A REVIEW OF RIGID 3D REGISTRATION METHODS

David BOJANIĆ; Kristijan BARTOL; Tomislav PETKOVIĆ; Tomislav PRIBANIĆ

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. In this work, we provide an overview of rigid 3D registration methods as well as a breakdown of the different parts of its process, namely, detection, description, and matching (if available). We describe the motivation behind the process and explain in detail the different used approaches in determining the aligning transformation.*
Keywords: *3D computer vision, 3D registration, keypoint detection, keypoint description, keypoint matching*

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 [1,2,3], object recognition and categorization [4,5,6], shape retrieval [7], robot navigation [8,9] and data fusion obtained from different sensors [10]. Fig 1. shows two 3D structures obtained from two different viewpoints of the same object that, when registered, merge into one seamless and coherent object.

Even though some of these problems might be solved using hardware solutions [11] 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 [12] 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.

3D registration methods are classified in two different categories: coarse and fine registration [13]. The former encompasses all techniques that return a rough initial alignment of the given 3D structures, without any given initial alignment. The latter starts from one such approximation and aims at finding a registration as accurate as possible.

Figure 1: Example of partial registration of two point clouds.

2. Data representation

Each of the many applications that use registration techniques has its preferred data representation type. The most commonly used one in the literature are point clouds, followed by meshes and volumetric data. The former is a collection of 3D points with no further information. The second is composed of a point cloud and additional connectivity information between points, usually represented as a graph. The most commonly used format are triangular meshes where the graph edges form triangles. Volumetric data is often used in medical imaging (e.g. MRI) due to the nature of acquisition, and is represented by an isotropic set of samples taken at regularly spaced intervals along three orthogonal axes. The values represent some measurable property of the data like colour, density or heat to name a few. In the next chapters we focus on point clouds and meshes.

3. 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:

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

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:

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

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:

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

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:

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

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:

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

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.

4. 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

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

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). Fig. 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

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

and centroids

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

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.
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

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. The second step of the pipeline, description, consists of assigning values to the detected key points according to the local shape around them. 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. The goal here is to avoid exhaustive search of the whole correspondence space which would yield a cost of $O(n^6)$ since triplets from both clouds need to be checked.

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.

Figure 2: Pipeline of the detection-description-matching approach. Taken from [13].

Detection is a crucial step in the optimization of the whole pipeline. A good detector should be computationally efficient and should extract repeatable and distinctive key points under a number of nuisances that can affect the input data, like for example viewpoint changes, missing parts, point density, clutter and sensor noise [14,15]. Repeatability is the capability to detect the same keypoints accurately under various nuisances, whilst distinctivity is the ability to detect keypoints that can be effectively described and matched, to prevent wrong point-to-point correspondences.

Following evaluation papers and algorithm comparisons [13-17], some of the most promising hand-crafted detectors are: ISS [18], MeshDoG [19], Harris 3D [20] and HKS [21]. ISS uses eigen decomposition of the neighbourhood's covariance matrix of a point to describe it. Keypoints are selected as the points with large three dimensional variations in their neighbourhood. These variations are measured using the smallest eigenvalue of the covariance matrix of its spherical neighbourhood. MeshDoG is based on the Difference of Gaussians (DoG) operator. It finds the extrema of the Laplacian of a scale-space representation of any scalar function defined on a discrete manifold. Harris 3D is a 3D version of the Harris operator which is applied on a 2D projection of the points, without losing relevant information. The point set is rotated and translated according to the centroid of the neighbourhood around a point with the goal of finding the best projection. Finally, HKS, both a detector and descriptor, detects mesh zones with high curvatures and uses the points with the highest curvatures as keypoints. This results in very repeatable keypoints.

On the deep learning side in Salti et al. [16] use random forests to learn the best keypoint detector for a hand-crafted descriptor as to improve the final step of the pipeline, matching. Their method betters the matching of three distinct descriptors, namely SHOT [4], FPFH [22] and Spin Images [6], but can be trained for any descriptor. Lin et al. [17] formulate 3D keypoint detection as a regression problem using deep neural networks with sparse autoencoders. The model makes use of both local and global information of a 3D mesh in multi-scale space to detect whether a vertex is a keypoint or not. Suwajanakorn et al. [23] introduce KeypointNet which tries to learn an optimal set of category-specific 3D keypoints; more concretely it tries to learn a list of 3D keypoints on both views that would result in the best matching of the point clouds. This pose estimation objective helps significantly in producing a reasonable and natural selection of latent keypoints.

