Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Lidar point–to–point correspondences for rigorous registration of kinematic scanning in dynamic networks

Aurélien Brun [a], Davide A. Cucci [b,*], Jan Skaloud [a]

[a] EPFL ENAC TOPO, Bâtiment GC - Station 18, 1015 Lausanne, Switzerland
[b] Geneva School of Economics and Management, University of Geneva, Bd du Pont-d'Arve 40, 1205 Genève, Switzerland

ARTICLE INFO

Keywords:
Lidar
Georeferencing
Point cloud registration
UAVs
Pose-graph optimization
Dynamic networks

ABSTRACT

With the objective of improving the registration of lidar point clouds produced by kinematic scanning systems, we propose a novel trajectory adjustment procedure that leverages on the automated extraction of selected reliable 3D point–to–point correspondences between overlapping point clouds and their joint integration (adjustment) together with raw inertial and GNSS observations. This is performed in a tightly coupled fashion using a dynamic network approach that results in an optimally compensated trajectory through modeling of errors at the sensor, rather than the trajectory, level. The 3D correspondences are formulated as static conditions within the dynamic network and the registered point cloud is generated with significantly higher accuracy based on the corrected trajectory and possibly other parameters determined within the adjustment. We first describe the method for selecting correspondences and how they are inserted into the dynamic network via new observation model while providing an open-source implementation of the solver employed in this work. We then describe the experiments conducted to evaluate the performance of the proposed framework in practical airborne laser scanning scenarios with low-cost MEMS inertial sensors. In the conducted experiments, the method proposed to establish 3D correspondences is effective in determining point–to–point matches across a wide range of geometries such as trees, buildings and cars. Our results demonstrate that the method improves the point cloud registration accuracy (∼5× in nominal and ∼10× in emulated GNSS outage conditions within the studied cases), which is otherwise strongly affected by errors in the determined platform attitude or position, and possibly determine unknown boresight angles. The proposed methods remain effective even if only a fraction (∼0.1%) of the total number of established 3D correspondences are considered in the adjustment.

1. Introduction

1.1. Challenges in kinematic laser scanning

Thanks to active sensing that increases the discrimination of fine elements, an ability to ”peer through” vegetation and directly reconstruct three-dimensional surfaces, kinematic laser scanning remains a key technology to provide foundational data for the creation of digital twins of infrastructure (both outdoor and indoor) cities as well as high-resolution terrain models. Together with its independence on external illumination as an active sensor, these properties make mobile laser scanning (MLS) indispensable for rapid digitalization of fine features in urban infrastructure and airborne laser scanning (ALS) the essential tool for deriving digital elevation models, precise quantification of above ground biomass, canopy structure, snow accumulation, detection of stream networks, high-resolution bathymetry, etc. However, this modern technology does not come without disadvantages. Under certain circumstances, due to its need for direct orientation, kinematic laser scanning is challenging in dense urban environments, indoors and in areas where GNSS signal occlusions occur in general; as well as on small UAVs or high-altitude carriers where the influence of attitude noise on point cloud registration surpasses that of ranging accuracy. A standard processing scheme of kinematic laser scanning is depicted in Fig. 1a. As discussed in detail by Glennie (2007), the accuracy of individual points is, apart from ranging quality, affected by the forward propagation of errors starting from the trajectory determination by navigation sensors (outdoors mainly INS/GNSS), the incertitude in mounting (lever-arm, boresight) and the parameters related to scanner interior orientation. These, together with sampling density, influence the geometrical quality of derived elevation models (Skaloud and Schaer, 2012) or other objects

[*] Corresponding author.
E-mail addresses: aurelien.brun@alumni.epfl.ch (A. Brun), davide.cucci@unige.ch (D.A. Cucci), jan.skaloud@epfl.ch (J. Skaloud).

https://doi.org/10.1016/j.isprsjprs.2022.04.027
Received 30 September 2021; Received in revised form 28 April 2022; Accepted 29 April 2022
Available online 19 May 2022

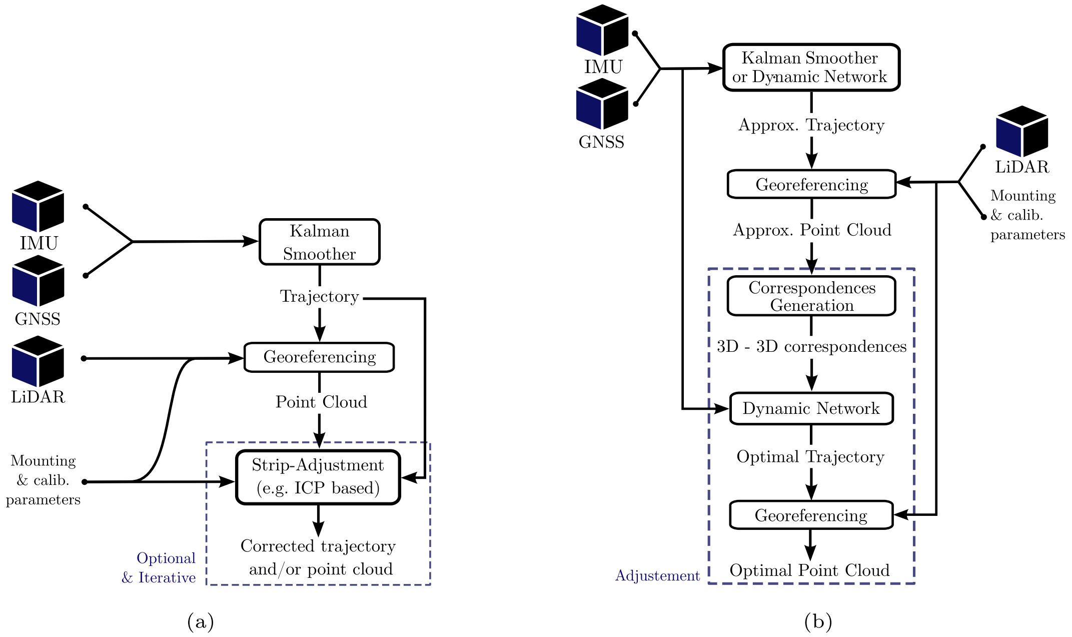

Fig. 1. Overview of lidar georeferencing approaches. (a) Conventional: Kalman smoother and strip adjustment (b) Proposed: Dynamic Network with raw IMU, GNSS and 3D correspondences.

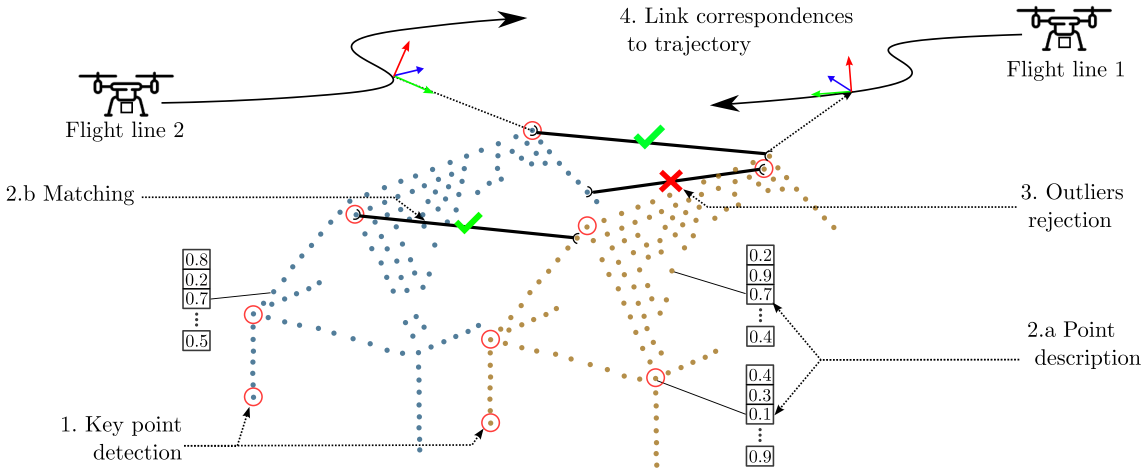

Fig. 2. Overview of steps to connect trajectories via lidar pulse observations through the *key point–to–point* detection and matching principle. Two overlapping and misaligned lidar point clouds in blue and orange are related to a portion of two overlapping flight lines 1 and 2.

of interest. The prevailing source of error depends on the application and sensor quality. Generally, indoor and MLS applications are more influenced by position error due to GNSS signal occlusion; while ALS is generally more affected by the projection of attitude errors due to longer ranging distances.

1.2. Related work

When the accumulated errors in kinematic laser systems are higher than the ranging noise, the discrepancies between scans from different perspectives (directions, altitude, etc.) become noticeable and thus the need to mitigate them arises. This holds for conventional mapping as well as for specially designed "in-motion" calibration scenarios. In the latter case, the aspects of trajectory quality are usually controlled through planning, so the subsequent adjustment can be formulated in a rigorous way either with "good enough" trajectories as in e.g., Kager (2004); Filin (2003); Skaloud and Lichti (2006); Friess (2006); Kerstling et al. (2012), or with minimum assumptions on the trajectory deficiency as in e.g., Filin and Vosselman (2004); Hebel and Stilla (2012); Glira et al. (2015b). In the latter situation, trajectory corrections are often modeled as angular and/or position offsets that are considered time-invariant either per block or per strip/flight line. While these assumptions are possibly reasonable for situations in which laser scanners are mounted on stabilized platforms with high quality IMUs, they do not hold in general. For instance, in MLS the shape of the trajectory due to errors in position may change quickly and non-linearly in GNSS-denied environments (Schaer and Vallet, 2016). For non-stabilized lasers on platforms such as helicopters or Unmanned Aerial Vehicles (UAVs) that do not necessarily fly in regular grid pattern (e.g., with a constant velocity and minimal angular change during flight lines), the projection of orientation-induced errors varies substantially within a strip. Furthermore, in lightweight UAVs employing industrial-grade IMUs, the error signatures of point clouds created by attitude-projected errors become "too wavy" to be captured by such simple modelling (Vallet et al., 2020).

In short, in many cases the deficiencies in the estimated trajectory cause time-dependent, non-linear, and possibly fast-changing deformations of point clouds during data collection. To mitigate such errors one can either attempt to model their *effect* on the trajectory, as proposed for instance by Glira et al. (2016), or directly their *cause* as put forward by this contribution. We argue that while the former approach may work in certain scenarios, the latter is rigorous and thus can be applied more generally. Furthermore, and as we will demonstrate practically, the modelling and estimation of the *cause* requires a relatively small number of 3D tie-features to correct errors in the trajectory (and indirectly in the point cloud) that otherwise have complex appearances in the mapping frame.

A good selection of tie-features is an important prerequisite for the proposed method to be applied during general-operation as well as during in-motion calibration. The parameter recovery of non-calibrated kinematic scanning systems usually benefits from some a priory knowledge on the existing physical form of scanned patches (e.g., planarity, etc.) as in Kager (2004); Filin (2003); Friess (2006); Glennie and Lichti (2010). When coupled with a strong observation geometry, such conditioning tolerates poor initial registration quality, guarantees parameter observability (Skaloud and Lichti, 2006) and allows automation (Skaloud and Schaer, 2007). Nevertheless, this approach is better suited for calibration than for general mapping where the a priory assumption on the presence of certain surface geometries can not be made. Hence, the conditioning is usually formulated as some surface to surface intrinsic, point to surface patch (Kerstling et al., 2012) or point–to–point. The last option usually stems from some evolved variants of the Iterative Closest Point (ICP) algorithm, independently proposed by Besl and McKay (1992) and Chen and Medioni (1992) for registration between terrestrial scanning stations. As point–to–point minimization is practically simpler to deal with, it is also widely applied in ALS (Glira et al., 2015a). Nevertheless, this approach exhibits only a local linear convergence (Pottmann et al., 2006) and therefore requires a good approximation of the initial registration; a case that may not hold in UAV based ALS or MLS due to previously mentioned reasons. In the proposed methodology we assume that only some point–to–point correspondences can be found within the overlapping regions of measured point clouds based on the analysis of their surroundings. In this respect, our goal is not to propose a new algorithm to establish such correspondences, but rather to exploit and adapt some recent advances in this domain when utilizing it within the dynamic networks adjustment approach. Although we previously studied the potential benefits of employing a common modeling-estimation approach in kinematic laser scanning for trajectory estimation within a simulated scenario (Rouzaud and Skaloud, 2011), the proposed adjustment approach involves basic steps introduced within our former work in photogrammetry (Cucci et al., 2017b) while expanding it to make use of 3D point–to–point correspondences (derived from an approximated point cloud) instead of 2D tie-points derived from images.

