LIDAR BASED POSE TRACKING OF AN UNCOOPERATIVE SPACECRAFT USING THE SMOOTHED NORMAL DISTRIBUTION TRANSFORM

Léo Renaut(1, 2), Heike Frei(1), Andreas Nüchter(2)

(1)*German Aerospace Center (DLR), 82234 Wessling, Germany; leo.renaut@dlr.de, heike.frei@dlr.de*
(2)*Julius-Maximilians-Universität Würzburg, Informatics XVII-Robotics, 97074 Würzburg, Germany; andreas.nuechter@uni-wuerzburg.de*

ABSTRACT

Lidar sensors provide precise 3D point cloud measurements, and can be used for pose estimation of an uncooperative satellite during space rendezvous. For updating the estimated pose of a the target, iterative closest point (ICP) or one of its variants is usually applied as a tracking method. However, for dense point clouds and space hardware with reduced computing power, the execution time of ICP can become a limiting factor. Normal distribution transform (NDT) is an alternative algorithm for fine point cloud registration, which can be faster than ICP. Yet, NDT can be less robust than ICP due to the discontinuities in its cost function. This work proposes a smoothing method of the NDT map in order to mitigate this robustness problem. In addition, a strategy for correcting motion blur observed with lidar sensors when recording a tumbling target is developed. Experiments at the European Proximity Operations Simulator demonstrate the efficiency, precision and robustness of the smoothed NDT algorithm when compared to ICP, as well as the importance of motion blur correction for precise pose estimation.

1 INTRODUCTION

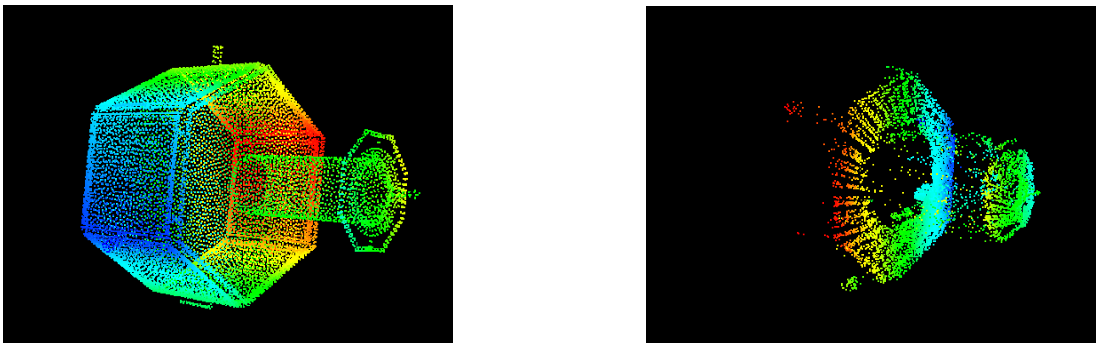
In order to reduce the number of space debris in Earth orbit and to avoid potential collisions with other satellites, there is a need for active debris removal (ADR) missions. An additional way to bring more sustainability in orbit are on-orbit-servicing (OOS) missions, which aim at repairing a satellite in orbit, or extending its lifetime by refuelling. For both ADR and OOS missions, an active chaser satellite needs to rendezvous with an uncooperative target spacecraft or debris, before performing proximity operations. Because the target is uncontrolled, it might be tumbling freely. For potential robotic operations, it is necessary to estimate its full pose (position and attitude) in close-range (typically at relative distances below $30m$).

The illumination conditions in orbit are harsh due to the frequent day/night eclipses, reflections, and the high luminosity contrasts. These effects can impact the reliability of the pose estimation when using an optical camera as a navigation sensor [1]. On the opposite, active sensors such as time-of-flight (ToF) cameras or scanning lidars are less perturbed by these effects, and provide an additional depth information. ToF cameras only have a limited working range with a maximum relative distance of around $10m$ [2], [3], while scanning lidars can range over several hundred meters [4]. However, lidar sensors are also subject to motion blur effects due to their low scanning rate, especially when the target spacecraft is tumbling rapidly. In this work, we investigate the use of a scanning lidar for relative pose tracking.