Descriptors are sets of values that represent the contained geometric and spatial information within the local surface. These descriptors 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 [24]. Some LRF based methods are SHOT [4], RoPS [5] and LoVS [25]. SHOT divides the local 3D volume into a set of subspaces and concatenates the histograms of normal deviation in each subspace as the final feature representation. RoPS proposes a *rotation and projection* mechanism that continually rotates the local surface with respect to the LRF and performs 3D-to-2D projections for each rotated surface. The eventual feature is the integration of the statistical information of the projected maps. LoVS uses a cubic volume to determine the local surface and performs a uniform spatial partition to generate voxels. Each voxel is encoded by a binary code to judge whether the voxel is empty or not.

Some LRF-independent methods are Spin Images (SI) [6], Point feature histogram (PFH) [26] and Fast PFH (FPFH) [22]. Spin Images projects the neighbouring points of a keypoint on a 2D plane by calculating their horizontal and vertical distances with respect to the tangent plane of the keypoint normal, and the ratio of points in each 2D grid is taken as the bin value. PFH and FPFH leverage the point pair features extracted from the local surface to generate statistical histograms, where PFH considers all possible points but FPFH speeds up the procedure by requiring each point pair to include the keypoint.
There are deep learning based descriptors as well, such as: PointNet [27], 3DMatch [28], CGF [29], PPFNet [30], PPF-FoldNet [31] and 3DFeat-Net [32]. Deep learning descriptors have surpassed many hand-crafted

local geometric descriptors. Nevertheless, the majority of learned descriptors from raw data suffer from sensitivity to rotation. An effective solution is to first use traditional feature descriptors for parametrization and then employ convolutional neural networks to further boost the performance.

PointNet is a pioneer in designing a permutation invariant network which doesn't use voxelized inputs but rather point clouds. 3DMatch is a learning based representation that uses truncated distance function (TDF) to parametrize input local patches and learns the feature representation using a Siamese network paired with a metric learning network. PPFNet employs the point pair features between the keypoint and its neighbors to encode the raw local point cloud and proposes an N-branch network for feature learning. PPF-FoldNet further improves PPFNet by leveraging rotational invariant point pair features and uses a point cloud auto-encoder network to achieve unsupervised feature learning.

Once the point clouds have been filtered (keypoints have been detected) and described, in order to solve the registration problem, the correspondences are needed. Rather than using brute-force methods which are computationally expensive, 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 [33,34], the state-of-the-art methods for finding correspondences are geometric consistency (GC) [35], 3D Hough Voting (3DHV) [36] and game theory matching (GTM) [37].

For an initial set of correspondences C as in (7), the goal is to find a set of inlier points that truly are matches. GC determines the inlier set by finding the largest cluster formed by calculating compatibility score of a single correspondence with all the other correspondences. The compatibility score is a thresholded absolute difference between the distance of the correspondence points. In 3DHV, each correspondence casts a vote in a 3D Hough space. It finds the vectors from the correspondent points to its appropriate centroids, and transforms them into their appropriate LRFs. If the points truly do match, these vectors should be equal. GTM interprets the correspondence grouping as a non-cooperative game. Candidates in the initial correspondence set C are treated as available strategies. Pairs of players play a symmetric game and adapt their behaviour to prefer strategies that receive larger payoffs. Eventually such dynamics converge to a Nash equilibrium.

4.2 All-in-one approaches

In this chapter we present approaches that cannot be separated into the detection-description-matching pipeline as the approaches before. There are several methods [38] that take completely different approaches of finding the optimal transformation. 4-point congruent sets (4PCS) [39] is a global registration algorithm. The global optimality references the finding of the global minima when solving (8). The goal of 4PCS is to find the best aligning transformation according to the largest common point set (LCP) between the source X and target Y point clouds. The LCP of a rigid transformation is the cardinality of the largest subset of the transformed source point cloud X, with the property that every point in the subset is within an ε distance to Y. 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 tat ab intersects cd at the intermediate point e. Given a 4-point base constructed from two intersecting pairs, two ratios can be defined:

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

These ratios are preserved under affine transformations and therefore act as invariants to constrain the search for congruent 4-point bases in X. Under rigid transformations, distances are also preserved and therefore the distances of the two pairs, d_1 and d_2, are also used as invariants. The runtime complexity of the algorithm is $O(N^{2+k})$ where N is the number of points in X and k is the number of congruent bases to be reported. Generalized 4PCS [40] 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 [41] 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. In order to extract pairs efficiently, X is organized using a 3D grid. A regular splitting strategy, like the one used for an octree, is applied to the 3D grid. Using the 3D grid, for each point p from X, the cubes that intersect the spheres of radius d_1 and d_2 cantered at p are computed to form the pairs. To address the second bottleneck, the search is constrained to searching source bases that have the same angle of intersection as the target base, since angles are preserved under rigid transformation. Lastly, Super Generalized 4PCS [42] combines the two solutions.

