Evaluation of 3D Registration Deep Learning Methods Using Iterative Transformation Estimations

David BOJANIĆ*[1], Kristijan BARTOL[1], Tomislav PETKOVIĆ[1],
Nicola D'APUZZO[2], Tomislav PRIBANIĆ [1]
[1] University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia;
[2] Hometrica Consulting, Ascona, Switzerland

https://doi.org/10.15221/20.31

Abstract

3D registration is a process of aligning multiple three-dimensional (3D) data structures (such as point clouds or meshes) and merging them into one consistent and seamless 3D data structure. With the scope of 3D reconstruction, 3D human body scans from multiple views need to be registered into a single point cloud to create a seamless 3D representation.

Following current state-of-the-art deep learning approaches [1,2,3], we argue that an encoder-decoder approach, where the decoder part of the architecture uses a recursive layer that iteratively estimates the rigid transformation, should provide the best results. We adapt an approach created for the task of 3D segmentation called `RSNets` [4] to the task of 3D registration and compare it to the current state-of-the-art algorithm `PCRNet` [3].

Keywords: 3d computer vision, 3d registration, deep learning

1. Introduction

3D registration is a fundamental problem in computer and robot vision. Given two 3D structures (usually represented as a set of points) in different coordinate systems, or equivalently in the same coordinate system with different poses, the goal is to find the transformation that best aligns one structure to the other. It arises as a subtask in many different vision applications such as: 3D reconstruction [5,6,7], object recognition and categorization [8,9,10], shape retrieval [11], robot navigation [12,13] and data fusion obtained from different sensors [14]. Figure 1. shows two 3D human point clouds obtained from two different viewpoints of the same object that, when registered, merge into one seamless and coherent object.

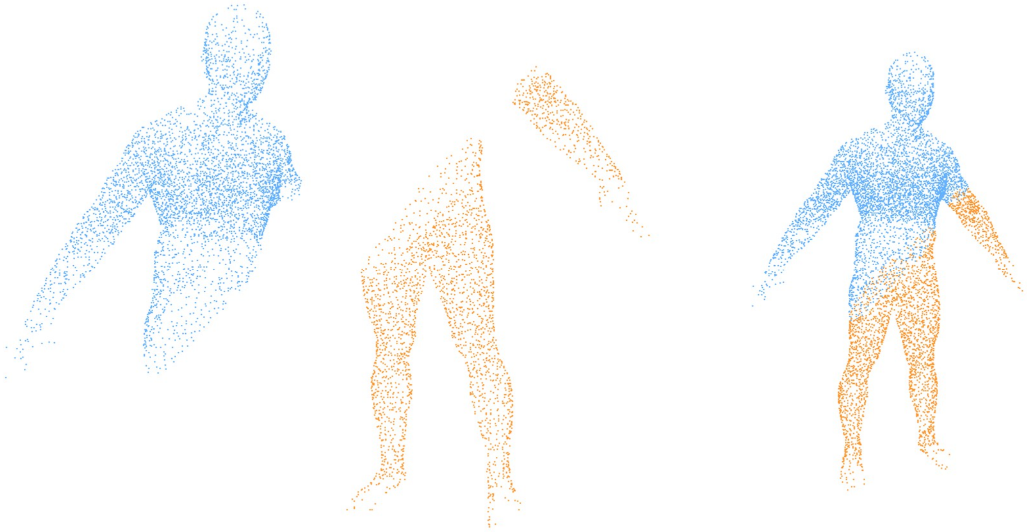

Fig 1. Example of partial registration of two point clouds.

Even though some of these problems might be solved using hardware solutions [15] such as calibrated mechanics (e.g. movable robot arms) aware of their positional displacement, the applicability of such solutions is poor. Furthermore, problems such as object recognition, still require software solutions, thus making 3D registration a prominent research field.

Whereas this paper focuses on rigid registration, where we assume a fixed rigid environment, there are approaches [16] that tackle the more general non-rigid registration problem in which articulated objects and soft bodies that can change shape over time might be present.