1.3. Proposed approach

We adapt the final stage of the formerly described kinematic laser scanning registration methodology (Fig. 1a). There we propose to utilize raw inertial and GNSS observations together with a percentage of the laser measurements within the overlapping point cloud regions in a common adjustment step as schematically shown in Fig. 1b. Thanks to the rigorous modeling of observations and error sources on the sensor and system level that this approach yields, an optimal registration of the whole trajectory is achieved. The obtained optimal trajectory (and possibly other system parameters) are then applied in the final registration (georeferencing) of the measured point clouds.

The remainder of this work is organised as follows. In Section 2 we describe the novel aspects of the methodology depicted in Fig. 1b. These consist in the algorithm to extract point–to–point correspondences from overlapping areas between scans and in the way dynamic network fuses those with raw inertial and GNSS observations to obtain an optimal trajectory. In Section 3 we describe the design and implementation of empirical tests that were established to assess the merits of the proposed method with respect to high-quality references in position, attitude and final registered point clouds. In Section 4 we provide a reference implementation of the Dynamic Network solver employed in this work and discuss the results of conducted experiments: we first consider a nominal scenario where we correct the registration errors (mainly) caused by the noise of the MEMS-IMU attitude over distances that are expected to be achieved by modern UAV based laser scanners (Riegl, 2021). We then study the possibility of recovering some calibration parameters, such as an imperfect lidar boresight, within the adjustment process. Finally, we look at the capacity of the method to correct errors in the point cloud registration for a trajectory where GNSS outages occur.

2. Methodology

This section describes how lidar point–to-point correspondences are identified between overlapping point clouds and how these can be included in a dynamic network adjustment to determine continuous, non-rigid and optimal correction of the whole trajectory.

2.1. Point detection and description

Preprocessing:the overlapping region between point clouds from two different flight lines is extracted and divided in rectangular tiles of tractable size. This is done by splitting the two point clouds in smaller rectangular grids and keeping only the tiles where points of both clouds are present. The steps that follow, namely detection, description and matching, are performed independently on each pair of tiles.

Point detection:the 3D key point detection step aims at selecting salient points from a point cloud (edges, corners) where their distinctiveness with respect to their neighborhood is expected to ease matching with corresponding points in other point clouds. We do not aim to design a detector, rather we have evaluated several state-of-the-art detectors on close-range ALS data (ranges < 300 m) and applied the most suitable one. We have selected the ISS (Intrinsic Shape Signature) algorithm (Zhong, 2009) to detect key points because of its robustness and performance compared to the other algorithms tested, as mentioned in Salti et al. (2012). ISS was originally designed to be a descriptor but its first 3 steps can be used independently to detect key points. We use the open-source implementation shipped with the Point Cloud Library (Rusu and Cousins, 2011), which enables the detection of key points in a few seconds for one tile.

ISS employs the following steps to detect key points:

1. For every point, its spherical neighborhood is extracted and a weight (inversely proportional to its number of neighbors) is computed. The radius of the spherical neighborhood to extract is related to the typical size of geometric elements one wants to observe. In denser ALS clouds it is of the order of magnitude of one meter to include cars, fences, trees, etc.
2. For every point, the covariance matrix of its neighborhood is computed and its three eigenvalues are extracted.
3. Points that show high ratio in the successive eigen values of their covariance matrix are selected as key point candidates.

The complete formulation of the algorithm can be found in the original publication as well as in the PCL source code (Rusu and Cousins, 2011). The detection is performed separately on the two point clouds of each tile pair, returning for each of these pairs two independent sets of key points.

Point description:the aim of point description step is to compute a lower dimensionality representation of a point and its surroundings or, in other words, to encode its (geometric) neighborhood into a feature vector so that similar structures have similar feature vectors. It is then possible to match points by comparing their feature vectors using a distance metric, e.g., the L2 norm of the difference of the two descriptors.

In the case of close-range ALS over a mixed natural-built environment, LCD (Pham et al., 2019) showed the best descriptiveness and robustness, the reason for which it was selected in our final implementation. Other descriptors are compared in Appendix A. LCD is a Siamese neural network originally designed for 2D to 3D matching between images and point clouds but the neural network responsible for description of 3D points can be used independently. LCD is able to establish lidar correspondences on a diverse range of surfaces (e.g., man made constructions and vegetation). We used the open source pre-trained version of the network (trained on indoor point clouds). LCD works as follows: given a selected point, its spherical neighborhood is extracted and for each neighbor, its relative coordinates are computed taking the key-point as the origin of the sphere, making the descriptor translation invariant. The neighborhood is then fed to the neural network, which outputs a feature vector for the point. The description is performed on the two point clouds of each tile pair and outputs one feature vector for each point.

2.2. Point matching

The matching step aims at establishing the point–to–point correspondences between the two tiles of each pair given the sets of key-points and the point-wise feature description. It follows the steps below, performed independently on every pair of tiles:

1. For every key-points in the first tile, the Euclidean distance between its feature vector and the feature vector of every points in the second tile is computed.
2. A correspondence is built for every key-point by selecting the point with the smallest distance in feature space, arguably implying the highest geometric similarity, as corresponding point.

Note that contrary to common matching procedures we compare key-points with all other points from the second cloud (instead of key-points to key-points). This is because when matching key-points to key-points, the detector must be able to retrieve the exact same key-points between the two point clouds of a scene. This characteristic is called detector ”repeatability” (Salti et al., 2012) and it is hard to obtain in practice, especially with ALS data because of the difference in laser sampling density and the variable viewpoints at which ALS flight lines are flown. The method proposed here mitigates the dependency on detector repeatability, at the price of obtaining a larger number of candidates and subsequently increasing the risk of outliers. The matching time is also increased but stays comparable to the points detection and description time, two to three minutes per tile in our case.

3. The set of correspondences is filtered using RANSAC algorithm (Fischler and Bolles, 1981) with the goal to remove as much outliers as possible. The method is chosen for its efficiency in processing moderately to highly contaminated datasets.

The particular RANSAC algorithm employed in this work is based on the following assumptions: in ALS, the error in vehicle position and attitude will vary in time during the flight, causing georeferencing errors to vary in the point cloud along the flight lines. This implies that the misalignment between two point clouds is not rigid and changes along the overlapping region. The amplitude of these changes is nevertheless limited within one tile (e.g., 50 by 50 meters) since this is scanned in a few seconds (at the most) by the aerial platform. During such a short time span the navigation errors are i) mainly caused by IMU noise, ii) are strongly correlated in time and iii) have a similar impact on the entire tile. This observation allows us to establish the following statement: given a set of contaminated correspondences, there exists a 6 DoF transformation applicable within the period of few seconds of a tile that approximately aligns the correct correspondences, but which does not align the incorrect correspondences (outliers). Additionally, the inliers are only approximately aligned, e.g., because of non-rigid misalignment between the two tiles and because of the discrete nature of the point cloud. This makes it necessary to use a qualitative criteria to separate inliers and outliers. Formally, given the correct 6 DoF transformation (**T** and **R**) and two points of a correspondence ($\mathbf{p}_a$ and $\mathbf{p}_b$), this correspondence is classified as an inlier if the following condition is true:

$$||\mathbf{R}\mathbf{p}_a + \mathbf{T} - \mathbf{p}_b|| < \tau. \tag{1}$$

In other words, a correspondence is classified as an inlier if the misalignment between the two points after applying the transformation is below a certain threshold τ. The value of the threshold is chosen such that it is above the expected amplitude of the non-rigid component of the misalignment at the scale of a tile. As a result, the non-rigid nature of the alignment is accounted for in the final set of correspondences (something that is not possible using ICP). The detailed version of the RANSAC algorithm is presented in Appendix B.

2.3. Dynamic network with lidar constraints

Dynamic Networks (DNs) are probabilistic graphical models useful for solving a wide class of sensor fusion problems in inertial navigation, photogrammetry and laser scanning. They are an extension of conventional geodetic networks and were first introduced in (Colomina and Blázquez, 2004; Skaloud et al., 2018). DNs share many similarities with those formulations of graph-SLAM (Simultaneous Localization and Mapping) which consider raw inertial observations, see for example (Cioffi and Scaramuzza, 2020; Campos et al., 2021).

In DNs, as in graph-SLAM, a graph is constructed where the nodes are the unknown body frame position and orientation at discrete time instants, $\Gamma^n_{b,t}$, calibration parameters (e.g., the lidar boresight, R^b_L) and inertial sensor biases. Edges encode constraints between the connected unknowns given by sensor measurements. This graph is a graphical model, known as factor-graph (Loeliger et al., 2007), of the joint probability distribution of the observed sensor measurements given the unknowns. An optimal estimator for the unknowns can be obtained by determining the values that maximize the joint likelihood of the sensor measurements.

Typically, we assume that all the random effects are zero mean and Gaussian distributed. In this setting each edge evaluates the difference between the predicted sensor measurement given the connected unknowns (via a measurement model), and the observed one. The max-likelihood estimator for the unknowns is then obtained minimizing all such differences (residuals) by solving a non-linear weighted least-squares optimization problem. Nowadays, solvers originated in the robotics community, such as (Kümmerle et al., 2011), leverage on the graph representation of the least-squares problem and exploit sparse linear algebra methods (Davis, 2006) to solve it efficiently, allowing us to post-process real-world datasets in minutes or hours at most.

In this work we extend the approach proposed in (Cucci et al., 2017b) to consider lidar correspondences, as it will be discussed in the following.

2.3.1. Lidar correspondences error model

In order to consider lidar point–to–point correspondences in DNs, it is necessary to specify the measurement model for one such correspondence. Let us assume that the algorithm presented in Sections 2.1 and 2.2 has established that a certain point p_1, found in one point cloud, corresponds to point p_2, found in an overlapping one. These points can be traced back to the original lidar measurements, corresponding to two 3D vectors v^L_1 and v^L_2, expressed in lidar frame L, acquired at t_1 and t_2. Since p_1 and p_2 are corresponding points, v^L_1 and v^L_2 should match once expressed in the navigation frame n. This can be formally expressed as follows:

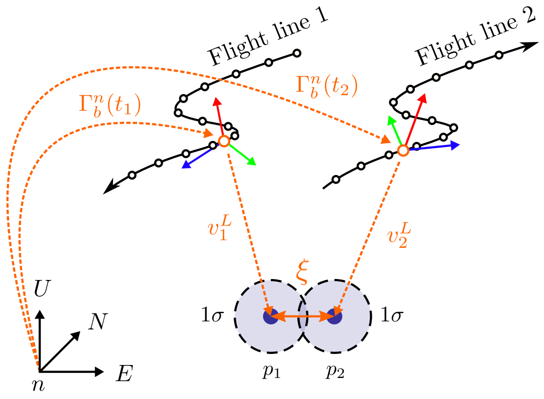

Fig. 3. Graphical representation of the geometrical constraint that can be formulated if the laser point p_1, belonging to one point cloud, has been matched with p_2, belonging to an overlapping one. The unknowns, sensor measurements, and random variables appearing in Eq. 2 are shown in orange.