A completely different approach is done by PointNetLK [43] which tries to utilize the classical Lucas & Kanade (LK) algorithm [44]. PointNetLK uniquely uses the Lie algebra se(3) representation of the transformation matrix, as shown in (6), that needs to be found 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 in each iteration is the difference of the embedded point clouds. DeepICP [45] is an end-to-end learning-based point cloud registration framework. The algorithm firstly extracts feature descriptors from both the point clouds using PointNet++ [46]. After that, a point weighting layer is executed to learn the saliency of each point. Ideally, points with invariant and distinct features on static objects should have higher weights. 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 that can better represent their geometric characteristics. More concretely, a smaller PointNet, denominated mini-PointNet, network is applied for extracting those features. 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 [47] is another end-to-end deep learning framework that could potentially be classified as a *detection-description-matching* approach since the various components of the algorithm resemble approaches from that group. The algorithm firstly embeds the point clouds using PointNet or DGCNN [48]. 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 [49] 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 [50] 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 (14) and the softmax solution from DCP. The reason being that (14) 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 [51] approach to sample a matching matrix. Here, a "blurring" parameter is added to use a softer matching in the begging of the training, to a more sharp matching at the end. From there, the alignment is easily determined. 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 [52] 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.

After two point clouds have been coarsely matched, the alignment can further be refined by fine registration. The goal in fine registration is to align the two point clouds as best as possible given the initial conditions calculated by the coarse registration algorithm. Here, Iterative Closest Point (ICP) [53] is the go-to method. Even though the algorithm has problems with wrong local minima solutions, it is still one of the most famous and used algorithms today since its conception in 1992. Very similarly to (7) and (8), ICP tries to minimize the point-to-point distances between the clouds:

$$E(R,t) = \sum_{i=1}^{N} e_i(R,t)^2 = \min_{R,t} \sum_{(x_i,y_j)\,\epsilon\,C} \left\| R \cdot x_i + t - y_{j*} \right\|_2^2 \qquad (13)$$

Where $e_i(R,t)$ is the per-point residual error for x_i. Given R and t, the point y_{j*} from Y is denoted as the optimal correspondence of x_i, which is the closest point to the transformed x_i from Y, i.e.

$$j* = \underset{j\,\epsilon\,\{1,\ldots,M\}}{\operatorname{argmin}} \left\| R \cdot x_i + t - y_j \right\|_2^2. \qquad (14)$$

Again, these equations pose an chicken-and-egg problem: if the true correspondences are known a priori, the transformation can be optimally solved in closed form (9); if the optimal transformation is given, correspondences can also be readily found. However, the joint problem cannot be trivially solved. Hence, given

an initial transformation (R,t), ICP iteratively solves the problem by alternating between estimating the transformation with (13), and finding closest-point matches with (14). However, since (13) is non-convex, there is no guarantee that ICP will reach a global optimum.

Another issue with point-to-point distance is that the correspondence of a given point in the first cloud may not exist in the second cloud because of the limited number of points acquired by the sensor. Point-to-plane ICP tries to address this issue using the distance between a point and a planar approximation of the surface at the corresponding point. More concretely point-to-plane minimizes:

$$\underset{R,t}{\operatorname{argmin}} \sum_i \left\| (R \cdot x_i + t - y_j) \cdot n_j \right\|_2^2 . \qquad (15)$$

where n_j is the surface normal in point y_j. When the initial position of the data is close to the model, and when the input has relatively low noise, ICP with the point-to-plane error metric has faster convergence than the point-to-point version.