It is supposed that a 3D model of the target satellite is known, as shown on Fig. 1. This model can possibly be the result of an inspection flight around the unknown object. Given the 3D model, pose estimation is usually performed in two steps, pose initialization and pose tracking. Pose initialization consists in finding the target's pose without an initial guess of the solution, it is a global optimization problem. On the contrary, pose tracking consists in refining an initial guess (previous pose or filter prediction) to find the corrected pose, it is a local optimization problem. After pose initialization is performed, pose tracking is usually run continuously to update the pose estimation whenever a new point cloud is received. Several techniques exist for pose initialization [5]–[7], yet this work will address only the topic of pose tracking.

The process of aligning two point clouds given an initial estimate is also referred to as point cloud registration. The most widespread algorithm for point cloud registration is the iterative closest point (ICP) [8]. If ICP is an efficient method, for large point clouds, it can become challenging to execute it in real-time, especially on space hardware with limited computing power [3], [9]. Normal distribution transform (NDT) [10] is an alternative registration method which was reported to be as precise as ICP while being more efficient [11], [12]. However, NDT can be less robust to the uncertainty in the initial estimate than ICP [11].

We introduce a smoothing method of the NDT map in order to increase the robustness of the NDT algorithm while maintaining its computational efficiency, and apply the smoothed NDT algorithm to the problem of lidar tracking of an uncooperative spacecraft. Additionally, we propose a strategy for deblurring the point clouds relying on the use of a Kalman filter to mitigate the effects of motion blur observed with tumbling targets. The proposed method is compared to ICP and evaluated during hardware-in-the-loop tests at the European Proximity Operations Simulator (EPOS).



(a)

(b)

Figure 1: Point clouds of the target satellite. **(a)** Point cloud obtained from a CAD model of the satellite. **(b)** Point cloud captured by the lidar at the test facility.

2 RELATED WORK

ICP was first introduced by Besl and McKay [8] and has multiple variants, such as point-to-plane ICP [13] or trimmed ICP [14]. For testing the TriDAR sensor with the Space Shuttle, a point-to-plane variant of ICP was applied [4]. Likewise, Opromolla et al. apply ICP with normal shooting after the first iteration, and test their method on simulated LiDAR data [15]. Liu et al. use standard ICP to track an uncooperative target with a ToF camera on simulated data and field experiments [16]. However, no ground truth is provided for error evaluation the error of their method during the field experiment, and problems due to noisy point clouds and reflections are reported. For mitigating the effects of sensor noise, ICP can be combined with RANSAC to reject wrong associations [17]. Reflective materials such as golden multi-layer insulation (MLI) sheets can lead to reflections and outliers in the sensor data [18]. Therefore, the authors apply a trimmed ICP variant to filter out point pairs which have a

distance greater than the median distance of the point pairs. Because of the relatively low frame-rate of scanning lidars, the relative motion between two scans might be important. Thus instead of using the previous registration estimate as an initial guess, Opromolla and Nocerino use the prediction result of an unscented Kalman filter to initialize ICP registration with better precision [19].
When the number of points in the cloud becomes important, especially with ToF sensors which provide high density point clouds, ICP registration in real-time is challenging. Thus to reduce the point cloud size, Zhao et al. combine ICP with an adaptive down-sampling strategy and depth image processing to ensure real-time execution [9]. Alternatively, Sun et al. propose to extract salient points from a dense point cloud, and to apply ICP on the reduced point cloud, in order to achieve computation times compatible with real-time requirements [3]. The authors additionally assume that no 3D model of target is known in the beforehand, and build a point cloud map of the target on-line. Their method is also compared with a standard NDT implementation from the Point Cloud Library.
NDT was initially developed for registration of 2D point clouds by Biber and Straßer [10], and later extended to 3D by Magnusson et al. [20]. If NDT is a precise and efficient registration method [11], its cost function is discontinuous because of the voxelized representation of the probability function. Therefore, in their original implementation, Biber and Straßer propose to use four overlapping voxel grids to smoothen the cost function [10]. In 3D, the same strategy can be used, and corresponds to evaluating eight overlapping grids simultaneously [21]. Likewise, Magnusson et al. use tri-linear interpolation to evaluate the eight neighbouring cells [11]. In both cases, the authors report an increase in the computation time by four to eight times due to this strategy. We propose a way to smoothen the NDT map in a pre-processing step, so that the computation time of the registration remains unchanged.

