, computer vision, and deep learning.

Josep Forest received his BSc in Industrial Informatics from the University of Girona in 1992, MSc in Electronics engineering from the Autonomous University of Barcelona in 1998, and PhD in 2004 from the University of Girona. His research interests are focused on 3D-machine vision, including laser triangulation, calibration, detection, and point cloud processing. His research also includes the usability of 3D applied to the industry for dimensional testing and quality control applications. He is former co-founder of AQSENSE (now part of COGNEX) and OPSIS Vision Technologies Spin-Off companies.

Tomislav Petković is Associate Professor at the University of Zagreb Faculty of Electrical Engineering and Computing. He received the engineer's degree, the magister degree, and PhD in electrical engineering all from the University of Zagreb, in 2002, 2006, and 2010, respectively. His main fields of research interest are digital image processing and analysis, 3D imaging, and computational imaging. He teaches several graduate courses in the field of digital signal and image processing. He is a member of IEEE and ACM.

Tomislav Pribanić was a Visiting Researcher with the INRIA Grenoble Rhône-Alpes, Grenoble, France, and the Fraunhofer IGD, Darmstadt, Germany. He was a Fulbright Visiting Scholar with the University of Wisconsin, Madison, USA. He is currently a Professor with the Faculty of Electrical Engineering and Computing, University of Zagreb. He teaches several undergraduate and graduate courses in the field of algorithms and data structures, image processing, sensors, and human motion analysis. He has led a number of domestic and international scientific projects, collaborating with researchers from within and outside the EU. His main research interests include computer vision, machine learning, and biomedical signal measurement and analysis. The results of his research have been implemented in technological projects and received recognition for innovations. He is a Senior Member of IEEE and a Collaborating Member of the Croatian Academy of Engineering.