Many other variants of ICP [54] have been proposed that try and solve its drawbacks. Gelfand et al. [55] try to improve the efficiency and quality of the algorithm by introducing a sampling method of the point clouds. If too many points are chosen from featureless regions of the data, the algorithm converges slowly, finds the wrong pose, or even diverges, especially in the presence of noise or miscalibration in the input data. Hence, they try different cloud sampling methods such as uniform, random and normal-space sampling. They conclude that covariance sampling gives the best results along with the fastest convergence. Other than the point-to-point and point-to-plane methods mentioned before, there are other proposed variants to determine the matches. The point-to-projection [56] approach finds the correspondence of a source control point by projecting it onto a target surface from the point of view of the target. Combinations of point-to-projection and point-to-plane approaches have also been explored [57].

The proposed variants can still create mismatches. By rejecting point pairs using a threshold and reserving only the points with the smallest distances, the algorithm becomes more robust. Other criteria to remove mismatches include geometric properties and invariance of data, such as normal consistency and reciprocal correspondence.

Nevertheless, most ICP variants, as well as ICP, have a few fundamental drawbacks and because of these reasons, Yang et al. propose GO-ICP [58] which was the first globally optimal algorithm that performed rigid registration. The algorithm 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. Fig. 3 shows how the BnB and ICP methods complement each other to find a globally optimal solution.

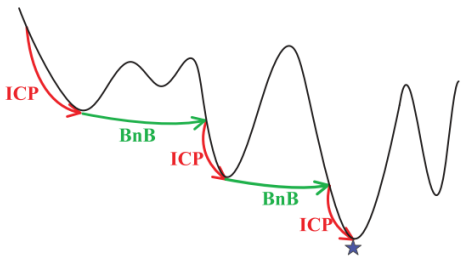

Figure 3: Alternation of the BnB and ICP for global optimality [58].

Other than ICP, there are other fine matching algorithms that are based on random sample consensus (RANSAC) [59], normal distribution transform methods (NDT), genetic algorithms or usage of auxiliary data along with the point clouds. RANSAC [60,61,62] methods involve three steps: find 3 correspondences by some heuristic, find the alignment of the point clouds with these 3 correspondences and finally count the number of inliers (points between two clouds that are within a threshold distance). The quality of every choice of 3 correspondences can be measured using the number of inliers. The more inliers a choice has, the more trustworthy is the alignment.
NDT was originally developed in the context of 2D laser scan registration [63]. The algorithm represents the observed range points as a set of Gaussian probability distributions. Assuming the point clouds have been drawn from a Gaussian distribution, the maximum-likelihood estimates of the mean and covariance can be

obtained from the observations. The fitness of the Gaussian distribution might not be good for the entire point cloud since generally there is no reason a surface should behave normally. Nevertheless, at a sufficiently small scale, a normal distribution can be considered a good estimate of the local surface shape. Therefore, the basic principle of NDT is to represent space using a set of Gaussian probability distributions.

The point-to-distribution (P2D) variant of NDT for 3D registration [64] maximises the likelihood of points from one scan, given the NDT model created from the source scan X, whilst the distribution-to-distribution (D2D) variant of NDT [65] minimizes the sum of L2 distances between pairs of Gaussian distributions in two NDT models. The NDT algorithms have fast computational speed and high precision. They are especially suitable for processing large-scale point clouds. However, initialization requirements still remain high.

5. Conclusion

This paper presented the 3D registration process and the most prominent techniques currently present in the literature. The different used data types have been presented, as well as the different transformation representations. The approaches have been classified in two different categories: *detection-description-matching* and *all-in-one*. The former use detectors to filter the number of points, descriptors to describe the remaining points and "matchers" to find correspondences from which the rigid transformation can be estimated. The latter try to approximate the transformation directly. The approaches have also been classified to coarse and fine registration methods. The former roughly align the point clouds without any initial requirements, while the latter need an already good estimation of the transformation to further improve the alignment. The approaches have also been labelled as hand-crafted or deep learning approaches depending on the algorithm.

Acknowledgement

This research has been (partly) supported by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS).