2. 3D Transformations

A rigid 3D rigid-body transformation can be represented in several ways. The core elements of the transformation are a rotational component R and a translational component t which, obviously, rotate and translate the 3D object in hand. The rotational component R is a 3x3 matrix from the special orthogonal group SO(3), also called the rotation group, which contains all 3x3 orthogonal matrices having determinant equal to 1. The orthogonality condition is necessary because the rotation connects two coordinate systems while the unit determinant condition follows from the orthogonality condition and preservation of the "handedness" of the coordinate system. More intuitively, the rotation matrix R can be further divided into three matrices representing the rotation around each of the three axes x, y and z by the angles α, β and γ in the following way:

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \qquad (1)$$

where:

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \quad (2)$$

This representation indicates that there are only 3 degrees of freedom (DOF) when determining a rotation matrix as opposed to 9 when looking at the matrix as an element of the SO(3) group.
The translational component t is a 3x1 vector from R^3 and has 3 DOF as well. Consequently, in combination with a rotation, the rigid transformation has 6 DOF.
If $p = [x, y, z]^T$ is a point in space, then a rigid transformation can be written in matrix form as:

$$p' = R \cdot p + t \qquad (3)$$

where · represents matrix multiplication and $p' = [x', y', z']^T$is the transformed point.

We can combine the rotation and translation matrices to form a more compact representation of the rigid transformation using homogeneous coordinates. Homogeneous coordinates are usually used in projective geometry and offer a simplified way of combining transformations using only matrix multiplications. They extend 3D points $[x, y, z]^T$ with equivalence classes $[kx, ky, kz, k]^T$ which represent the same point for any $k \epsilon R \setminus 0$. Now, the transformation is as a 4x4 matrix from the special Euclidean group SE(3) with the form:

$$T = \begin{bmatrix} & R & & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

where again $R \, \epsilon \, SO(3)$ and $t \epsilon R^3$.
If $p = [x, y, z, 1]^T$$ is a homogeneous representation of a point in space, the rigid transformation now takes form:

$$p' = T \cdot p \qquad (5)$$

where again · is the matrix multiplication and p' is the new transformed point in homogeneous coordinates. Both SO(3) and SE(3) are Lie groups with their appropriate Lie algebras so(3) and se(3). An exponential map connects these two structures and allows us to represent a transformation matrix $T \, \epsilon \, SE(3)$ as an element of the se(3) algebra as so:

$$T = exp(\textstyle\sum_i \varepsilon_i T_i) \quad (6)$$

where T_i are the generators of the exponential map with twist parameters $\varepsilon \, \epsilon \, R^6$. Now, the rigid transformation takes the same form as equation (5).
There exist other transformation representations, such as quaternions, but are rarely used in the context of 3D registration and are hence not relevant to our discussion.

3. Problem Formulation

As we've seen, the registration problem comes down to finding the rotation R and translation t matrices that best align the two point clouds. The problem can be approached by defining a cost function that represents the current error, and indicates how well the two point clouds overlap. This cost function is then minimized using common optimization techniques. The most common cost function is the L2 norm of the point cloud displacements.

Let $X = \{x_i\}_{i=1}^{N}$ be the source and $Y = \{y_i\}_{j=1}^{M}$ the target point clouds that need to be registered. Usually this terminology indicates that we are searching for a transformation of the target point cloud Y that registers it to the source X.

Let

$$C = \{(x_i, y_j) | \, x_i \in X, y_j \in Y \; holding \; \forall \, y_k \in Y \; d(x_i, y_j) < d(x_i, y_k) \; and \; d(x_i, y_j) < thr\} \quad (7)$$

be a set of correspondences between points from X and Y where $d(\cdot,\cdot)$ is the Euclidean distance and thr is a threshold that discards distances larger than it, as to omit larger errors when dealing with partial registration. As opposed to full registration, where all the points from the source point cloud have a match in the target point cloud, partial matching assumes only some points are correspondent (as is the more typical case). Figure 1 shows one such example. Then, the registration problem can be written as a minimization problem:

$$\min_{R,t} \sum_{(x_i, y_j) \in C} \left\| R \cdot x_i + t - y_j \right\|_2^2 \qquad (8)$$

Here, the set C was determined as the points from both clouds that have the smallest distance to one another which is a technique usually used in fine matching rather than coarse matching. More generally, the set C can be determined in many alternative ways as we'll see in later chapters. In practice, the correspondences are unknown which makes (8) a classic "chicken-and-egg" problem: if the correspondences are known, R and t can be easily found; if R and t are known, the correspondences are easily derived.

To conquer this, some methods interchangeably search for the correspondences and transformation. Most of them, however, focus on finding reliable correspondences after which the transformation is derived.