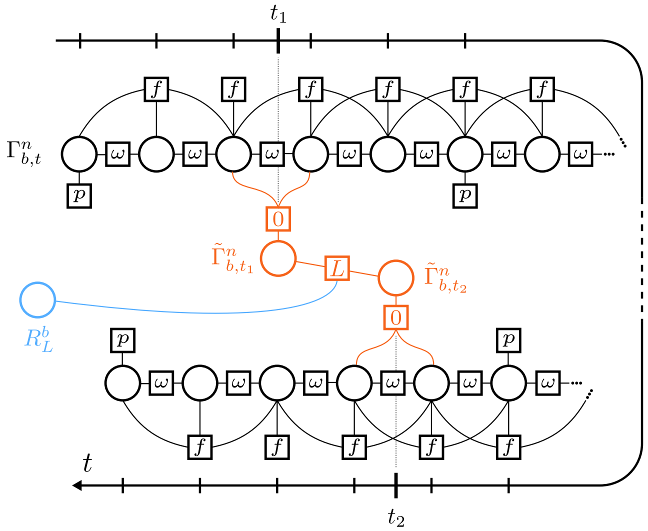
t_1
f
$\Gamma^n_{b,t}$
ω
p
0
$\widetilde{\Gamma}^n_{b,t_1}$
L
$\widetilde{\Gamma}^n_{b,t_2}$
R^b_L
t
t_2

Fig. 4. A simplified Dynamic Network in which only one correspondence edge L is shown (in orange) with the optional lidar boresight node (in blue). The nodes and edges required to handle the stream of GNSS and inertial measurements are shown in black. For simplicity the inertial sensor bias nodes are not displayed.

$$\Gamma^n_b(t_1)R^b_L v^L_1 - \Gamma^n_b(t_2)R^b_L v^L_2 = 0 + \xi, \quad \xi \sim \mathcal{N}(0, \ \sigma \mathrm{I}_{3\times 3}). \tag{2}$$

where $\Gamma^n_b(t) \in \mathrm{SE}(3)$[1] is the *continuous* time body frame position and orientation (pose) at time t and R^b_L is the lidar boresight. Here, the product operator corresponds to the usual composition of rigid transformations. ξ is a zero-mean Gaussian noise whose variance has to be adjusted depending on the point cloud density and the assumed statistics of the error in the generated correspondences. Please see Fig. 3 for a graphical representation of Eq. 2 and the discussions in Section 3.2 and 4.1 for the matching-filtering threshold and the choice of ξ, respectively.

In DNs, body frame poses are kept at discrete time instants, i.e., $\Gamma^n_{b,t}$, synchronous with the IMU. Since the rate of typical lidar sensors is substantially higher with respect to IMUs, t_1 and t_2 in general do not match the timestamp of any IMU measurement and thus $\Gamma^n_b(\cdot)$, as it appears in Eq. 2, is not available in general. Thus, we replace $\Gamma^n_b(t_1)$ and $\Gamma^n_b(t_2)$ with two new poses, $\widetilde{\Gamma}^n_{b,t_1}$ and $\widetilde{\Gamma}^n_{b,t_2}$, which are constrained to lie on the geodesic in SE(3) between the nearest poses available in the DN. This approach is described in detail in (Cucci et al., 2017b, Section 4.2), where it was employed to handle image observations at arbitrary timestamps, and in (Ceriani et al., 2015) in the context of mobile laser scanning. Eq. 2 thus becomes

$$\widetilde{\Gamma}^n_{b,t_1} R^b_L v^L_1 - \widetilde{\Gamma}^n_{b,t_2} R^b_L v^L_2 \qquad = R^b_L v^L_1 - \underbrace{\left[\widetilde{\Gamma}^n_{b,t_1}\right]^{-1} \widetilde{\Gamma}^n_{b,t_2}}_{\Gamma^{b,t_1}_{b,t_2}} R^b_L v^L_2 = 0 + \xi, \tag{3}$$

where in the second line it has been rearranged to show that the introduced measurement model only constrains the *relative* body frame position and orientation between t_1 and t_2. Eq. 3 is formulated for each couple of matched points and builds an edge in the DN graph for each point–to–point correspondence, as it will be discussed in the following.

2.3.2. *Dynamic network structure*

The resulting DN structure considered in this work is presented in Fig. 4, where the circles represent unknowns, and the squares represent measurement edges, respectively. Five types of edges are considered:

1 p: GNSS position measurements,

2–3 ω and f: angular velocity and specific force edges. Here, IMU preintegration (Lupton and Sukkarieh, 2011) is employed in order to reduce the overall number of unknowns in the graph without loss of information. The exact formulation is presented in (Cucci and Skaloud, 2019), which extend classical IMU pre-integration approaches to consider a gravity model and the Earth rotation, thus removing limitations on the size of the mapped area while making it also applicable to inertial systems of higher quality. The inertial sensor biases and the related stochastic models are handled as in (Cucci et al., 2017a).

4 0: *zero*-observation edges (edges for which no actual sensor measurement exists) that constrain $\widetilde{\Gamma}^n_{b,\cdot}$ to lie on the geodesic in SE(3) connecting adjacent, IMU synchronous, poses. Please refer to (Cucci et al., 2017b, Section 4.2).

5 L: lidar correspondences, the error model being defined in Eq. 3. L edges connect two poses nodes, corresponding to $\widetilde{\Gamma}^n_{b,t_1}$ and $\widetilde{\Gamma}^n_{b,t_2}$ and, optionally, the lidar boresight node.

For further details, the reader is invited to refer to the referenced publications and to the released source code, see Section 3.7 below.

It is worth nothing that the actual 3D coordinates of the lidar point in navigation frame do not appear as unknowns in the proposed DN, c.f., Fig. 4, resulting in reduced computational complexity. This is different with respect to classical image-to-image (or point–to–point) correspondences commonly employed in bundle adjustment or in visual SLAM where an additional unknown is considered for all observations referring to the same point in object space.

3. Experiments

In this section, we describe the particular setup of the data acquisition and details of the data processing. We also present experiments that were conducted with the aim of highlighting the limitations of trajectory determination using lightweight IMUs via Kalman filtering and optimal Smoothing (KS)[2], as well as their impact on the georeferencing of the measured point cloud. Moreover, we seek to demonstrate the potential benefit of employing the proposed point–to–point lidar correspondences together with raw navigation in dynamic networks.

[1] The special Euclidean group of dimension 3, $\mathrm{SE}(3) = \mathrm{SO}(3) \times \mathbb{R}^3$.

[2] More precisely, with the term “Kalman smoother” we refer to the Rauch-Tung-Striebel smoother (Rauch et al., 1965), abbreviated with “KS” in the following for the sake of brevity.

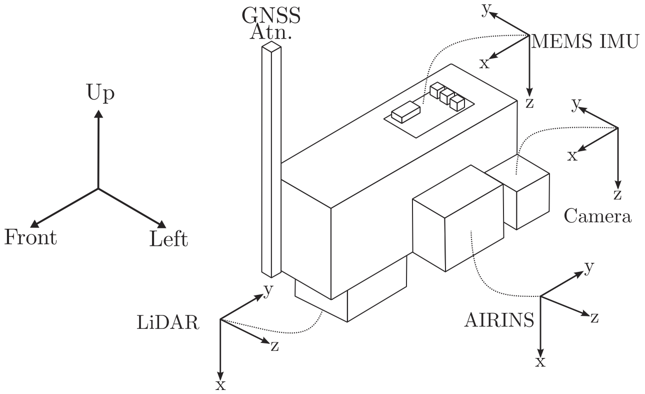

Fig. 5. Illustration of the mounting used to capture the data.

Table 1
Manufacturer or confirmed specifications (Vallet et al., 2020; Clausen and Skaloud, 2020).

IMUs	Gyro drift [°/h]	Accel. drift [mg]	RP/Y acc. (PPK) [°]	XYZ acc. (PPK) [m]
AIRINS	<0.01	<0.25	0.002/0.005	ref
Navchip	20	2	0.03/0.18	<0.03
Lidar	Beam divergence [mrad]	Point meas. rate [kHz]	Scan rate [Hz]	Return mode
VQ-480	0.27	200	100	multi

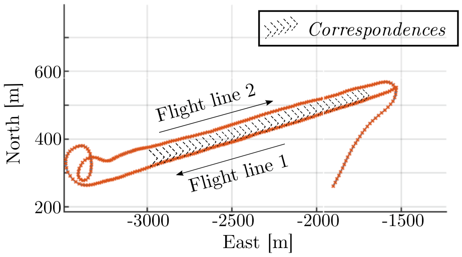

Fig. 6. Flight lines and position of the overlapping areas where correspondences were established.

3.1. Data acquisition

For the experiments, we use the airborne dataset presented in (Vallet et al., 2020). There, reference and lower-cost sensors (lidars, IMUs and cameras) were rigidly mounted together on a vibration dampened assembly and installed in a helicopter. The common mounting provides the same flight conditions for all sensors (temperature, dynamics, height of flight, GNSS availability). In our experiments, we consider only part of those sensors, namely the reference lidar and two (reference and low-cost) IMUs. The reference lidar is a medium range VQ-480 (Riegl). The reference IMU is a navigation grade AIRINS (iXblue) while the small MEMS-IMU is a Navchip v1 (Thales), the performance of which is similar to popular commercial UAV-grade INSs such as the APX15 (Applanix) (see Vallet et al. (2020) or Clausen and Skaloud (2020) for a detailed evaluation). A scheme of the experimental mounting is shown in Fig. 5 and specifications of all sensors are presented in Table 1.

During the flight, the helicopter flew profiles close to the ground to mimic a typical UAV flying altitude and at a speed similar to that of a small multi-copter (around 12 m/s) over an area featuring various terrain types including urban and rural areas, forest, croplands, roads, railroads and power lines. We focus on two successive flight lines depicted in Fig. 6 that are approximately 2 km long, for a total flight time of around 6 min. The overlapping sections where correspondences can be established are depicted in the same figure. The swath width of each flight line is about 180 m and the side overlap is close to 40%. It is important to mention that the reference lidar is not a small, UAV grade sensor and thus generates a higher point cloud density when flown at a typical UAV flight altitude. To mitigate this aspect, the helicopter was flown at 230 meters above ground level (AGL) while small UAV-based lidars are generally flown below 180 m AGL (see for example mini-VUX (Riegl, 2021)). This results in point clouds with ground densities varying between 35 pts/m^2 and 50 pts/m^2 on bare ground. Similar point cloud densities could be obtained by flying a UAV at a slightly lower speed (e.g., 10 m/s) and between 100 m and 160 m AGL. This density corresponds to a point cloud GSD (average distance between one point and its closest neighbor) of between 10 cm and 20 cm.

Thanks to this particular setup, it is possible to obtain a reference dataset that is based on a carrier phase differential post-processed (PPK) trajectory generated with the AIRINS coupled with the medium range lidar. The trajectory integrated with this kind of IMU shows attitude errors typically one order of magnitude lower than those based on MEMS-IMUs, e.g., $<0.003°$ see Vallet et al. (2020). This ensures low errors in the subsequent georeferenced point cloud ($\leqslant$5 cm). This dataset was considered as the ground-truth reference for our experiments due to its high accuracy.

A “UAV-grade” data set was also generated with the setup described above. This was accomplished by employing the small MEMS-IMU (as is used on UAVs) to estimate the trajectory used for the subsequent georeferencing of laser observations. Albeit based on a larger lidar, the newly introduced line of UAV-targeted laser scanners from the same manufacturer (miniVUX) is reported to have ranging errors within PPK noise level (i.e. < 2–3 cm) at up to ~250 m AGL (Riegl, 2021). Further, as the attitude quality based on pre-calibrated NavChip data is similar to that of APX15 (Vallet et al., 2020) used on such scanners (Riegl, 2021), we obtain a representative “UAV-grade” point cloud in terms of geometrical accuracy through applying the standard georeferencing procedure. This quality is then compared to that obtained via the proposed methodology.

3.2. Data processing

In all cases the absolute position and velocity observations are obtained via PPK from Javad multi-constellation and multi-frequency receivers (rover and master) using GrafNav (Novatel) software. Lever-arms between all sensors on the system are determined within 1 cm following the laboratory calibration approach of Vallet and Skaloud (2004). The processing steps for the standard approach and the dynamic network approach with correspondences are summarized in Fig. 1a and Fig. 1b, respectively. For the sake of clarity we refer to the standard trajectory estimation approach based on the optimal Kalman smoother as “KS“ and to the dynamic network approach as“DN” or “DN + C”, the latter being the DN which considers lidar correspondences (“C”) as additional observations.

In the KS approach, we generate trajectories via an optimal smoothing to integrate IMU readings in a loosely coupled manner with vehicle PPK position and velocity. We use the APPS software (iXblue) when integrating AIRINS reference data and Posproc (Applanix) with internally designed filters for the MEMS-IMU data. The lidar’s boresight matrix is estimated with Riprocess (Riegl), which produces practically the same value as estimated with LIBOR (Skaloud and Schaer, 2007). These parameters are estimated using separate flight data and reused for every dataset (with the exception of Case 3). The boresight between the AIRINS and the MEMS-IMU is determined as a mean value that minimizes the attitude differences between the two trajectories obtained by applying the Kalman filtering and optimal smoothing to data from both IMUs over almost 1 h. While the boresight determined this way may be slightly incorrect due to residual misalignments in the trajectory determined with the MEMS-IMU, it ensures the best results for all “KS” trajectories in later comparisons. We thus stay on the “safe side” when comparing reference methodologies with the proposed one. Finally, the point cloud is georeferenced based on the previously estimated trajectory and mounting parameters using an in-house software, LIEO (Skaloud, 2017).