3 METHODS

3.1 Smoothed NDT

NDT is a fine registration algorithm aiming at aligning two point clouds, a target point cloud $\boldsymbol{X}$ and a source point cloud $\boldsymbol{Z} = (\boldsymbol{z_1}, ..., \boldsymbol{z_n})$. Typically, $\boldsymbol{X}$ is a point cloud generated from a CAD model of the target spacecraft, while Z is the point cloud captured by the lidar during operations. The principle of NDT is to subdivide the target point cloud into a voxel grid. For each voxel k, the mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{C}_k$ of the points of the target cloud X which belong to that voxel is computed.
The objective is to find the 3D transformation $\boldsymbol{T} \in SE(3)$ which aligns the source point cloud with the target point cloud. Each transformed source point $\boldsymbol{T}(\boldsymbol{z}_i)$ falls into a voxel k_i of the target point cloud, which is represented by the probability distribution with parameters $(\boldsymbol{\mu}_{k_i}, \boldsymbol{C}_{k_i})$. Contrarily to classical NDT [10], we propose the use of a relaxed cost function in order to formulate the optimization problem as a weighted least squares problem and speed up the registration, similarly to [22]. The optimal pose T which aligns both point clouds is the one which minimizes:

$$s(\boldsymbol{T}) = \sum_{i=1}^{n} \left(\boldsymbol{\mu}_{k_i} - \boldsymbol{T}(\boldsymbol{z}_i)\right)^T \boldsymbol{C}_{k_i}^{-1} \left(\boldsymbol{\mu}_{k_i} - \boldsymbol{T}(\boldsymbol{z}_i)\right) \ . \tag{1}$$

The optimal pose is found iteratively following a Gauss-Newton procedure. We refer to our previous work for a simple analytical formulation of the gradients for solving the optimization problem in the 6D manifold of the homogeneous 3D transformation group $SE(3)$ [23].
The second proposed modification of NDT addresses the discontinuity issue of the NDT map presented in Section 2. In [24], we introduced a smoothing method of the NDT map that we summarize here. The target point cloud is split into m cells using a kd-tree partition. For each cell k, the NDT map parameters $(\boldsymbol{\mu}_k, \boldsymbol{C}_k)$ are computed, as well as the number of points n_k that belong to this voxel.

If $\boldsymbol{c}_k$ is the center of the voxel, for a radius r, the set of indexes of the neighbouring distributions within this radius is

$$\mathcal{N}_k = \{i \in \{0, ..., m\} \mid ||\boldsymbol{\mu}_i - \boldsymbol{c}_k|| < r\}\,. \tag{2}$$

The idea of smoothed NDT is to replace the probability distribution $\mathcal{D}_k$ characterized by $(\boldsymbol{\mu}_k, \boldsymbol{C}_k)$ by the weighted sum of the neighbouring distributions $\mathcal{D}_i$ with $i \in \mathcal{N}_k$. Given a fixed central cell k, the weights w_i are given by a continuous Gaussian blur with a standard deviation σ, and multiplied by the number of points n_i of each distribution

$$w_i = n_i \exp(-\frac{||\boldsymbol{\mu}_i - \boldsymbol{c}_k||^2}{2\sigma^2})\,. \tag{3}$$

σ can be chosen to equal the cell size. The resulting smoothed distribution represents not only the probability distribution of the points within the cell, but also of the points in the neighbourhood of the cell:

$$\widetilde{\mathcal{D}}_k \sim \frac{\sum_{i\in\mathcal{N}_k} w_i \mathcal{D}_i}{\sum_{i\in\mathcal{N}_k} w_i}\,. \tag{4}$$

Each distribution $\mathcal{D}_k$ is thus replaced by the smoothed distribution $\widetilde{\mathcal{D}}_k$ with parameters $(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{C}}_k)$. During the registration process, these smoothed parameters are used in the cost function of Eq. 1 for computing the optimal pose. In consequence, the discontinuity of the cost function when a point is shifted from a voxel to another during registration is reduced.