If C is the set of correspondences, (8) has a closed form solution:

$$R = VU^T, \; t = -R\bar{x} + \bar{y} \qquad (9)$$

where U and V are obtained using the singular value decomposition (SVD) $H = USV^T$ of the covariance matrix

$$H = \sum_{(x_i, y_j) \in C} (x_i - \bar{x})(y_j - \bar{y})^T \qquad (10)$$

and centroids

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \,, \; \bar{y} = \frac{1}{M} \sum_{j=1}^{M} y_j \qquad (11)$$

More intuitively, the process is similar to the principal component analysis (PCA) and extracts the major directions of shared change from the origin centered point clouds.

4. Related work

As said before, there are two approaches towards solving (8). One is to first determine the correspondences and use them along with (9). We denominate this approach as *detection-description-matching* as the three major components of the pipeline. The second method is to try and solve directly for R and t in various methods. We denominate this approach as *all-in-one* since they cannot be broken down into the detection-description-matching pipeline.

4.1 Detection-description-matching approaches

In this approach, a detection step is firstly used to reduce the number of points being considered in the registration process. It consists of detecting a certain number of key points that are prominent according to a specific criterion. The sizes of the input data make the detection step necessary in many approaches to obtain computationally manageable datasets.

Following evaluation papers and algorithm comparisons [18,19,20,21,22], some of the most promising hand-crafted detectors are: ISS [23], MeshDoG [24], Harris 3D [25] and HKS [26] whilst most promising learned detectors are described in: Salti et al. [20], Lin et al. [22], Suwajanakorn et al. [27].

The second step of the pipeline, description, consists of assigning values to the detected key points according to the local shape around them. They can be classified by the fact if they are based on a local reference frame (LRF) or just the local geometry. LRF is an independent 3D coordinate system from the world coordinate system that is established on the local surface. The goal of a LRF is mainly to make the feature description invariant to rigid transformation. LRF based methods have generally surpassed LRF-independent ones on most publicly available datasets [28]. Some LRF based methods are SHOT [8], RoPS [9] and LoVS [29]. Deep learning based descriptors, such as: PointNet [30], 3DMatch [31], CGF [32], PPFNet [33], PPF-FoldNet [34] and 3DFeat-Net [35], have surpassed many hand-crafted local geometric descriptors. Nevertheless, the majority of learned descriptors from raw data still suffer from sensitivity to rotation.

Finally, searching strategies are used to find correspondences between points in the two point sets. Descriptor values are used to prioritize the best apparent correspondences and a minimum of three are needed to determine the coarse alignment in 3D. Rather than using brute-force methods which are computationally expensive and yield a cost of $O(n^6)$ (triplets from both clouds need to be checked), we strive for more elaborate algorithms in order to report results in a reasonable amount of time. This is where searching (matching) algorithms take over.

Following [36],[37], the state-of-the-art methods for finding correspondences are geometric consistency (GC) [38], 3D Hough Voting (3DHV) [39] and game theory matching (GTM) [40].

After achieving a coarse alignment, a refinement step is applied. This step usually consists of using iterative methods to align the shapes as accurately as possible.

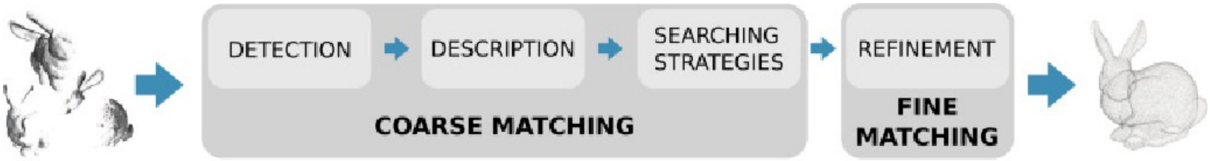

Fig 2. Pipeline of the detection-description-matching approach. Taken from [21].

4.2. All-in-one approaches

There are several methods [41] that cannot be separated into the detection-description-matching pipeline as the ones before.

4-point congruent sets (4PCS) [42] is a global registration algorithm. The global optimality references the finding of the global minima when solving (8). The method is based on efficiently finding the set of congruent 4-point bases in the source point cloud X, to a 4-point base selected from the target point cloud Y.

A set of 4 coplanar points is selected from X, S = { a,b,c,d }, not all collinear, such that ab intersects cd at the intermediate point e. Given a 4-point base constructed from two intersecting pairs, the ratios:

$$r_1 = \|a - e\| / \|a - b\|, \; r_2 = \|c - e\| / \|c - d\| \quad (12)$$

are preserved under affine transformations and therefore act as invariants to constrain the search for congruent 4-point bases in X. Generalized 4PCS [43] generalizes 4PCS by allowing the pairs to fall on two different planes which have an arbitrary distance between them. This separation exponentially decreases the search space of matching bases. 4PCS presents two bottlenecks: the extractions of congruent pairs from X and the verification of the large number of reported congruent sets.