In the DN + C approach (Fig. 1b), the two first steps are common to the KS approach, where an approximate trajectory and subsequent point cloud are obtained using a Kalman smoother (or using a dynamic network without lidar correspondences). We then establish correspondences as detailed in Sec. 2.1 and 2.2. For the preprocessing, a size of 50 by 50 meters was used to tile the point clouds. For the ISS key point detection method, point neighborhoods are extracted with a support radius of 1 m and key points are selected using an eigenvalue ratio of 0.5, which enables the extraction of key-points from a variety of elements (man made and vegetation). For the LCD key-point description method, point neighborhoods are extracted with a support radius of 2 meters to include typical terrain features (e.g., building sections, cars, etc.) that lie within the neighborhood of the described points. A threshold of 25 cm (1.2–2.5× GSD) was selected for the RANSAC algorithm. This threshold is important to perform non rigid matching and has been selected in consideration of the dynamic of the flight, the GSD and the size of the point cloud tiles. As will be shown later, such a threshold is suitable for the dynamic of standard UAV flights as well as for challenging scenarios such as during GNSS outages or in the case of miscalibrated boresight matricies, where trajectory estimation is less accurate and larger misalignments occur. It is important to note that the proposed approach is permeable to variations in the choice of the threshold to some extent. Similar behavior in the corrections were observed when taking a threshold twice as large, of 50 cm (corresponding to 2.4–5× GSD) by adapting the corresponding weighting in the Dynamic Network, reflected by the parameter σ in Eq. 2. On the other hand, taking a smaller threshold might lead to a "pseudo rigid" selection of the correspondences. In this case the the Dynamic Network will still be able to correct with those correspondences but might show patterns on certain tiles, when the non-rigid component of the misalignment is not brought into the correspondences due to a too severe pruning by RANSAC.

Once 3D point–to–point lidar correspondences are generated, they are processed in a single pass of the DN along with raw inertial observations (angular rate and specific force), and GNSS positions. After such "DN + C" adjustment, the point cloud is georeferenced from scratch based on the adjusted trajectory in LIEO for comparison with the reference. As noted above, trajectories can also be estimated in the dynamic network without using lidar correspondences ("DN"). We investigate the proposed methodology within the following use cases and scenarios.

Table 2
Overview of evaluated trajectories and their parameters.

		Trajectory Name	Est. Method	% Corresp.	Known bore.	GPS outage
CASE 1	Nominal	T.1 KS	KS	n.a.	✓	✗
		T.2 DNC	DN	100%	✓	✗
		T.3 DN	DN	0%	✓	✗
		T.4 DNC (from dn)	DN	100%	✓	✗
CASE 2	Nominal with min. constrain	~T.4 DNC (from dn)	DN	100% →0.1%	✓	✗
CASE 3	Boresight calibration	T.5 KSB	KS	n.a.	✗	✗
		T.6 DNCB	DN	100%	✗	✗
CASE 4	GNSS outage	T.7 KSo	KS	n.a.	✓	✓
		T.8 DNCo	DN	100%	✓	✓
		T.9 DNCo1	DN	100%	✓	✓ (single FL)
		T.10 DNo	DN	0%	✓	✓

3.3. Case 1: nominal (optimal GNSS reception)

The trajectories described in this section are generated under optimal GNSS signal reception to serve as reference trajectories and point clouds obtained in ideal flight conditions. We define five scenarios:

- Trajectory 0 R: reference trajectory, employing navigation grade INS and KS approach. All other trajectories and point clouds will be compared to this "ground-truth" dataset to estimate their respective accuracy.
- Trajectory 1 KS: employing MEMS-IMU (as that employed in UAVs) and KS approach.
- Trajectory 2 DNC: employing MEMS-IMU (as that employed in UAVs) and DN + C approach. This dataset is obtained following the methodology depicted in Fig. 1b. We establish correspondences on the UAV grade point cloud from trajectory 1 in the area depicted in Fig. 6. In total, $\sim 62'000$ correspondences are generated and used to constrain the adjusted trajectory.
- Trajectory 3 DN: employing MEMS-IMU (as that employed in UAVs) and DN approach (without correspondences). We produce this trajectory to be able to assess separately the impact of the lidar correspondences and the impact of the DN itself compared to a Kalman smoother.
- Trajectory 4 DNC (from DN): employing the same procedure as trajectory 2 DNC except that correspondences are generated from point clouds from trajectory 3 DN instead of trajectory 1 KS. The purpose of this setup is to show that the proposed approach does not rely on standard KS to generate the approximate trajectory and point cloud and that the DN can be used at both trajectory estimation steps.

3.4. Case 2: nominal with reduced set of point–to–point correspondences

In this case we aim to assess the performance of the proposed procedure with a sparser set of lidar correspondences. This is to consider mapping an area with fewer geometric features to match, for example in a built up area with bare ground and without vegetation. To do so, we down-sample the set of point–to–point lidar correspondences used in T.2 DNC from 50% to 0.1% of its initial size. We finally estimate a trajectory and subsequent point clouds for each previously described scenario and estimate the respective attitude and georeferencing errors.

3.5. Case 3: Boresight self-calibration

This scenario aims at assessing the ability of the method to calibrate the lidar boresight, R_L^b, "in one go" along with the trajectory determination, despite the sub-optimal geometry (two opposite flight lines at the same altitude). The following trajectories and subsequent point clouds will be compared:

- Trajectory 5 KSB: employing the same technique as trajectory 1 KS, but without using the pre-calibrated boresight. To do so, each boresight angle is set to 0° when registering the point cloud in LIEO.
- Trajectory 6 DNB: employing the same technique as trajectory 2 DNC, but without using the pre-calibrated boresight. The boresight is considered as a supplementary unknown parameter to estimate by the DN with an initial value of 0°. Please see Fig. 4, in blue.

3.6. Case 4: GNSS outage

In this case we are interested to assess the ability of the proposed method (DN + C) to generate trajectories and point clouds of a higher accuracy under challenging scenarios such as during a GNSS outage.

- Trajectory 7 KSo: employing the same technique as trajectory 1 KS, but with a simulated GNSS outage impacting the two flight lines during ∼ 60 seconds each (2 min in total). This corresponds roughly to a 750 m long area without GNSS signal reception that could occur for example when operating in obstructed corridors, urban canyons or due to radio signal interference, or possibly jamming.

- Trajectory 8 DNCo: employing the same technique as trajectory 2 DNC, but with the same simulated GNSS denied area as trajectory 7 KSo.
- Trajectory 9 DNCo1: employing the same technique as trajectory 2 DNC, but with a GNSS outage impacting only one of the two flight lines. This is to simulate a perturbation due to close physical obstruction (building, mountain) or a temporal electromagnetic disturbance for example. This scenario is considerably different from trajectory 8 DNCo because for all correspondences, only one of the 2 points is affected by the outage.
- Trajectory 10 DNo: employing the same technique as trajectory 3 DN, but with the simulated GNSS outage of 60 s over the two flight lines. We consider this case to assess the performances of DN (without correspondences) with respect to KS.

Table 2 summarizes the different scenarios of trajectory estimation and their respective parameters.

3.7. Reference implementation

We release a reference implementation of the DN solver employed in this work. The code to compute all “DN” trajectories referenced in Table 2, along with the necessary data, is available at[3]. This implementation is based on the ROAMFREE sensor fusion framework (Cucci and Matteucci, 2014b; Cucci and Matteucci, 2014a), a general purpose, open-source dynamic network solver built on the top of the widely employed g^{2}o least-squares solver (Kümmerle et al., 2011) and sparse linear algebra libraries (Davis, 2006), allowing the adjustment of, for example, trajectory 2 DNC, within a few minutes on a standard desktop computer.

4. Results and analysis

The dataset generated with navigation grade INS/PPK and precise airborne scanner (T.0 R) serves as the ground-truth reference with respect to which all other generated datasets are compared. It is important to keep in mind that the same lidar measurements are used in every dataset but the IMUs and methods used in georeferencing vary. This means that point clouds registered using different trajectories contain exactly the same number of points and that, for each point in these point clouds, the corresponding point in the ground-truth can be (easily) recovered. Thus, the “mapping” quality can be evaluated by comparing the difference between the coordinates of each point with the reference.

4.1. Case 1: nominal (optimal GNSS reception)

Correspondences: Correspondences are established following the methodology from Sections 2.1 and 2.2 (Fig. 2) over an area depicted in Fig. 6 independently on two datasets. First, on the UAV-grade trajectory T.1 KS (to be used in T.2 DNC) and second, on the T.3 DN (to be used in T.4 DNC (from DN)).

In both cases, the overlapping area is split into 91 tiles of 50 by 50 meters. Before filtering, between 200 and 1200 correspondences are retrieved per tile in built-up zones, while for tiles with vegetation between 1’000 and 6’000 correspondences are retrieved. This is due to the higher number of key points identified by ISS in vegetation. Overall, ∼50’000 correspondences are kept after filtering. We trace the two correspondence sets in the ground-truth point cloud to estimate the correspondence quality and estimate the error distribution as shown in Fig. 7.

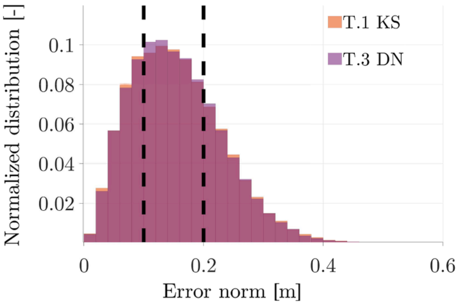

Fig. 7. Correspondence normalized error distribution. The actual interval of point cloud GSD is depicted in dashed black lines.

First of all, we can see that the error distribution is almost identical in the two cases which means that the methodology for establishing the correspondences is not particularly dependent on method for the initial trajectory estimation. We observe that, after outlier rejection, 89 of the 91 tiles have correspondences discrepancies below 20 cm (∼ 1–2× point cloud GSD), which we consider as a successful run of the RANSAC algorithm. In particular, most of the correspondences have an error between 10 cm to 20 cm. This is mostly due to the GSD of the point cloud and the limited characterization of LCD that can not efficiently separate points that are very close when projected in feature space (due to high similarity of their neighborhood). In other words, the mean fit is directly related to the point cloud GSD (in our case 0.1–0.2 m). Although increasing the sampling density could possibly allow detecting closer correspondences and thus obtaining stronger constraints between trajectories, we will demonstrate later that it is not necessary. In both cases, the mean error of retrieved correspondences is 15.6 cm with a 1σ dispersion of 7.6 cm. As 15 cm corresponds to the average point cloud GSD, this confirms the suitability of choosing the GSD as a 1σ value to weight correspondence constraints in the dynamic network (Fig. 4, right and Eq. 2).

Trajectory: To estimate the accuracy of the trajectory, we compute the error in position (East, North, Up) and attitude (Roll, Pitch, Yaw), with respect to the reference. The impact of correspondences on the *position* correction is expected to be limited within the nominal (no GNSS outage) scenario compared to the effect on *attitude*. This is because, under such conditions, and with well calibrated system parameters, the main source of error affecting the point cloud georeferencing is the attitude obtained from small MEMS-IMUs (Glennie, 2007). Indeed, the error in vehicle position with correctly determined differential carrier-phase ambiguities (PPK) is lower than the precision of correspondences.

We can see in Table 3 that the DN estimates the position slightly better than the Kalman smoother with respect to the reference. Overall, the position mean error between T.1 and T.3 appears to be reduced by a factor of 1.7 (from 2.3 cm to 1.4 cm) but the respective differences are close to the reference precision. Hence, the inclusion of correspondences have practically negligible influence on the airborne position for both T.2 and T.4 against T.3. This is somewhat expected since, as explained before, for the given GSD, the mean accuracy of correspondences is worse than vehicle position under PPK/INS.