3.2 Filter and motion compensation

NDT being a local optimization procedure, a precise initial estimate of the solution is important for a good convergence of the algorithm. In particular, when the lidar frame-rate is low with respect to the angular tumbling velocity of the target, the previous pose estimate might not be a good initial estimate of the current pose, because it is outdated. Instead, a prediction of the current pose computed by a Kalman filter can be used, similarly to the work of Opromolla and Nocerino [19]. If the authors use an unscented Kalman filter, we opt for a more simple strategy. For the position propagation, we use a linear Kalman filter under the assumption of constant relative velocity for modelling the dynamics. The attitude estimation is performed using an invariant extended Kalman filter (IEKF) [25], an extension of the extended Kalman filter for state estimation on a matrix Lie group. Likewise, the attitude dynamics is modelled by a simple assumption of constant relative angular velocity.

The Kalman filter provides an estimate of the current pose for the time step k denoted by $(\boldsymbol{p}_k, \boldsymbol{R}_k)$, the relative position and attitude of the target satellite. The prediction of the filter for the following time step is $(\boldsymbol{p}_{k+1|k}, \boldsymbol{R}_{k+1|k})$. This prediction is used to initialize the smoothed NDT algorithm. The result of the algorithm, a pose estimation $(\bar{\boldsymbol{p}}_{k+1}, \bar{\boldsymbol{R}}_{k+1})$, is then used to update the filter.

Lidar sensors are subject to the motion blur effect when tracking a rapidly tumbling target, as illustrated on Fig. 2. The lidar scans during a time interval $[t_k, t_{k+1}]$. Each point of the cloud has a time stamp comprised in this time interval. Given an estimate of the current relative velocity and angular velocity of the target $(\boldsymbol{v}_k, \boldsymbol{\omega}_k)$ provided by the filter, the position of each point can be corrected to its propagated position at the end of the scanning time t_{k+1} [23]. The result of such a process is illustrated on Fig. 2.

The motion blur correction can be performed as a pre-processing step, in order to correct the point cloud before passing it to the smoothed NDT algorithm. The resulting logic of the pose tracker, when smoothed NDT is combined with a Kalman filter for providing an initial estimate and performing motion compensation of the point cloud, is presented in Fig. 3.

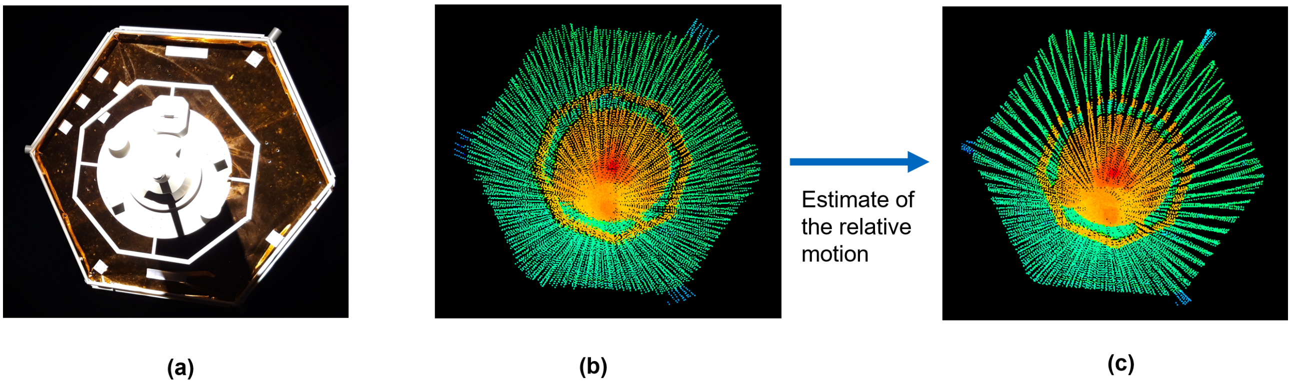


Figure 2: Illustration of the motion blur correction principle. **(a)** Picture of the target satellite mock-up used for the experiment. **(b)** Lidar point cloud captured with 1Hz of the target tumbling with 10 deg/s. The point cloud is blurred. **(c)** Motion compensated point cloud. The edges of the point cloud are sharper. Warm colors indicate points closer to the sensor, cold colors points further away.