By addressing these bottlenecks, Super 4PCS [44] improves the total runtime complexity to $O(n+k_1+k_2)$ where k_1 is the number of pairs of a given distance, and k_2 is the number of congruent bases. Lastly, Super Generalized 4PCS [45] combines the two solutions.

PointNetLK [46] utilizes the classical Lucas & Kanade (LK) algorithm [47] and tries to find the transformation in se(3) space between the PoinNet embeddings of the source and target point clouds. With the inverse compositional formulation of the problem, linearization, and approximation of the Jacobian matrix of the linearization process with finite differences (that only need to be computed once), PointNetLK iteratively updates the transformation matrix that it searches for. This process exhibits great efficiency since the only calculation after the first iteration is the difference of the embedded point clouds.

DeepICP [48] is an end-to-end learning-based point cloud registration framework. The algorithm firstly extracts feature descriptors from both the point clouds using PointNet++ [49]. After that, a point weighting layer is executed to learn the saliency of each point with, ideally, assigning higher weights to points with invariant and distinct features. The most significant K points are selected as the keypoints. Next a deep feature embedding (DFE) layer is applied to learn even more detailed keypoint descriptions. After that, a corresponding point generation (CGP) layer is applied to generate correspondences from the extracted features. The alignment is generated from (9).

Admittedly, the algorithm resembles a more complex approach from the *detection-description-matching* pipeline since the layers can be observed as detection, description and matching layers. Nevertheless, since the method offers an end-to-end process, it makes more sense to describe it as such, and not split the different parts in different sections.

Deep Closest Point [1] (DCP) is another end-to-end deep learning framework that could potentially be classified as a *detection-description-matching* approach. The algorithm firstly embeds the point clouds using PointNet or DGCNN [50]. Next, they encode contextual information using an attention-based module that modifies the embeddings taking into consideration all of the information gathered from the source and target point clouds. The correspondences are generated using a softmax function over the matrix product of the point cloud embeddings.

Iterative Matching Point [51] (IMA) is a very similar approach to DCP, with the biggest difference being that it wraps the whole algorithm in an iterative process. Every iteration then inputs the newly updated transformed point clouds which allows for more refined transformation results.

PRNet [2] is a sequential decision-making framework designed to solve a broad class of registration problems. As in DCP, the network embeds points using PointNet or DGCNN after which it selects keypoints as the ones with the greatest L2 norms. The correspondence is generated using a combination of the closest point in the other point cloud and the softmax solution from DCP. The closest point offers a sharp keypoint matching at the cost of non-differentiability, whereas softmax offers a "blurred" keypoint matching at the cost of the matches not being resolute. Hence, they use a Gumbel-Softmax [52] approach to sample a matching matrix with the use of an added "blurring" parameter. PRNet is designed to be iterative, and the process above is repeated multiple times using the newly transformed point cloud with the approximation of the alignment.

PCRNet [3] uses PointNet in a Siamese architecture to encode the shape information of a source and target point clouds into feature vectors. Next it concatenates those representations and uses a fully connected layer to estimate the transformation matrix. The whole approach is wrapped in an iterative component that in each iteration tries to align the target to the newly aligned source point cloud.

Yang et al. propose GO-ICP [53] that parametrizes the rotation by using a solid radius-π ball in R^3 with the angle-axis representation. It parametrizes the translation with a bounded cube $[-\varepsilon, \varepsilon]^3$. The algorithm uses the branch-and-bound (BnB) method to repeatedly search the space of SE(3). Whenever a better solution is found, it calls the ICP algorithm initialized at this solution to refine the objective function value. Next, it uses the ICP result as an updated upper bound and continues the BnB search. The procedure is repeated until convergence.

With the lack of a recent 3D registration evaluation in literature, we follow the comparisons of each individual method [1,2,3,42,43,44,45,46,51,53] and observe that approaches with an iterative component, like PRNet and PCRNet present the best performances to date. We hypothesize the reason behind such results is that these approaches offer enough "time" or iterations to correct the transformation matrix instead of trying to predict it at once. Multiple evidence that back up the hypothesis can be found in literature. Iterative matching point generalizes deep closest point with an iterative component and obtains better results. Sarode et al. [3] tested versions with and without an iterative component of their algorithm PCRNet and showed that the iterative approach performed better.

Hence, further approaches toward 3D registration should have iterative components in determining the transformation matrix.