However, Fig. 8 shows that using correspondences as input to the DN leads to a significant reduction of the error for all three attitude angles. Comparing the standard Kalman smoother (T.1) to the proposed approach (T.2), the last three columns of Table 3 show that the smallest correction occurs in the roll angle, with a 1.6× reduction of the RMSE (from 0.037° to 0.023°). This relatively small improvement is related to the fact that the absolute magnitude of the roll error is smaller compared to that of pitch and yaw, leaving less room for improvement. The improvements on pitch and yaw are more substantial. Pitch RMSE is reduced from 0.060° to 0.021° while Yaw RMSE is reduced from 0.190

[3] https://github.com/TOPO-EPFL/DN-LiDAR

Table 3
Position and attitude relative errors for trajectories n°1, 2, 3 and 4 with respect to the reference trajectory (T.0).

Traj.	**Position Error [m]**					**Attitude Error [°]**		
	East	North	Up	Norm		Roll	Pitch	Yaw
	RMSE			Mean	Std	RMSE		
T.1 KS	0.016	0.016	0.017	0.023	0.017	0.037	0.060	0.190
T.2 DNC	0.008	0.010	0.011	0.014	0.010	0.023	0.021	0.051
T.3 DN	0.007	0.009	0.014	0.014	0.010	0.073	0.053	0.150
T.4 DNC (from DN)	0.008	0.010	0.011	0.014	0.009	0.023	0.021	0.046

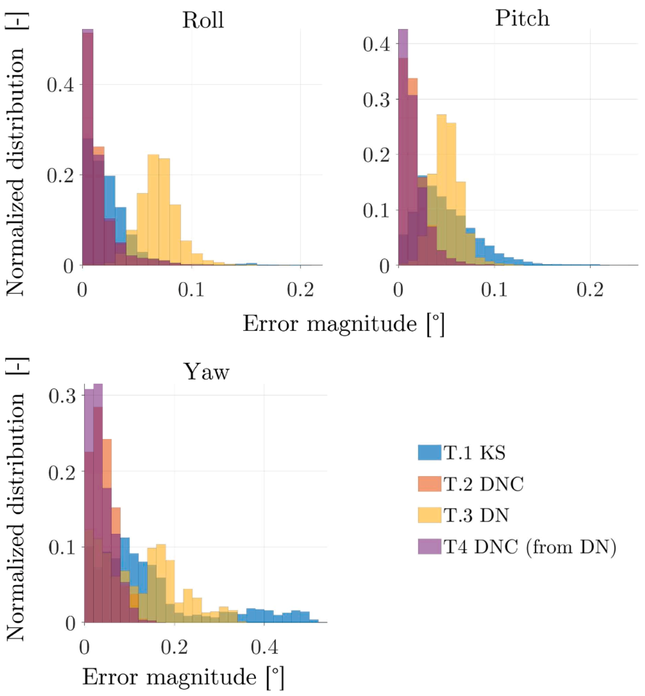

Fig. 8. Normalized distribution of attitude errors in 4 trajectories.

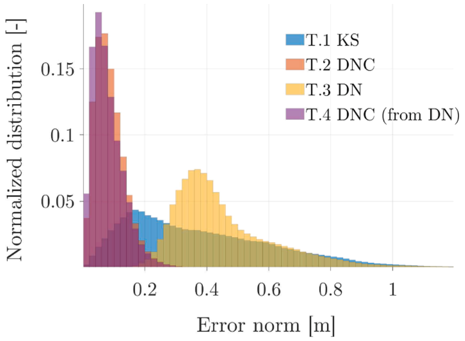

Fig. 9. Normalized distribution of point cloud georeferencing error in magnitude for the four trajectories.

Table 4
Point cloud georeferencing error in the 1st flight line.

Traj.	RMSE [m]			Norm [m]	
	East	North	Up	Mean	Std
T.1 KS	0.428	0.120	0.034	0.379	0.234
T.2 DNC	0.079	0.058	0.018	0.087	0.049
T.3 DN	0.310	0.361	0.094	0.455	0.166
T.4 DNC (from DN)	0.072	0.058	0.018	0.080	0.049

to 0.051; a reduction factor of 3× and 3.7× respectively. Considering the fact that attitude quality improvement is proportional to the square (or even the cube) of the IMU size and weight (as well as cost), such improvement is substantial. Also, the ability to correct angles with larger initial errors is critical as they are responsible for most of the georeferencing errors (in the nominal mapping scenario with optimal GNSS signal reception). In this particular geometry, those larger angular errors cause the greatest discrepancies between the two flight lines and become observable with the inclusion of the 3D-lidar point–to–point correspondences. Pitch and yaw errors are observable along track (long overlapping area) while roll errors are visible across track (short overlapping area). Comparing T.2 and T.4 (i.e., the two DNC approaches based either on KS or DN to generate approximate trajectory and point clouds), we can see both reaching the same level of agreement with respect to the reference, despite their respective differences in initial trajectory errors and initial point cloud misalignment within the overlapping section. In this respect the error reduction is higher between both trajectories from the dynamic network (T.3 vs. T.4 as purple vs. yellow in Fig. 8), which demonstrates that the use of correspondences provides the improvements in attitude estimation in this framework even with less complex inertial error modeling (e.g., with respect to Posproc) (which is the primary cause of attitude differences between T.1 and T.3 with respect to the reference).

Point cloud: As we have observed that the attitude estimation is greatly improved by the use of lidar correspondences within the dynamic network with respect to T.1, we expect the DN + C trajectory to deliver a point cloud of better quality than the one from the Kalman smoother. To verify this, we estimate the point-wise georeferencing error by estimating for every point, the distance with respect to the corresponding point in the reference point cloud.

The results are visible in Fig. 9. On flight line 1, when using the standard KS approach (T.1 KS, in blue), the maximum georeferencing error is about 1.1 m and most of the points have errors between 15 cm and 50 cm (1× to 3× GSD). These errors are significantly reduced when using the DNC methodology. Considering T.2 DNC, most of the points are georeferenced within 5 cm to 15 cm (0.3× to 1× GSD). The maximum georeferencing error is also greatly reduced, to 28 cm in this flight line. Comparing the two DNC cases (T.2 and T.4), georeferencing errors are almost identical due to similar trajectory improvements (see Table 3). Finally, the errors are much larger using the dynamic network without correspondences (T.3 DN, yellow), which demonstrates that the reduction in georeferencing error is due to the use of correspondences. As shown in Table 4, the mean error using the standard Kalman smoother approach (T.1 KS) is 38 cm and is decreased to 8 cm using the proposed approach (T.4 DNC), corresponding to a reduction by a factor of ~5, from 2.5× to 0.6× GSD. This improvement is mostly in planimetry, as shown by the RMSE per axis (East/North columns in Table 4). Please note that the reference point cloud, obtained with a navigation grade IMU, is assumed to be accurate up to 5 cm.

It is important to keep in mind that the magnitude of the georeferencing errors is proportional to the flight altitude, in our case 230 m AGL. Flying at a lower altitude and with less powerful and smaller scanners intended for use with UAVs, i.e., miniVUX1-3, Riegl (2021) at ~150 m AGL, would allow the registration refinement to fully benefit

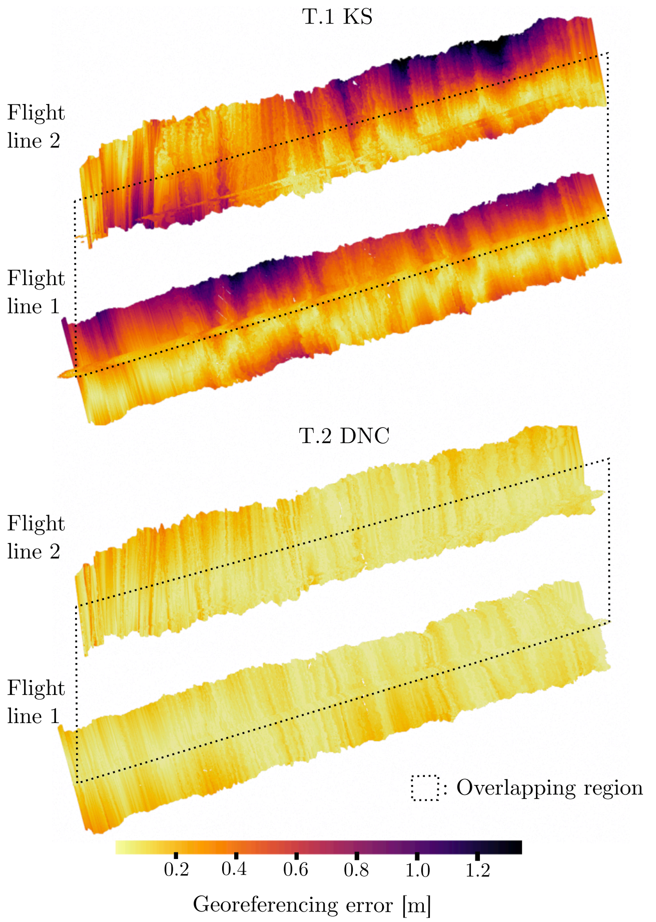

Fig. 10. Magnitude of error in points coordinates based on trajectory KS T.1 (top) and T.2 DNC (bottom).

from attitude improvement and the desired point cloud accuracy of 5 cm could be obtained.

Fig. 10 shows georeferencing errors at the scale of the point cloud by coloring each point according to its spatial distance with respect to the reference. The point clouds in the upper part of this figure are georeferenced using T.1 that is based on the KS. As expected, larger errors are observed at the edges of the scan, due to the distance from the vehicle. While the error is below 30 cm at the center of the point cloud, it reaches 1.2 meters in the edges. The irregular patterns in the error are due to the orientation dynamic of the flight causing the rapid fluctuations in attitude error projection in the mapping frame (which are difficult, if not impossible to correct with ICP). The improvements are very clear when comparing these errors to the bottom part of the figure, which is based on DN + C trajectory (T.2). Here, the georeferencing errors are considerably reduced and most of the points have a georeferencing errors lower than 15 cm.

Summary: This practical example demonstrates that within the nominal mapping scenario (i.e., stable GNSS signal reception) using a small MEMS-IMU, incorporating point–to–point correspondences into the dynamic network significantly improves the quality of the computed trajectory and in particular the three attitude angles. In turn, this ameliorated estimation of the vehicle attitude results in residual point cloud registration errors that are lower than the GSD; a significant improvement compared to the direct georeferencing approach where the errors are ~5× larger.

4.2. Case 2: nominal with reduced set of point–to–point correspondences

Correspondences: In this situation we use the same set of correspondences as in T.4 DNC (from DN), except that we downsample them, or in other words we use a small sub-set of the original correspondence set in subsequent adjustments. Thus their quality is the same as depicted in Fig. 7, but fewer are used in the adjustment. In particular, we down-sample progressively from 50% to 0.1% of the original set of correspondences and estimate both trajectories with their respective point clouds for each step.

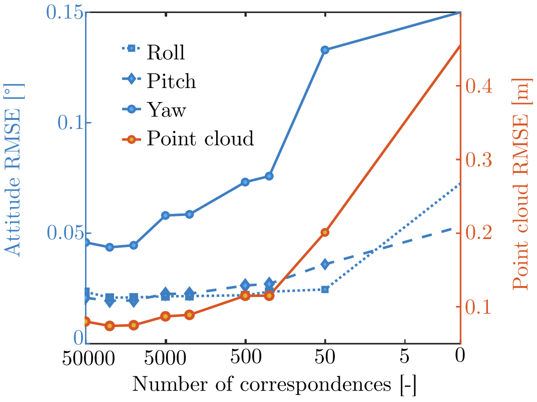

Fig. 11. Effect of correspondences down-sampling on attitude estimation and subsequent point cloud regeneration.

Impact: Fig. 11 illustrates the effect of down-sampling correspondences (x-axis, in logarithmic scale) on the attitude RMSE (left y-axis, in blue) and on the RMSE of the registered point cloud (right y-axis, in red). Yaw angle is the first angle affected by the down-sampling since its RMSE increase significantly when down-sampling more than 25% (keeping ~12’500 correspondences). On the other hand, the estimation of roll and pitch is less sensitive to down-sampling, their RMSE being stable up to down-sampling to 0.1% (keeping only ~50 correspondences) for roll and 0.5% (~250 correspondences) for pitch. Due to the geometry of nadir ALS, laser beams are mostly projected along the vertical axis. While a perturbation in roll and pitch will impact more its vertical component, a perturbation in yaw affects more the horizontal component towards the swath extremities. For this reason, despite being more influenced by down-sampling as compared to other angles, yaw error has generally smaller effect on point cloud georeferencing than pitch and roll. The stability in roll and pitch estimation allows the point cloud to be accurately registerd even with strong correspondence down-sampling. In fact, the point cloud RMSE is kept below 10 cm when down-sampling at 5% (~2500 correspondences), and is still remarkably stable up to a down sampling at 0.5% (~250 correspondences) which results in an increase of the georeferencing RMSE by a factor of 1.4 × .