References

[1] Yang, J. et al.: Aligning 2.5D Scene Fragments With Distinctive Local Geometric Features and Voting-Based Correspondences, *IEEE Transactions on Circuits and Systems for Video Technology*, **29** (2019) 3, pp. 714-729, ISSN: 1051-8215
[2] Blais, G. & Levine, M. D.: Registering multiview range data to create 3D computer objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17** (1995) 8, pp. 820-824, ISSN: 1939-3539
[3] Huber, D. & Hebert, M.: Fully automatic registration of multiple 3D data sets, *Image Vis. Comput.*, **21** (2003), pp. 637-650, ISSN: 0262-8856
[4] Tombari, F. et al: Unique Signatures of Histograms for Local Surface Description, *Proceedings of 11th IEEE European Conference on Computer Vision (ECCV) 2010*, pp. 356-369, ISBN: 978-3-642-15558-1, Greece, 2010, Springer, Berlin, (2010)
[5] Guo, Y. et al.: Rotational projection statistics for 3D local surface description and object recognition, *International journal of computer vision*, **105** (2013) 1, pp. 63-86, ISSN: 1573-1405
[6] Johnson, A. E. & Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **21** (1999) 5, pp. 433-449, ISSN: 1939-3539
[7] Li, Y. et al.: Database-Assisted Object Retrieval for Real-Time 3D Reconstruction, *Comput. Graph. Forum*, **34** (2015) 2, pp. 435-446, ISSN: 0167-7055
[8] Magnusson, M. et al.: Scan Registration for Autonomous Mining Vehicles Using 3D-NDT, *Journal of Field Robotics*, **24** (2007) 10, pp. 803-827, ISSN: 1556-4959
[9] Whelan, T. et al.: Real-time large-scale dense RGB-D SLAM with volumetric fusion, *Int. J. Rob. Res.*, **34** (2014) 4-5, pp. 598-626, ISSN: 0278-3649
[10] Zhao, W. et al.: Alignment of continuous video onto 3D point clouds, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27** (2004) 8, pp. 1305-1318, ISSN: 1939-3539
[11] Tsai, C.-Y. & Huang, C.-H.: Indoor Scene Point Cloud Registration Algorithm Based on RGB-D Camera Calibration, *Sensors*, **17** (2017) 8, 19 pages, ISSN: 1424-8220
[12] Dai, H. et al.: Non-rigid 3D Shape Registration Using an Adaptive Template, *Proceedings of 18th IEEE European Conference on Computer Vision (ECCV) 2018*, vol. 11132, pp. 48-63, ISBN: 978-3-030-11018-5, Germany, 2018, Springer, Berlin, (2018)
[13] Diez, Y. et al: A Qualitative Review on 3D Coarse Registration Methods, *ACM Computing Surveys*, **47** (2015) 3, pp. 45, ISSN: 1557-7341
[14] Tombari, F. et al: Performance Evaluation of 3D Keypoint Detectors, *Int. J Comput. Vis.*, **102** (2013), pp. 198-220, ISSN: 1550-6185