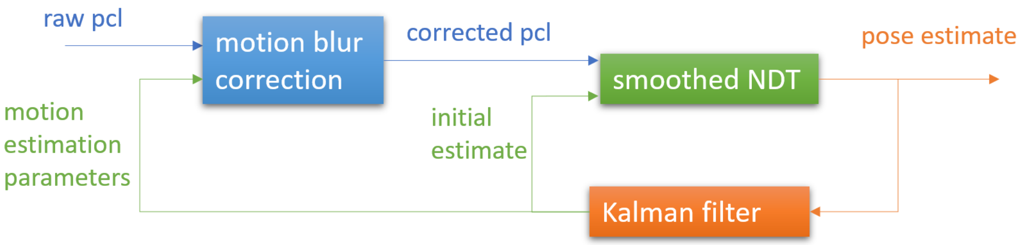


Figure 3: Logic of the pose tracking. The incoming point cloud is motion compensated before being passed to the smoothed NDT algorithm, which is initialized with the Kalman filter's prediction.

4 RESULTS AND DISCUSSION

4.1 Hardware-in-the-loop test environment

The proposed method for lidar pose tracking was tested at the European Proximity Operations Simulator (EPOS) [26]. EPOS is a hardware-in-the-loop test facility consisting of two robots which can move with 6DOF. The first robot, shown on Fig. 4, carries a mock-up of the DEOS satellite comprised of realistic materials such as the multi-layer-insulation (MLI) sheets and solar panels. The second robot can translate on a linear rail in order to reach relative distances between both satellites of up to 25m. This second robot carries a sensor plate to which a Livox Mid-40 lidar from the automotive domain was attached, and used for the experiments.

The algorithms were implemented in C++17 and run on one core of an Intel Core i7 CPU. The proposed method was compared to an efficient implementation of ICP from the 3D toolkit [27]. ICP was found to work best when setting a maximum point-to-point distance of 10cm, and the norm of the termination criteria to 10^{-6}, similarly to the work of Zhao et al. [9]. For smoothed NDT, the cell size and the maximum point-to-cell distance were set to 7.5cm. The termination criteria was set to when the increment was below $0.05°$ and 1mm. In addition, for both ICP and smoothed NDT, incoming point clouds captured by the lidar were down-sampled using a voxel grid with a resolution of 2cm.

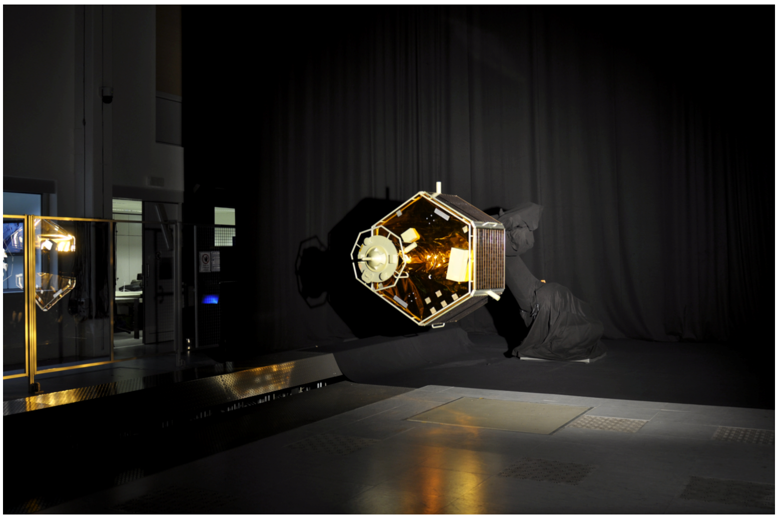


Figure 4: Picture of the European Proximity Operations Simulator. A robotic arm carries a mock-up of the target satellite. The sensors of the chaser satellite are mounted on a second robotic arm, which is not shown in this picture.