5. RSNets

3D segmentation is the task of determining the class of each point in a 3D scene without any prior knowledge. Figure 3 shows one such example, where an office interior has been segmented into various classes (objects) like chairs, walls, whiteboards, etc.

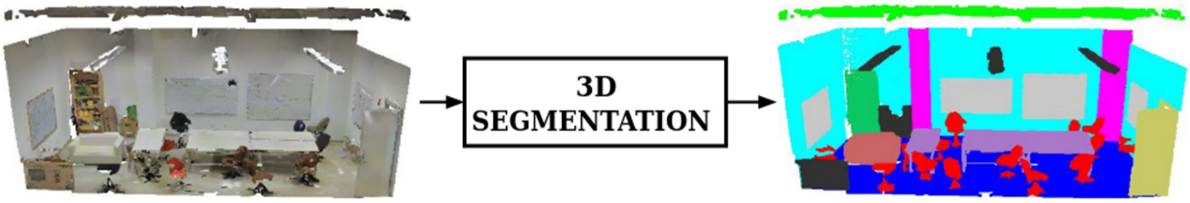

Fig. 3 Example of 3D segmentation of a point cloud. Each color represents an object in the scene such as: chairs (red), turquoise (walls), blue (floor), etc. Taken from [4].

RSNets [4] is a 3D segmentation framework that solves the local context problem by projecting unordered points into ordered features and applying deep end-to-end learning algorithms. Firstly, features are extracted from point clouds using convolutional layers. Next, 3 slice pooling layers are applied separately for every ax which divide the space into N evenly spaced "cubes". Max pooling is applied on every cube to obtain global cube features. This results in an ordered structure that is convenient for a Recurrent Neural Network (RNN). Next, the features are unpooled to obtain a feature representation for every point in the point cloud after which a convolutional layer can predict the segmentation class of the object.

6. Experiment

Since PCRNet presented state-of-the-art performance, we compare it to the newly adapted algorithm (described below), which we denominate as 3DReg-RSNet. The evaluation is performed using the ModelNet40 [54] dataset which contains CAD models of 40 different object categories and has been used for training of many learning approaches such as PCRNet, PRNet, PointNetLK, iterative matching point, etc.

To adapt RSNet to the task of 3D registration there are a few necessary modifications. Firstly, RSNet inputs 9-dimensional points where the first three dimensions represent the x, y and z coordinates, the next three dimensions represent the rgb colors and the last three dimensions represent the batch normalized coordinates. Hence, we expand ModelNet40 points by adding values of 0 for the rgb colors and batch normalized coordinates.

Next, we omit the last few convolutional layers that compress the output to a 13-dimensional vector for every point which represents the class of the 3D segmentation task. We end up with a 512 dimensional representation of each point in the point cloud. We treat this embedding as a description of each point and find correspondences between the point clouds using the first K=10 closest points by their description. Using (9) we find the transformation matrix.

To test the 3D registration capabilities of the algorithms each example in the ModelNet40 dataset is rotated by a random angle around each ax chosen from $\left[0,\frac{\pi}{3}\right]$ and translated by a vector chosen at random from $[-1,1]^3$. The performance of each method is evaluated observing the rotational error between the ground truth rotation and estimated rotation.

The rotational error is calculated as the angle of the angle-axis representation of the matrix product

$$R \cdot R_{est}^{-1}, \qquad (13)$$

where R is the ground-truth rotation matrix and R_{est}^{-1} is the inverse of the estimated rotation matrix. The translational error is simply the Euclidean distance between the ground-truth translation vector and the estimated translation vector.

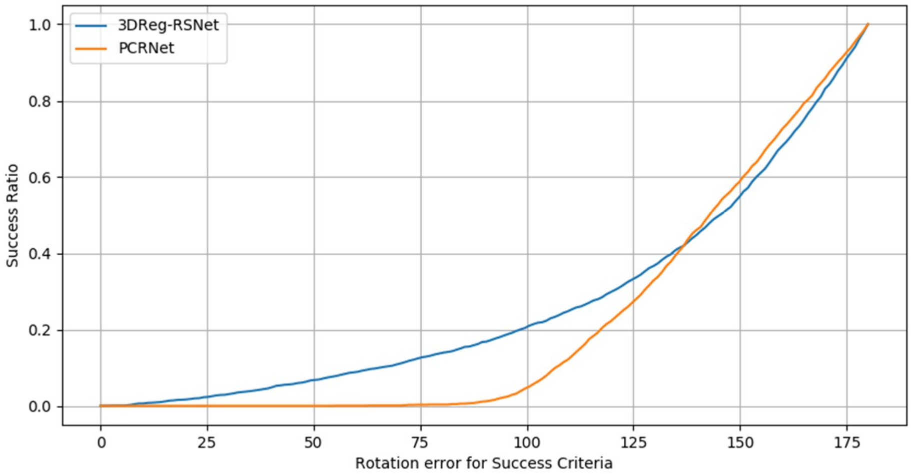

Fig. 4 Performance comparison of `PCRNet` and `3DReg-RSNet`.