Summary: This example illustrates practically the ability of the proposed method to correctly adjust the trajectory even with a considerably smaller number of correspondences. This can be practical when scanning over an areas with limited distinctive features (e.g., crops with sparse and low vegetation) where establishing correspondences is more

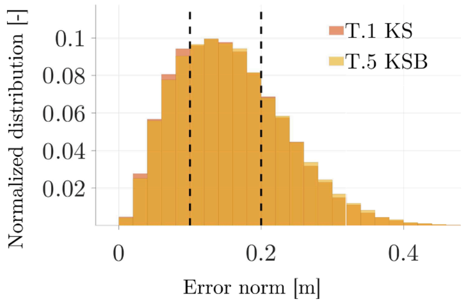

Fig. 12. Correspondences’ normalized error distribution. The actual interval of point cloud GSD is depicted in dashed black lines.

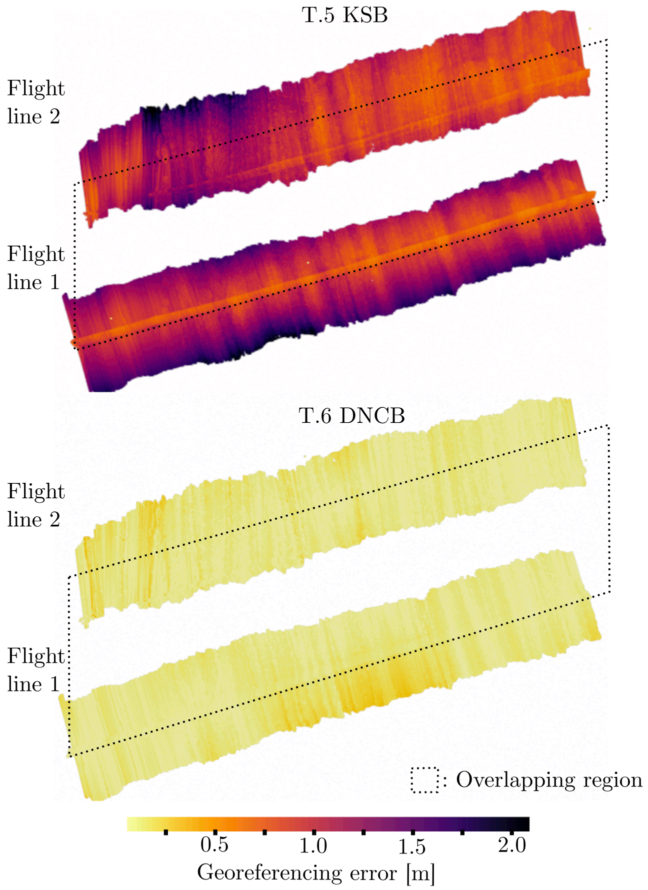

Fig. 13. Points georeferencing error with unknown (top) end recovered boresight (bottom) via constrained dynamic network.

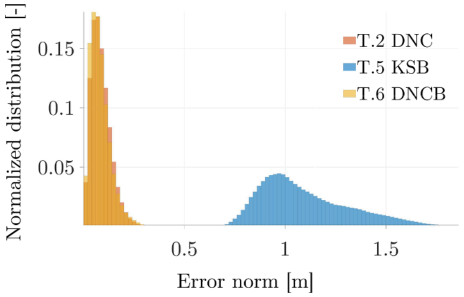

Fig. 14. Normalized distribution of point cloud georeferencing error in magnitude for trajectories T.2, T.5 and T.6.

challenging.

4.3. Case 3: Boresight self-calibration

Correspondences: Correspondences are established on the point cloud registered using the KS trajectory with unknown (zero) boresight (T.5 KSB). This dataset shows significantly higher misalignment between the point clouds due to the absence of the boresight parameters (e.g., between 1 and 3.5 meters in the overlapping section, in the top part of Fig. 13). We first compare the agreement of established lidar correspondences with those obtained in the “boresight-perfect” dataset from T.1 KS. As shown in Fig. 12, the histogram of the two error distributions overlaps, indicating that the quality of the correspondences is almost identical in both cases. This confirms the translation invariance of the LCD descriptor and the ability of the method to establish sufficiently accurate correspondences even with a poor initial registration (i.e., between two strongly misaligned point clouds).

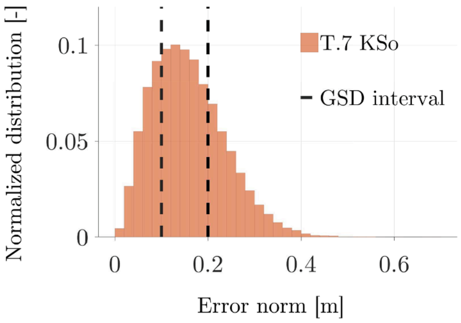

Fig. 15. Normalized error distribution of established correspondences. The actual interval of point cloud GSD is depicted by dashed black lines.

The trajectory is then re-estimated along with the three lidar boresight angles by the dynamic network with the set recovered of correspondence. Both outputs (corrected trajectory and the estimated boresight) are finally used as input to LIEO to refine the registration of the point clouds.

Point cloud: The quality of the point cloud registration is depicted in Fig. 14 as a histogram of the georeferencing error (point-wise magnitude) for T.5 KSB (blue) and T.6 DNCB (yellow). For the sake of completeness, the same figure displays the error histogram for the reference point cloud based on T.2 DNC (orange) as well.

We first observe the effect of the absence of boresight parameters on the point cloud (T.5 KSB, blue). This point cloud shows large georeferencing errors ranging from 0.7 m to 1.8 m and a mean error of 1.1 m (e.g., compared to 0.38 cm with the calibrated boresight, T.1 KS). On the other hand, we can see that, when coupled with correspondences, the dynamic network is able to estimate the boresight matrix as well as correct the systematic causes for the errors in the trajectory at the same time: as shown in bottom part of Fig. 13, the resulting point cloud based on T.6 DNCB is well aligned. Indeed, the mean error is approximately 8 cm, which is about 13× smaller that that of the point cloud based on T.5 KSB.

The boresight estimated by the dynamic network is slightly different from that obtained via the calibration flight using planar constraints (using Riprocess, Riegl), with values per angle of [-0.276; 0.049; 0.154] and [-0.213; 0.010; 0.191], respectively. This difference is mainly due to the fact that boresight angles and IMU attitude errors (caused by imperfect initialization of orientation that is, among others, related to the unknown biases of the inertial sensors) have similar effect on the trajectory, at least within this scenario. Since the dynamic network estimates those simultaneously, the change in boresight angles is compensated with IMU biases, leading to a resulting point cloud registration that is very similar to T.2 DNC.

Summary: In this case we first showed practically that the proposed approach to estimated correspondences is robust to large misalignment in the approximate point cloud. Second, we demonstrated the ability of the “DNC” approach to correct for point cloud deformations due to unknown mounting parameters, in this case the three boresight angles, without any loss in accuracy in the final georeferenced point cloud.

4.4. Case 4: GNSS outage

Correspondences: Correspondences are established on the point cloud generated from the KS trajectory with a relatively long (for a low-cost MEMS-IMU) GNSS position outage of 60 s (T.7 KSo). We have seen in Section 5.1 that the quality of correspondences is not necessarily impacted when using a point cloud georeferenced with a highly sub-optimal trajectory. This is again verified for a trajectory affected by a 2 min-long absence of GNSS position measurements. As can be seen in

Table 5
Position and attitude errors in trajectories T.7 to T.10 during the outage on line 1 with respect to the reference.

Traj.	**Position Error [m]**					**Attitude Error [°]**		
	East	North	Up	Norm		Roll	Pitch	Yaw
	RMSE			Mean	Std	RMSE		
T.7 KSo	0.560	0.282	0.123	0.566	0.297	0.051	0.083	0.665
T.8 DNCo	0.206	0.297	0.040	0.299	0.202	0.018	0.024	0.102
T.9 DNCo1	0.133	0.033	0.011	0.118	0.071	0.012	0.017	0.064
T.10 DNo	0.650	0.403	0.024	0.630	0.434	0.050	0.068	0.341

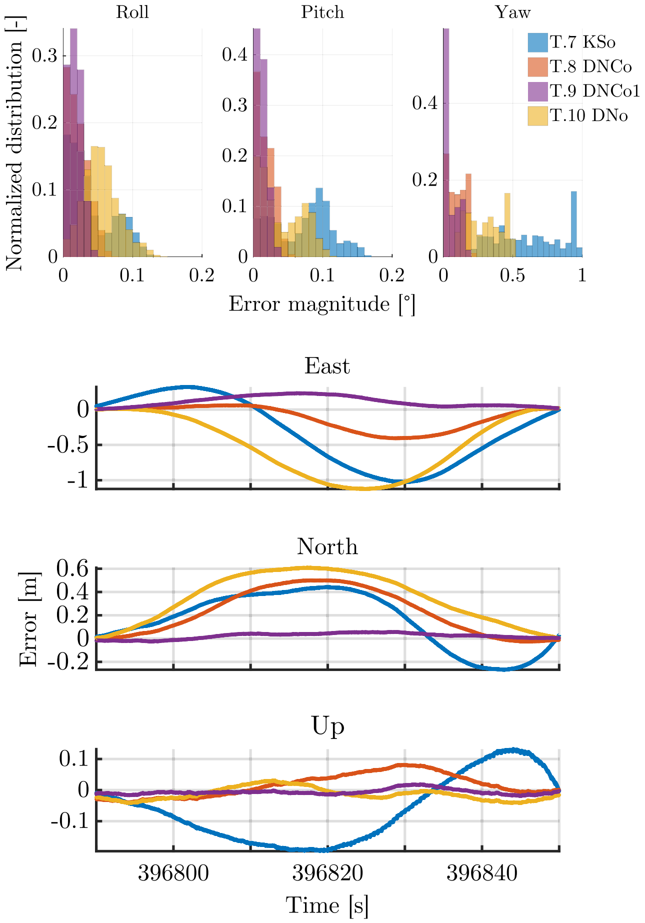

Fig. 16. Errors in position and attitude during outage on line 1 (60 s) on 4 tested trajectories.

Fig. 15, the error distribution is almost identical to the cases discussed previously (see Fig. 7). After filtering, most of the correspondences have an error between 10 cm and 20 cm, while the mean error in the correspondences is 15.8 cm (compared to 15.6 cm in T.1 KS and T.3 DN) with 1σ dispersion of 7.8 cm. Due to the similarity in correspondence quality, we use the same set of correspondences (originated from T.7 KSo dataset, containing a double outage) in both T.8 DNCo and T.9 DNCo1.

Trajectory: Similarly to Case 1, we estimate trajectory errors in terms of position and attitude. For a fair comparison, we estimate the errors only for the portion of outage within the first flight line, that is common to all four trajectories T.7 to T.10. The first 5 columns of Table 5 as well as Fig. 16 (left) illustrate the effect of the outage on the position estimation.

We first observe that the dynamic network without correspondences (T.10 DNo) estimates airborne positions during a GNSS outage are slightly less precisely than the Kalman smoother (T.7 KSo) in the horizontal direction and more precise in the vertical. This is likely due to modelling differences of the inertial errors (i.e., inclusion of scale-factors for accelerometers and gyroscopes in KS). On the other hand, the attitude errors of the smoother (T.7 KSo) are, especially in pitch and yaw, larger than those obtained via the dynamic network without correspondences (T.10 DNo).

Table 6
Point clouds georeferencing error with respect to the 1st flight line.