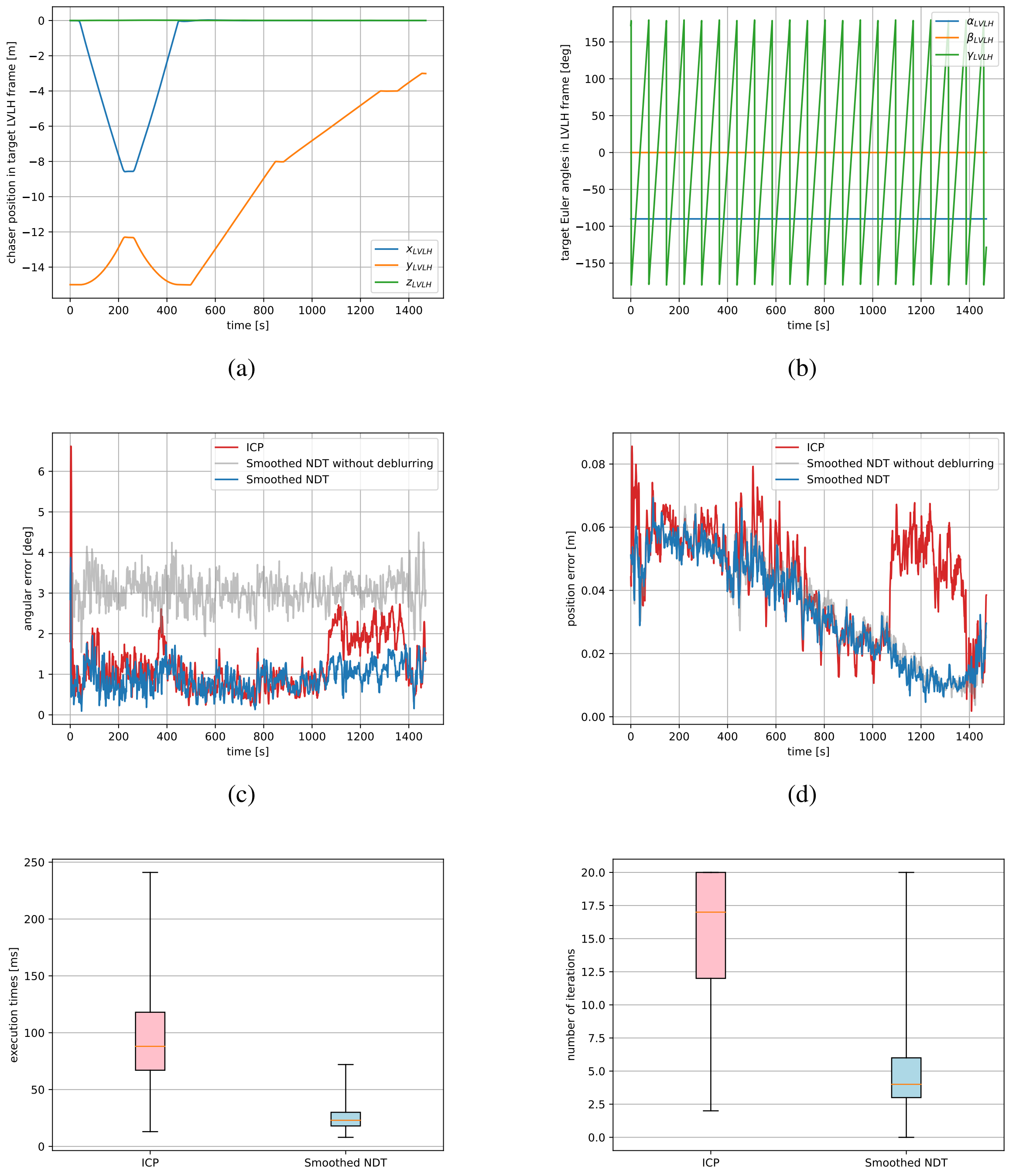
4.2 Tracking experiment of a spinning target

A hardware-in-the-loop experiment is performed to compare ICP with the proposed smoothed NDT algorithm. To ensure a fair comparison, both ICP and smoothed NDT are used in conjunction with the Kalman filter and motion compensation of the point clouds as presented in Fig. 3. The rendezvous scenario is comprised of two parts, as shown on the relative trajectory presented on Fig. 5a. Initially, the chaser satellite performs a fly-around of the target at a steady relative distance of 15m, with a relative velocity of 5cm/s. Starting from $t = 500$s, the approach towards 8m is initialized with 2cm/s. Then, at $t = 900$s, the approach is continued with 1cm/s, until a relative distance of 4m. After a second hold, the final relative distance of 3m is reached.