The plot shows the success ratio versus success criteria on rotation error (in degrees). The y-axis represents how many examples had fewer rotational error represented on the x-ax. The results indicate that `3Dreg-RSNet` has fewer rotational errors at smaller angles whilst having a higher number of errors at bigger angles. This indicates that the representation, meant for the task of 3D semantic segmentation, of `RSNets` is a good point cloud representation with the ability to register point clouds.

7. Conclusion and future work

In this work we presented the 3D registration process and the most prominent techniques currently present in the literature. The different transformation representations have been represented and the approaches have been classified into two categories: *detection-description-matching* and *all-in-one*. A 3D segmentation algorithms has been adapted to the task of 3D registration and compared with the state-of-the-art registration algorithm `PCRNet`. The results presented indicate that the newly adapted algorithm, named 3DReg-RSNet presents potential for the task of 3D registration.

Adapting the algorithm even further would surely provide even better results than those presented here. Changing the last few layers in `RSNet` to predict a transformation matrix instead of segmentation class and retraining the whole end-to-end process would surely provide better insight to the learning of the feature representations.

Acknowledgments

This research has been partly supported by the Croatian Science Foundation under the contract number IP-2018-01-8118 and partly by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS).

References

[1] Y. Wang and J. Solomon, “Deep closest point: Learning representations for point cloud registration,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 3522–3531, https://doi.org/10.1109/ICCV.2019.00362.

[2] Y. Wang and J. Solomon, “Prnet: Self-supervised learning for partial-to-partial registration,” in 33rd Conference on Neural Information Processing Systems, 2019

[3] V. C. Sarode, X. Li, H. Goforth, Y. Aoki, A. S. Rangaprasad, S. Lucey and H. Choset, “Pcrnet: Point cloud registration network using PointNet encoding,” 2019.

[4] Q. Huang, W. Wang and U. Neumann, "Recurrent Slice Networks for 3D Segmentation of Point Clouds," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2626-2635, https://doi.org/10.1109/CVPR.2018.00278.

[5] J. Yang, Y. Xiao and Z. Cao, "Aligning 2.5D Scene Fragments With Distinctive Local Geometric Features and Voting-Based Correspondences," in IEEE Transactions on Circuits and Systems for

Video Technology, vol. 29, no. 3, pp. 714-729, March 2019, https://doi.org/10.1109/TCSVT.2018.2813083.

[6] G. Blais and M. D. Levine, "Registering multiview range data to create 3D computer objects," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 820-824, Aug. 1995, https://doi.org/10.1109/34.400574.

[7] Huber, D. and M. Hebert. "Fully automatic registration of multiple 3D data sets," Image Vis. Comput., vol. 21, pp. 637-650, 2003, https://doi.org/10.1016/S0262-8856(03)00060-X.

[8] F. Tombari, S. Salti and L. Di Stefano, "Unique Signatures of Histograms for Local Surface Description," in 11th IEEE European Conference on Computer Vision (ECCV) 2010, vol. 6313, pp. 356-369, 2010, https://doi.org/10.1007/978-3-642-15558-1_26.

[9] Y. Guo, F. Sohel, M. Bennamoun, M. Lu and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," in International journal of computer vision, vol. 105, no. 1, pp. 63-86., 2013, https://doi.org/10.1007/s11263-013-0627-y.

[10] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 433-449, 1999, https://doi.org/10.1109/34.765655.

[11] Y. Li, A. Dai, L. Guibas and M. Niessner, "Database-Assisted Object Retrieval for Real-Time 3D Reconstruction," in Comput. Graph. Forum, vol. 34(2), pp. 435–446, 2015, https://doi.org/10.1111/cgf.12573.

[12] M. Magnusson, A. Lilienthal and T. Duckett, "Scan Registration for Autonomous Mining Vehicles Using 3D-NDT," in Journal of Field Robotics, vol. 24., pp. 803-827, 2007, https://doi.org/10.1002/rob.20204.