[15] Dutagaci, H. et al.: Evaluation of 3D interest point detection techniques via human-generated ground truth, *Vis. Comput.,* **28** (2012) 9, pp. 901-917, ISSN: 1432-2315
[16] Tonioni, A. et al.: Learning to Detect Good 3D Keypoints, *Int. J. Comput. Vis.*, **126** (2018), pp. 1-20, ISSN: 1573-1405
[17] Lin, X. et al.: 3D Keypoint Detection Based on Deep Neural Network with Sparse Autoencoder, *Available from* https://arxiv.org/abs/1605.00129.
[18] Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3D object recognition, *Proceedings of 12th International Conference on Computer Vision Workshops*, pp. 689-696, ISBN: 978-1-4244-4442-7, Japan, (2009)
[19] Zaharescu, A. et al.: Surface feature detection and description with applications to mesh matching, *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 373-380, ISBN: 1063-6919, USA, (2009), IEEE, Miami, (2009)
[20] Sipiran, I. & Bustos, B.: Harris 3d: A robust extension of the harris operator for interest point detection on 3D meshes, *The Visual Computer*, **27** (2011) 11, pp. 963-976, ISSN: 1432-2315
[21] Sun, J. et al.: A concise and provably informative multi-scale signature based on heat diffusion, *Proceedings of SGP09: Eurographics Symposium on Geometry Processing,* **28** (2009) 5, pp. 1383-1392, ISBN: 1467-8659, Germany, (2009)
[22] Rusu, R .B. et al.: Fast Point Feature Histograms (FPFH) for 3D registration, *Proceedings of 2009 IEEE International Conference on Robotics and Automation*, pp. 3212-3217, ISBN: 1050-4729, Japan, (2009)
[23] Suwajanakorn S. et al.: Discovery of latent 3D keypoints via end-to-end geometric reasoning, *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18),* Bengio, S. et al (Eds), pp. 2063-2074, Curran Associates Inc., NY, (2018)
[24] Guo, Y. et al.: A comprehensive performance evaluation of 3D local feature descriptors, *International Journal of Computer Vision*, **116** (2015) 1, pp. 66-89, ISSN: 1573-1405
[25] Quan, S. et al.: Local voxelized structure for 3D binary feature representation and robust registration of pointclouds from low-cost sensors, *Information Sciences*, **444** (2018) May 2018, pp. 153-171, ISSN: 0020-0255
[26] Rusu, R.B.: Aligning point cloud views using persistent feature histograms, *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3384-3391, ISBN: 2153-0866, France, (2008)
[27] Charles, R. Q. et al.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77-85, ISBN: 1063-6919, Hawaii, (2017)
[28] Zeng, A. et al.: 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions, *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 199-208, ISBN: 1063-6919, Hawaii, (2017)
[29] Khoury, M. et al: Learning compact geometric features, *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 153-161, ISSN: 2380-7504, Venice, (2017)
[30] Deng, H. et al.: PPFNet: Global Context Aware Local Features for Robust 3D Point Matching, *Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, ISBN: 1063-6919, (2018)
[31] Deng, H. et al.: PPF-Foldnet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors, *Proceedings of European Conference on Computer Vision (ECCV 2018)*, pp. 620–638, ISBN: 978-3-030-01228-1, Munich, (2018)
[32] Yew, Z.J. & Lee, G.H.: 3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration, *Proceedings of European Conference on Computer Vision (ECCV 2018)*, pp. 630-646, ISBN: 978-3-030-01267-0, Munich, (2018)
[33] Yang, J. et al.: A Performance Evaluation of Correspondence Grouping Methods for 3D Rigid Data Matching, *IEEE transactions on pattern analysis and machine intelligence*, **14** (2015) 8, 15 pages, ISSN: 1939-3539
[34] Azimi, S. & Gandhi, T.K.: Performance comparison of 3D correspondence grouping algorithm for 3D plant point clouds, *Available from* https://arxiv.org/abs/1909.00866
[35] Chen, H. & Bhanu B.: 3D free-form object recognition in range images using local surface patches, *Proceedings of the 17th International Conference on Pattern Recognition 2004*, vol. 3, pp. 136-139, ISBN: 1051-4651, Cambridge, (2004)
[36] Tombari, F. & Di Stefano, L.: Object Recognition in 3D Scenes with Occlusions and Clutter by Hough Voting, *Proceedings of Fourth Pacific-Rim Symposium on Image and Video Technology*, pp. 349-355, ISBN: 978-1-4244-8890-2, Singapore, (2010)
[37] Rodolà, E. et al.: A Scale Independent Selection Process for 3D Object Recognition in Cluttered Scenes, *Int. J. Comput. Vis.*, **102** (2013) 1-3, pp. 129-145, ISSN: 1573-1405