Traj.	RMSE [m]			Norm [m]	
	East	North	Up	Mean	Std
T.7 KSo	1.375	0.503	0.140	1.225	0.814
T.8 DNCo	0.326	0.301	0.042	0.376	0.248
T.9 DNCo1	0.158	0.055	0.019	0.130	0.107
T.10 DNo	1.089	0.312	0.099	0.977	0.582

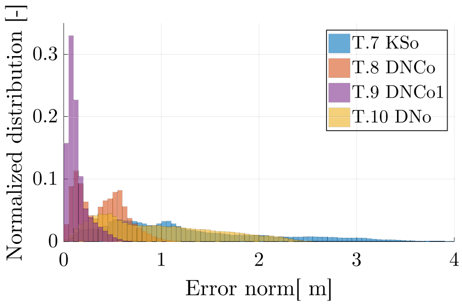

Fig. 17. Normalized distribution of point cloud georeferencing error in magnitude for the 4 trajectories.

Second, we notice that the impact of using correspondences depends on the outage scenario. When considering the outage on two flight lines (T.8 DNCo), both points in each correspondence fall within the “GNSS denied segment”. In such a situation, a constant shift in position estimation occurring along the same direction during the outages will not be detectable via correspondences because they are relative, as shown in Eq. 3. This explains somewhat lower efficiency of the correspondences to correct for trajectory position errors when both points are from a “GNSS denied area”. Nevertheless, the mean error in position is reduced ~2 × , and the RMSE in attitude ~2 × ,~3× and ~6× in roll, pitch and yaw, respectively. As it will be shown later, this reduction of errors will substantially improve the quality of point cloud registration.

Finally, we compare the Kalman smoother (T.7 KSo) with the dynamic network (T.9 DNCo1) facing a single outage. In this situation, only one of the flight lines is within the “GNSS denied” zone. Therefore a point originating from the trajectory that is not affected by the outage in each correspondence acts as an anchor for a point affected by the outage. This explains the superior correction capacity of the dynamic network in this scenario: while the position errors are maintained at ~0.1 m level (RMSE), (i.e., ~5× smaller than T.7 KSo) those in attitude are significantly lower (by a factor 4 × –10 ×) than T.7 KSo and even lower (2 × –3 ×) than those observed in the Kalman smoother without GNSS outage (T.1 KS).

Point cloud: We estimate the point-wise georeferencing error in the

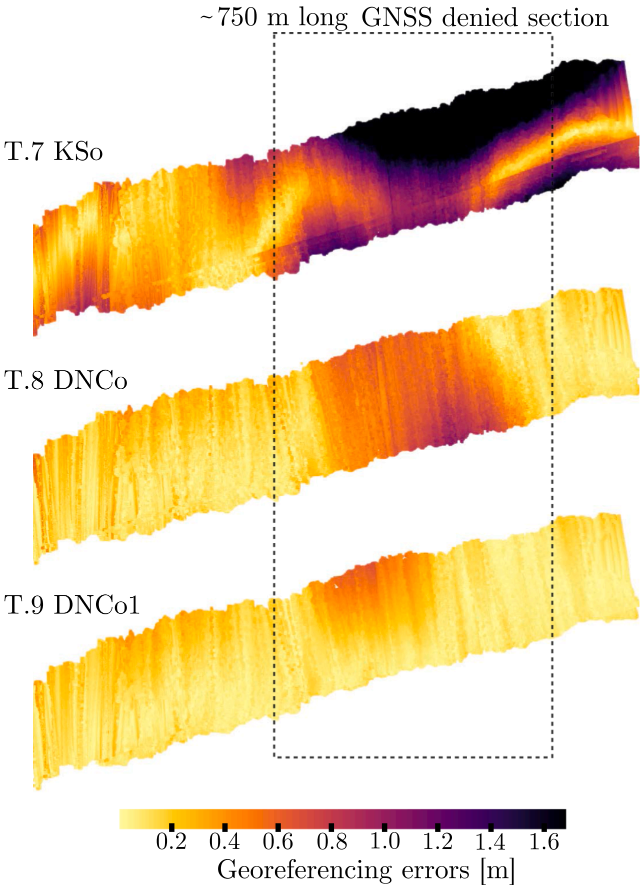

Fig. 18. Normalized distribution of point cloud georeferencing error (magnitude) for the 4 trajectories.

point clouds (with respect to the reference cloud) for trajectories T.7 to T.10. The axis-wise RMSE and mean georeferencing errors are displayed in Table 6.

First we compare T.7 KSo and T.8 DNCo. There the correspondences reduced the error norm (mean and std) 3 × –4 × . Note that the error distribution (marked red in Fig. 17) contains 2 peaks for T.8 DNCo. This is related to the residual error in trajectory estimation along East axis, that is correct only during the first half of the outage. The inclusion of correspondences considerably reduces also the tail of error distribution (i.e., the size of maximum errors) from 4 m (T.7 KSo) and 2.5 m (T.10 DNo) down to 1 m (T.8 DNCo).

The improvements are more substantial when only one flight line is impacted by the GNSS signal obstruction (i.e., T.9 DNCo1) as one side of the correspondences serves as a "position anchor"). In this situation, the georeferencing error is reduced by a factor of 9–10 compared to T.7 KSo – from 1.25 (∼ 9× GSD m to 0.13 m (∼ 1× GSD). Also importantly, the maximum errors are reduced substantially and do not exceed 0.5 m.

The point cloud georeferencing error along the entire flight line is illustrated in Fig. 18. Outside of the GNSS denied section, both T.8 DNCo and T.9 DCo1 show significantly lower errors than T.7 KSo. Inside the GNSS denied section, we can observe that the error varies rapidly with very large deviations in T.7 KSo as compared to T.8 DNCo, where the error pattern is smoother but still remains somewhat widespread. This illustrates the difference between the impact of attitude errors (of which the projection on the point cloud coordinates is rapidly varying – T.7 KSo) and the position errors (that are slowly drifting, T.8 DNCo) within the laser georeferencing. The remaining errors in the point-cloud for T.9 DCo1 are not far from those obtained by the same method without GNSS outage.

Summary: In this last case, we evaluated the performance of the proposed approach in the event of GNSS outages. We showed that the use correspondences within the proposed approach produces a significant reduction in the trajectory attitude error as compared to the standard approach. The error in trajectory position is also significant reduced using the described method when only one flight line is affected by an outage. This is due to the fact that one point in each correspondence serves as a position anchor to another point affected by the GNSS outage thereby enabling the correction of the part of trajectory within the GNSS denied segment. The improvement is somewhat lower when both flight lines experience an outage since and their respective "deviations" go in a similar direction and magnitude (in the mapping frame). Such a situation is not observable at the correspondence level that act as relative constraints. Nevertheless, in our experiment, their inclusion still leads to a reduction of ∼300% of the point cloud georeferencing error within the GNSS denied area.

5. Conclusions

In this work we first used "close-range" airborne laser scanning (AGL < 300 m) over terrain with different typology (low & high vegetation, buildings, infrastructure) to evaluate some existing methods in feature-based point–to–point matching. We combined the promising detector/descriptor with additional filtering to obtain robust lidar 3D point–to–point correspondences between overlapping areas. We integrated those as additional observations into a dynamic network which at its base considers GNSS positions and raw inertial data (specific forces and angular rates) in a single (iterative non-linear least-squares) adjustment. For a system with small MEMS-IMU (as used in UAVs), this "tight" sensor fusion resulted in an improved trajectory determination that in turn produced considerably better results in the point cloud registration process.

We investigated different scenarios related to the correspondence quality and quantity, lack of knowledge in system parameters, as well as limitations in GNSS signal reception, from which we can draw the following conclusions:

- The chosen detection, matching and filtering of over overlapping patches (50 × 50 m) resulted in a high number of correspondences irrespective of terrain type.
- The performance of correspondence quality was not affected by poor approximation of initial trajectory either due to a non-calibrated system or long GNSS outage. This indirectly indicates that the translation invariance properties of the employed descriptor were sufficient with respect to the size of patches used in the correspondence matching. Indeed, in all studied cases, the goodness of 3D point–to–point matching was close to the mean GSD value (≈ 0.15 m, the inverse of which was therefore used as a weighting for all correspondences within the dynamic network.
- The inclusion of correspondences into the dynamic network effectively mitigated the "wavy" patterns in the point cloud registration caused by the attitude errors in trajectory. This was the largest source of georeferencing error for a pre-calibrated systems and nominal GNSS reception.
- The observered improvement in the point cloud registration was practically the same when employing only 5% of the total number of correspondences; and still significant enhancement was detected using only 1% to 0.1% of correspondences.
- Thanks to the correspondences, the point cloud registration via dynamic networks was also able to cope well with an unknown lidar boresight. As the accuracy of the recovered boresight parameters is related to their correlation with systematic errors in attitude, their determination was restrained by the geometry of the studied case of two parallel flight lines. This had, however, very small impact on the residual errors in the point cloud due to the previously mentioned correlation.

Table 7
Descriptors parameters, run-time and performances.

Descriptor	Dim.	Parameters	Computation time [s]	Match time [s]	Inlier ratio @30 cm [%]
SHOT	352	$r_s = 15 * \text{GSD}$	49	73	13.3
FPFH	32	$r_s = 15 * \text{GSD}$	16	1,4	5.7
USC	1980	$r_s = 15 * \text{GSD}$	82	612	5.2
FCGF	32	vox. size = 0.2 kernel size = 7	0,21	1,3	27.3
SpinNet	32	$r_s = 2$, rad$_n = 9$, ele$_n = 30$, azi$_n = 60$ vox$_r = 0.3$, vox$_s = 30$	1410	2	30.7
LCD	256	$r_s = 2$ m patch pts N = 1024	128	31	53.7

- Utilizing correspondences within the dynamic network can also effectively mitigate the imprecise point cloud registration during the absence of GNSS signals, especially when affecting only one of a pair of flight lines.

In our opinion, methods that attempt to perform the integration of all the available information in a single step, such as the one put forward in this contribution, are to be preferred over multiple stages, cascade adjustment approaches because of their simplicity, rigour and effectiveness in correcting errors at the sensor measurement level, rather than in the intermediate products. The proposed approach can be applied to other kinematic scanning scenarios, such as terrestrial laser scanning, and generalise well to more complex multi-sensor adjustment problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The helicopter flight and the reference sensors were provided by Helimap System. The personal involvement and contribution of its director Dr. Julien Valet are highly appreciated. This contribution was partially supported by the Swiss National Foundation (200021182072) and partially by the European H2020 project GAMMS, EC No. 101004255. The authors would like to thank Jesse LaHaye for his help in proofreading the manuscript.

Appendix A. Descriptors evaluation

Describing feature characteristics observed by kinematic laser scanning is a challenging task due to the sparse nature of point clouds (i.e., compared to images) and to their high variability depending on the scanned surface, the technology and the vehicle used. Substantial effort has been devoted to the design efficient point cloud descriptors, and since the advent of deep-learning considerable progress was registered in this field (Zhang et al., 2020; Huang et al., 2021). These descriptors were usually designed and tested for TLS or MLS point clouds and their generalization potential to ALS point clouds remain uncertain. Therefore we evaluated multiple 3D descriptors, namely FPFH (Rusu et al., 2009), SHOT (Tombari et al., 2010b), USC (Tombari et al., 2010a), SpinNet (Ao et al., 2021), FCGF (Choy et al., 2019) and LCD (Pham et al., 2019). While not being fully detailed here, we provide some evaluation of their suitability with respect to our ALS experimental data in Table 7. There we summarize for each tested descriptor its number of dimension, parameters, run times and performances. Computation time is measured when estimating descriptors on a pair of ALS fragments (of around ~120’000 points each) on a modern computer with the following configuration under Ubuntu 20.04: Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz, 32 Gb RAM, NVDIA RTX 2070.

The inlier ratio at 30 cm represents the fraction of retrieved correspondences whose error in the reference point clouds is below 30 cm. This is the main metric used to select the final descriptor.

For both FCGF and SpinNet, pre-trained version of their network on KITTI outdoor dataset are used. For LCD, the pre-trained version on indoor dataset is used since it is the only one available.

Appendix B. RANSAC algorithm

In this section, we detail the version of the RANSAC algorithm used in this research and its parameters.