[13] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," in Int. J. Rob. Res., vol. 34(4–5), pp. 598–626, 2015, https://doi.org/10.1177/0278364914551008.

[14] W. Zhao, D. Nister and S. Hsu, "Alignment of continuous video onto 3d point clouds," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27(8), pp. 1305–1318, 2005, https://doi.org/10.1109/TPAMI.2005.152.

[15] C.-Y. Tsai and C.-H. Huang, "Indoor Scene Point Cloud Registration Algorithm Based on RGB-D Camera Calibration," in Sensors, vol. 17(8):1874, 2017, https://doi.org/doi:10.3390/s17081874.

[16] H. Dai, N. Pears, W. Smith, "Non-rigid 3D Shape Registration Using an Adaptive Template," in 18th IEEE European Conference on Computer Vision (ECCV) 2018, vol 11132, pp. 48-63, 2018, https://doi.org/10.1007/978-3-030-11018-5_5.

[17] N. J. Mitra, N. Gelfand, H. Pottmann and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (SGP '04), pp. 22–31, 2004, https://doi.org/10.1145/1057432.1057435.

[18] F. Tombari, S. Salti and L. Di Stefano, "Performance Evaluation of 3D Keypoint Detectors," in Int. J Comput. Vis., vol. 102, pp. 198–220, 2013, https://doi.org/10.1007/s11263-012-0545-4.

[19] H. Dutagaci, C.P. Cheung and A. Godil, "Evaluation of 3D interest point detection techniques via human-generated ground truth," in Vis. Comput., vol. 28, pp. 901–917, 2012, https://doi.org/10.1007/s00371-012-0746-4.

[20] A. Tonioni, S. Salti, F. Tombari, R. Spezialetti and L. Di Stefano, "Learning to Detect Good 3D Keypoints" in Int. J. Comput. Vis., vol. 126, pp. 1–20, 2018, https://doi.org/10.1007/s11263-017-1037-3.

[21] Y. Diez and F. Roure, Ferran, X. Llado and J. Salvi, "A Qualitative Review on 3D Coarse Registration Methods," in ACM Computing Surveys, vol. 47., https://doi.org/10.1145/2692160.

[22] X. Lin, C. Zhu, Q. Zhang and Y. Liu, "3D Keypoint Detection Based on Deep Neural Network with Sparse Autoencoder," ArXiv, https://arxiv.org/abs/1605.00129.

[23] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, pp. 689-696, 2009, https://doi.org/10.1109/ICCVW.2009.5457637.

[24] A. Zaharescu, E. Boyer, K. Varanasi and R. Horaud, "Surface feature detection and description with applications to mesh matching," in Conference on Computer Vision and Pattern Recognition, Miami, pp. 373-380, 2009, https://doi.org/10.1109/CVPR.2009.5206748.

[25] I. Sipiran and B. Bustos, "Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes," in The Visual Computer, vol. 27(11), pp. 963-976, 2011, https://doi.org/10.1007/s00371-011-0610-y.

[26] J. Sun, M. Ovsjanikov and L. Guibas, “A concise and provably informative multi-scale signature based on heat diffusion,” in Comput. Graph. Forum, vol. 28, pp. 1383-1392, 2009, https://doi.org/10.1111/j.1467-8659.2009.01515.x.

[27] S. Suwajanakorn, N. Snavely, J. Tompson and M. Norouzi, “Discovery of latent 3D keypoints via end-to-end geometric reasoning,” in Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), pp. 2063–2074, 2018

[28] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan and N. Kwok, “A comprehensive performance evaluation of 3d local feature descriptors,” in International Journal of Computer Vision, vol. 116, pp. 66-89, 2016, https//doi.org/10.1007/s11263-015-0824-y.

[29] S. Quan, M. Jie, F. Hu, B. Fang, and T. Ma, “Local voxelized structurefor 3d binary feature representation and robust registration of pointclouds from low-cost sensors,” in Information Sciences, vol. 444, 2018, https://doi.org/10.1016/j.ins.2018.02.070.

[30] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp. 77-85, 2017, https://doi.org/10.1109/CVPR.2017.16.

[31] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao and T. Funkhouser, "3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp. 199-208, 2017, https://doi.rog/10.1109/CVPR.2017.29.

[32] M. Khoury, Q.Y. Zhou and V. Koltun, “Learning compact geometric features,” in IEEE International Conference on Computer Vision (ICCV), pp. 153-161, 2017, https://doi.org/10.1109/ICCV.2017.26.