[38] Attia, M. et al.: On Performance Evaluation of Registration Algorithms for 3D Point Clouds, *Proceedings of 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pp. 45-50, ISBN: 978-1-5090-0811-7, Beni Mellal, (2016)
[39] Aiger, D. et al.: 4-points Congruent Sets for Robust Pairwise Surface Registration, *ACM Transactions on Graphics Article*, **27** (2008) 3, ISSN: 1557-7368
[40] Mohamad, M. et al: Generalized 4-Points Congruent Sets for 3D Registration, *Proceedings of 2nd International Conference on 3D Vision*, pp. 83-90, ISBN: 978-1-4799-7000-1, Tokyo, (2014)
[41] Mellado, N. et al.: Super 4PCS fast global pointcloud registration via smart indexing, *Computer Graphics Forum*, **33** (2014) 5, pp. 205-215, ISSN: 1467-8659
[42] Mohamad, M. et al.: Super Generalized 4PCS for 3D Registration, *Proceedings of 2015 International Conference on 3D Vision*, pp. 598-606, ISBN: 978-1-4673-8332-5, Lyon, (2015)
[43] Aoki, Y. et al.: PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet, *Proceedings of 2019 Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 7156-7165, ISBN: 2575-7075, Long Beach, (2019)
[44] Lucas, B.D. & Kanade, T.: An iterative image registration technique with an application to stereo vision, *Proceedings of the 7th international joint conference on Artificial intelligence*, vol. 2, pp. 674-679, Vancouver, (1981) *Available from* https://cecas.clemson.edu/~stb/klt/lucas_bruce_d_1981_1.pdf
[45] Lu, W.: DeepICP: An end-to-end deep neural network for 3D point cloud registration, 2019, *Available from* https://arxiv.org/abs/1905.04153
[46] Qi, C. R. et al.: PointNet++: deep hierarchical feature learning on point sets in a metric space, *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, pp. 5105–5114, ISBN: 978-3-319-94042-7, Long Beach, (2017)
[47] Wang, Y. & Solomon, J.: Deep closest point: Learning representations for point cloud registration, 2019, *Available from* https://arxiv.org/abs/1905.03304
[48] Wang, Y. et al.: Dynamic Graph CNN for Learning on Point Clouds, *ACM Trans. Graph.,* **1** (2019) 1, 13 pages, ISSN: 1557-7368
[49] Li, J. & Zhang, C.: Iterative Matching Point, *Available from* http://export.arxiv.org/pdf/1910.10328
[50] Wang, J. & Solomon, J.: Prnet: Self-supervised learning for partial-to-partial registration, *Proceedings of 33rd Conference on Neural Information Processing Systems*, pp. 8814-8826, Vancouver, (2019)
[51] Jang, E. et al.: Categorical Reparameterization with Gumbel-Softmax, *Available from* https://arxiv.org/abs/1611.01144
[52] Sarode, V. C. et al.: Pcrnet: Point cloud registration network using PointNet encoding, 2019, *Available from* https://arxiv.org/abs/1908.07906
[53] Besl, P. & McKay, H.: A method for registration of 3-d shapes, *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14** (1992) 2, pp. 239-256, ISSN: 1939-3539
[54] Rusinkiewicz, S. & Levoy, M.: Efficient variants of the icp algorithm, *Proceedings Third International Conference on 3-D Digital Imagingand Modeling*, pp. 145-152, ISBN: 0-7695-0984-3, Quebec City, (2001)
[55] Gelfand, N. et al.: Geometrically stable sampling for the icp algorithm, *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling*, pp. 260-267, ISBN: 0-7695-1991-1, Banff, (2003)
[56] Neugebauer, P. J.: Geometrical cloning 3D objects via simultaneous registration of multiview range images, *Digitally Archiving Cultural Objects,* pp. 71-88, 978-0-387-75807-7
[57] Park, S. & Subbarao, M.: A fast point-to-tangent plane techniquefor multi-view registration in *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling,* pp. 276-283, ISBN: 0-7695-1991-1, Banff, (2003)
[58] Yang, J. et al.: Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38** (2016) 11, pp. 2241-2254, ISSN: 0162-8828
[59] Fischler, M. A. & Bolles, R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, **24** (1981) 6, pp. 381-395, ISSN: 0001-0782
[60] Fontanelli, D. et al.: A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements,*Proceedings of 2007 IEEE International Conference on Automation Science and Engineering*, pp. 597-602, ISBN: 2161-8089, Scottsdale, (2007)
[61] Pankaj, D. S. & Nidamanuri, R. R.: A robust estimation technique for 3d point cloud registration, *Image Analysis Stereology*, **35** (2016) 1, pp.15-28, ISSN: 1580-3139
[62] Chen, C.S. et al.: Ransac-based darces: a new approach to fast automatic registration of partially overlapping range images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21** (1999) 11, pp. 1229–1234, ISSN: 1939-3539
[63] Biber, P. & Strasser, W: The normal distributions transform: A new approach to laser scan matching, *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2743-2748, ISBN: 0-7803-7860-1, Las Vegas, (2003)

[64] Magnusson, M. & Duckett T.: A comparison of 3D registration algorithms for autonomous underground mining vehicles, *Proceedings of the European Conference on Mobile Robots (ECMR) 2005*, pp. 86-91, Ancona, (2005)

[65] Stoyanov, T. et al.: Fast and accurate scan registration through minimization of the distance between compact 3D ndt representations, *The International Journal of Robotics Research*, **31** (2012) 12, pp. 1377-1393, ISSN: 0278-3649

Authors:

Mag. Math. David Bojanić (corresponding author)
Mag. Ing Kristijan Bartol; Assist. Prof. Tomislav Petković, PhD.; Prof. Tomislav Pribanić, PhD.
University of Zagreb Faculty of Electrical Engineering and Computing
Unska ul. 3, 10 000 Zagreb, Croatia
Phone: +(385) (1) 3712 543
E-mail: David.Bojanic@fer.hr
E-mail: tomislav.petkovic.ml@fer.hr