B.1. Parameters

- $\mathbf{C}$ the complete set of $\mathbf{K}$ correspondences.
- $\mathbf{p}_\mathbf{a}^\mathbf{k}$ and $\mathbf{p}_\mathbf{b}^\mathbf{k}$ the first and second point of the k^{th} correspondence.
- $\mathbf{R}$ and $\mathbf{T}$ the rotation matrix and translation vector defining the 6 DoF transformation.
- $\boldsymbol{\tau}$ the tolerance threshold to classify correspondences as inliers or outliers given an estimated 6 DoF transformation parameter.
- s the number of correspondences to select to form a subset.
- N$_{it}$ the number of iterations to perform, defined following the probabilistic approach from the original publication (Fischler and Bolles, 1981).

B.2. Algorithm

Algorithm 1. RANSAC

```
Data: Set of correspondences $\mathbf{C}$
Parameter: $\tau, N_{it}, s$
while $t < N_{it}$ do
    Randomly select $\mathbf{C_{sub}} \subset \mathbf{C}$ of s elements
    Estimate $\hat{\mathbf{R}}, \hat{\mathbf{T}}$:
        $\hat{\mathbf{R}}, \hat{\mathbf{T}} = \underset{R,T}{argmin} \sum_{\mathbf{i}}^{\mathbf{s}} ||\mathbf{R}p_a^i + \mathbf{T} - p_b^i||$
    for  $\mathbf{c} \in \mathbf{C} \setminus \mathbf{C_{sub}}$ do
        if $||\hat{\mathbf{R}}p_a^c + \hat{\mathbf{T}} - p_b^c|| < \tau$ then
            c is an inlier
        else
            c is an outlier
    end
    Inlier ratio$_t = \frac{\#Inliers}{\#Correspondences}$
    t++
end
Results: $\hat{\mathbf{R}}, \hat{\mathbf{T}}$ and inlier classification for the subset with most inliers
```

References

Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y., 2021. SpinNet: Learning a general surface descriptor for 3d point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11753–11762.

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-D shapes. In: Sensor fusion IV: control paradigms and data structures, Spie. pp. 586–606. https://doi.org/10.1117/12.57955.

Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D., 2021. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans. Rob. 37, 1874–1890. https://doi.org/10.1109/TRO.2021.3075644.

Ceriani, S., Sánchez, C., Taddei, P., Wolfart, E., Sequeira, V., 2015. Pose interpolation SLAM for large maps using moving 3D sensors. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 750–757. https://doi.org/10.1109/IROS.2015.7353456.

Chen, Y., Medioni, G., 1992. Object modelling by registration of multiple range images. Image Vision Comput. 10, 145–155. https://doi.org/10.1016/0262-8856(92)90066-C.

Choy, C., Park, J., Koltun, V., 2019. Fully convolutional geometric features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8957–8965. https://doi.org/10.1109/ICCV.2019.00905.

Cioffi, G., Scaramuzza, D., 2020. Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 5089–5095. https://doi.org/10.1109/IROS45743.2020.9341697.

Clausen, P., Skaloud, J., 2020. On the calibration aspects of MEMS-IMUs used in micro UAVs for sensor orientation. In: IEEE-ION Position Location and Navigation Symposium (PLANS), pp. 1457–1466. https://doi.org/10.1109/PLANS46316.2020.9110160.

Colomina, I., Blázquez, M., 2004. A unified approach to static and dynamic modeling in photogrammetry and remote sensing. International Archives of Photogrammetry, Remote Sensing and Spatial. Inf. Sci. 35–B1, 178–183.

Cucci, D.A., Clausen, P., Skaloud, J., Matteucci, M., 2017a. A general approach to time-varying parameters in pose-graph optimization. In: 2017 European Navigation Conference (ENC), IEEE, pp. 265–271. https://doi.org/10.1109/EURONAV.2017.7954217.

Cucci, D.A., Matteucci, M., 2014a. On the development of a generic multi-sensor fusion framework for robust odometry estimation. J. Softw. Eng. Robot. 5, 48–62 doi: 10446/87664.

Cucci, D.A., Matteucci, M., 2014b. Position tracking and sensors self-calibration in autonomous mobile robots by Gauss-Newton optimization, in. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 1269–1275. https://doi.org/10.1109/ICRA.2014.6907016.

Cucci, D.A., Rehak, M., Skaloud, J., 2017b. Bundle adjustment with raw inertial observations in uav applications. ISPRS J. Photogramm. Eng. Remote Sens. 130, 1–12. https://doi.org/10.1016/j.isprsjprs.2017.05.008.

Cucci, D.A., Skaloud, J., 2019. On raw inertial measurements in dynamic networks. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial. Inf. Sci. IV-2/W5, 549–557. https://doi.org/10.5194/isprs-annals-IV-2-W5-549-2019.

Davis, T.A., 2006. Direct methods for sparse linear systems. SIAM. https://doi.org/10.1137/1.9780898718881.

Filin, S., 2003. Recovery of systematic biases in laser altimetry data using natural surfaces. Photogramm. Eng. Remote Sens. 69, 1235–1242. https://doi.org/10.14358/PERS.69.11.1235.

Filin, S., Vosselman, G., 2004. Adjustment of airborne laser altimetry strips. In: ISPRS Congress, International Archives of Photogrammetry and Remote Sensing, Istanbul, Turkey, pp. 285–289.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395. https://doi.org/10.1145/358669.358692.

Friess, P., 2006. Towards a rigorous methodology for airborne laser mapping. In: International Calibration and Orientation Workshop (EuroCOW), ISPRS, Castelldefels, Spain, p. 7.

Glennie, C.L., 2007. Rigorous 3D error analysis of kinematic scanning lidar systems. J. Appl. Geodesy 1, 147–157. https://doi.org/10.1515/jag.2007.017.

Glennie, C.L., Lichti, D., 2010. Static calibration and analysis of the velodyne HDL-64E S2 for high accuracy mobile scanning. Remote Sens. 2, 1610–1624. https://doi.org/10.3390/rs2061610.

Glira, P., Pfeifer, N., Briese, C., Ressl, C., 2015a. A correspondence framework for ALS strip adjustments based on variants of the ICP algorithm. Photogrammetrie-Fernerkundung-Geoinformation, 275–289. https://doi.org/10.1127/pfg/2015/0270 .

Glira, P., Pfeifer, N., Brieses, C., Ressl, C., 2015b. Rigorous strip adjustment of airbone laserscanning data based on the ICP algorithm. ISPRS Ann. Photogramm. Remote Sens. Spatial. Inf. Sci. 2–3 (W5), 73–80. https://doi.org/10.5194/isprsannals-II-3-W5-73-2015.

Glira, P., Pfeifer, N., Mandlburger, G., 2016. Rigorous strip adjustment of uav-based laserscanning data including time-dependent correction of trajectory errors. Photogramm. Eng. Remove Sens. 82, 945–954. https://doi.org/10.14358/PERS.82.12.945.

Hebel, M., Stilla, U., 2012. Simultaneous calibration of ALS systems and alignment of multiview LiDAR scans of urban areas. IEEE Trans. Geosci. Remote Sens. 50, 2364–2379. https://doi.org/10.1109/TGRS.2011.2171974.

Huang, X., Mei, G., Zhang, J., Abbas, R., 2021. A comprehensive survey on point cloud registration. CoRR abs/2103.02690. arXiv:2103.02690.

Kager, H., 2004. Discrepancies between overlapping laser scanning strips - simultaneous fitting of aerial laser scanner strips. In: Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 555–560.

Kerstling, A., Habib, A., Bang, K.I., Skaloud, J., 2012. Automated approach for rigorous light detection and ranging system calibration without preprocessing and strict terrain coverage requirements. Opt. Eng. 51, 076201. https://doi.org/10.1117/1.OE.51.7.076201.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W., 2011. g^2o: A general framework for graph optimization. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3607–3613. https://doi.org/10.1109/ICRA.2011.5979949.

Loeliger, H.A., Dauwels, J., Hu, J., Korl, S., Ping, L., Kschischang, F.R., 2007. The factor graph approach to model-based signal processing. Proc. IEEE 95, 1295–1322. https://doi.org/10.1109/JPROC.2007.896497.

Lupton, T., Sukkarieh, S., 2011. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Trans. Rob. 28, 61–76. https://doi.org/10.1109/TRO.2011.2170332.

Pham, Q.H., Uy, M.A., Hua, B.S., Nguyen, D.T., Roig, G., Yeung, S.K., 2019. LCD: Learned cross-domain descriptors for 2D–3D matching. arXiv:1911.09326.

Pottmann, H., Huang, Q., Yang, Y., 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. Int. J. Comput. Vision 67, 277–296. https://doi.org/10.1007/s11263-006-5167-2.

Rauch, H.E., Tung, F., Striebel, C.T., 1965. Maximum likelihood estimates of linear dynamic systems. AIAA J. 3, 1445–1450. https://doi.org/10.2514/3.3166.

Riegl, 2021. miniVUX-UAV3. URL: http://www.riegl.com/products/unmanned-scanning/.

Rouzaud, D., Skaloud, J., 2011. Rigorous integration of inertial navigation with optical sensors by dynamic networks. NAVIGATION. J. Inst. Navigat. 58, 141–152. https://doi.org/10.1002/j.2161-4296.2011.tb01797.x.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217. https://doi.org/10.1109/ROBOT.2009.5152473.

Rusu, R.B., Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. https://doi.org/10.1109/ICRA.2011.5980567.

Salti, S., Petrelli, A., Tombari, F., Di Stefano, L., 2012. On the affinity between 3D detectors and descriptors. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission, pp. 424–431. https://doi.org/10.1109/3DIMPVT.2012.10.

Schaer, P., Vallet, J., 2016. Trajectory adjustment of mobile laser scan data in GPS denied environments. ISPRS - Int. Arch. Photogramm., Remote Sens. Spatial Informat. Sci. XL-3/W4, 61–64. https://doi.org/10.5194/isprs-archives-XL-3-W4-61-2016.

Skaloud, J., 2017. LIEO: LiDAR orientation and point cloud generation, software description n6.0704, licence via EPFL-TTO. URL: http://tto.epfl.ch.

Skaloud, J., Colomina, I., Parés, M.E., Blázquez, M., Silva, J., Chersich, M., 2018. A method of airborne gravimetry by combining strapdown inertial and new satellite observations via dynamic networks. In: Freymueller, J.T., Sánchez, L. (Eds.), International Symposium on Earth and Environmental Sciences for Future Generations. Springer International Publishing, Cham, pp. 111–122. https://doi.org/10.1007/1345_2016_229.

Skaloud, J., Lichti, D., 2006. Rigorous approach to bore-sight self calibration in airborne laser scanning. ISPRS J. Photogramm. Remote Sens. 61, 47–59. https://doi.org/10.1016/j.isprsjprs.2009.12.003.

Skaloud, J., Schaer, P., 2007. Towards automated lidar boresight self-calibration. Int. Arch. Photogramm., Remote Sens. Spatial Informat. Sci. 36.

Skaloud, J., Schaer, P., 2012. Automated assessment of digital terrain models derived from airborne laser scanning. Photogrammetrie, Fernerkundung, Geoinformation (PFG) 2, 105–114. https://doi.org/10.1127/1432-8364/2012/0105.

Tombari, F., Salti, S., Di Stefano, L., 2010a. Unique shape context for 3D data description. In: Proceedings of the ACM Workshop on 3D Object Retrieval, Association for Computing Machinery, New York, NY, USA. pp. 57–62. https://doi.org/10.1145/1877808.1877821.

Tombari, F., Salti, S., Di Stefano, L., 2010b. Unique signatures of histograms for local surface description. In: Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III. Springer-Verlag, Berlin, Heidelberg, pp. 356–369. https://doi.org/10.1007/978-3-642-15558-1_26.

Vallet, J., Gressin, A., Clausen, P., Skaloud, J., 2020. Airborne and mobile lidar, which sensors for which application? ISPRS - Int. Arch. Photogramm., Remote Sens. Spatial Informat. Sci. XLIII-B1-2020, 397–405. https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-397-2020.

Vallet, J., Skaloud, J., 2004. Development and experiences with a fully-digital handheld mapping system operated from a helicopter. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Istanbul. pp. Part B, Commission 5.

Zhang, Z., Dai, Y., Sun, J., 2020. Deep learning based point cloud registration: an overview. In: Virtual Reality Intell. Hardware, 2, pp. 222–246. https://doi.org/10.1016/j.vrih.2020.05.002, 3D Visual Processing and Reconstruction Special Issue.

Zhong, Y., 2009. 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. https://doi.org/10.1109/ICCVW.2009.5457637.