[33] H. Deng, T. Birdal, and S. Ilic, “PPFNet: Global Context Aware Local Features for Robust 3D Point Matching,” 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, https://doi.org/10.1109/CVPR.2018.00028.

[34] H. Deng, T. Birdal, and S. Ilic, “PPF-Foldnet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors,” in Lecture Notes in Computer Science, pp. 620–638, 2018, https://doi.org/10.1007/978-3-030-01228-1_37.

[35] Z.J Yew., G.H. Lee, “3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration,” in European Conference on Computer Vision (ECCV 2018), pp. 630-646, https://doi.org/10.1007/978-3-030-01267-0_37.

[36] J. Yang et al. “A Performance Evaluation of Correspondence Grouping Methods for 3D Rigid Data Matching.” in IEEE transactions on pattern analysis and machine intelligence, 2019, https://doi.org/10.1109/TPAMI.2019.2960234.

[37] S. Azimi and T.K. Gandhi, “Performance comparison of 3d correspondence grouping algorithm for 3d plant point clouds,” 2019

[38] H. Chen and B. Bhanu, "3D free-form object recognition in range images using local surface patches," in Proceedings of the 17th International Conference on Pattern Recognition 2004, vol. 3, pp. 136-139, 2004, https://doi.org/10.1109/ICPR.2004.1334487.

[39] F. Tombari and L. Di Stefano, "Object Recognition in 3D Scenes with Occlusions and Clutter by Hough Voting," Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore, pp. 349-355, 2010, https://doi.org/10.1109/PSIVT.2010.65.

[40] E. Rodolà, A. Albarelli, F. Bergamasco et al., “A Scale Independent Selection Process for 3D Object Recognition in Cluttered Scenes” in Int. J. Comput. Vis., vol. 102, pp. 129–145, 2013, https://doi.org/10.1007/s11263-012-0568-x.

[41] M. Attia, Y. Slama and M. A. Kamoun, "On Performance Evaluation of Registration Algorithms for 3D Point Clouds," 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), pp. 45-50, 2016, https://doi.org/10.1109/CGiV.2016.18.

[42] D. Aiger, N. J. Mitra, and D. Cohen-or, “4-points Congruent Sets for Robust Pairwise Surface Registration,” in ACM Transactions on Graphics Article, vol.27(10), 2008, https://doi.org/10.1145/1399504.1360684.

[43] M. Mohamad, D. Rappaport and M. Greenspan, "Generalized 4-Points Congruent Sets for 3D Registration," 2nd International Conference on 3D Vision, Tokyo, pp. 83-90, 2014, https://doi.org/10.1109/3DV.2014.21.

[44] N. Mellado, D. Aiger and N. J. Mitra, “Super 4PCS fast global pointcloud registration via smart indexing,” in Computer Graphics Forum, vol. 33(5), pp. 205-215, 2014, https://doi.org/10.1111/cgf.12446.

[45] M. Mohamad, M. T. Ahmed, D. Rappaport and M. Greenspan, "Super Generalized 4PCS for 3D Registration," 2015 International Conference on 3D Vision, Lyon, pp. 598-606, 2015, https://doi.org/10.1109/3DV.2015.74.

[46] Y. Aoki, H. Goforth, R.A. Srivatsan and S. Lucey, “PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet,” 2019 Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7156-7165, 2019, https://doi.org/10.1109/CVPR.2019.00733.

[47] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proceedings of the 7th international joint conference on Artificial intelligence, vol. 2, pp. 674–679, 1981.

[48] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, “DeepICP: An end-to-end deep neural network for 3d point cloud registration,” 2019.

[49] C.R. Qi, L. Yi, H. Su and L. J. Guibas, “PointNet++: deep hierarchical feature learning on point sets in a metric space,” in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), pp. 5105–5114, 2017.

[50] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, “Dynamic Graph CNN for Learning on Point Clouds,” in ACM Trans. Graph., vol. 38(5), 2019, https://doi.org/10.1145/3326362.

[51] J. Li and C. Zhang, “Iterative Matching Point,” in ArXiv abs/1910.10328, 2019.

[52] E. Jang, G. Shixiang and B. Poole, “Categorical Reparameterization with Gumbel-Softmax,” in ArXiv abs/1611.01144, 2017.

[53] J. Yang, H. Li, D. Campbell and Y. Jia, "Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38(11), pp. 2241-2254, 2016, https://doi.org/10.1109/TPAMI.2015.2513405.

[54] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1912-1920, 2015, https://doi.org/10.1109/CVPR.2015.7298801.