During the whole experiment, the target spins around its main axis with an angular velocity of $5°$/s, as illustrated on Fig. 5b. Lidar scans are sampled with a frequency of 1Hz, so that motion blur can be observed. After down-sampling with the 2cm voxel grid, the clouds have between $6,000$ and $15,000$ points. To ensure real-time execution, the maximum number of iterations of both ICP and smoothed NDT is set to 20. Additionally, for demonstrating the importance of motion compensation, an experiment was run using smoothed NDT without the deblurring module. All algorithms are initialized with a perfect pose initialization originating from the ground truth in the beginning of the experiment, in order to start the tracking.

Fig. 5c presents the total magnitude of the angular errors observed for both algorithms. After a small convergence period of the Kalman filter, both ICP and smoothed NDT algorithms converge towards a relatively small angular error for the rest of the experiment. While the ICP error remains below $2.7°$ after the first time steps, the smoothed NDT algorithm has a better angular accuracy with an error under $2°$. The angular error when removing the deblurring module (grey curve) is around $3°$ to $4°$, demonstrating the importance of a motion compensation strategy when using a scanning lidar to track a spinning target.

The position errors of the tracking are shown on Fig. 5d. For both algorithms, the position estimation error decreases with the relative distance, before slightly increasing in the end, when the target satellite is only party visible in the lidar's field of view. Smoothed NDT leads to a position error around 6cm when at 15m distance, before decreasing to around 2cm at the end of the experiment. On the other hand, ICP presents slightly larger position error, especially at close range, when the relative distance



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5: Results of the tracking experiment. **(a)** Relative position of the target over time. **(b)** Attitude of the target over time. **(c)** Compared angular errors of the tracking methods. **(d)** Compared position errors of the tracking methods. **(e)** Distribution of the execution times of both algorithms, including down-sampling and registration. **(f)** Distribution of the number of iterations of both algorithms. For the boxplots, the boxes show the first and third quartile, while the whiskers extend to the extreme values.

to the target is below $4\mathrm{m}$. The absence of the deblurring module (grey curve) does not significantly modify the position estimation error, because the relative velocities are less important compared to the angular spinning rate.

The Figs. 5e and 5f present the distributions of the total execution times and the number of iterations of both algorithms, respectively. With an average execution time of 25ms, smoothed NDT is significantly faster than ICP which has an average runtime of 93ms. This also applies for the worst case execution times (72ms for smoothed NDT versus 241ms for ICP). Likewise, the number of iterations of smoothed NDT is lower with 4.8 on average than the average number of iterations of ICP of 15.6. Both are bounded by the maximum number of iterations which was set to 20.

4.3 Robustness evaluation

The second experiment aims at comparing the robustness of both algorithms with respect to the precision of the initial estimate. Because both ICP and smoothed NDT are local optimization methods relying on an initial estimate of the solution, the objective is to evaluate how precise the initial estimate has to be in order for the algorithms to converge. A similar trajectory to the one presented in the first experiment (section 4.2) was flown in the EPOS facility. However, in order to only evaluate the robustness of both algorithms without being stronlgy affected by motion blur, the angular velocity of the target satellite was lowered to 1°/s. The lidar frame-rate was kept at 1Hz.

Contrarily to the previous sequential tracking experiment, each point cloud is here considered in isolation. The evaluation of the robustness domain of both algorithms is performed following a Monte-Carlo approach. For each point cloud chosen at random, a random initial pose is chosen, with a maximal initial position error of 50cm, and a maximal initial angular error of 30°. Both ICP and smoothed NDT are then run with this initial pose, and a decision is made on whether each algorithm was successful based on following criteria: If the final translation error is below 7.5cm and the angular error below $3.5°$, the registration is considered successful. In total, $50,000$ random experiments were performed for each algorithm.

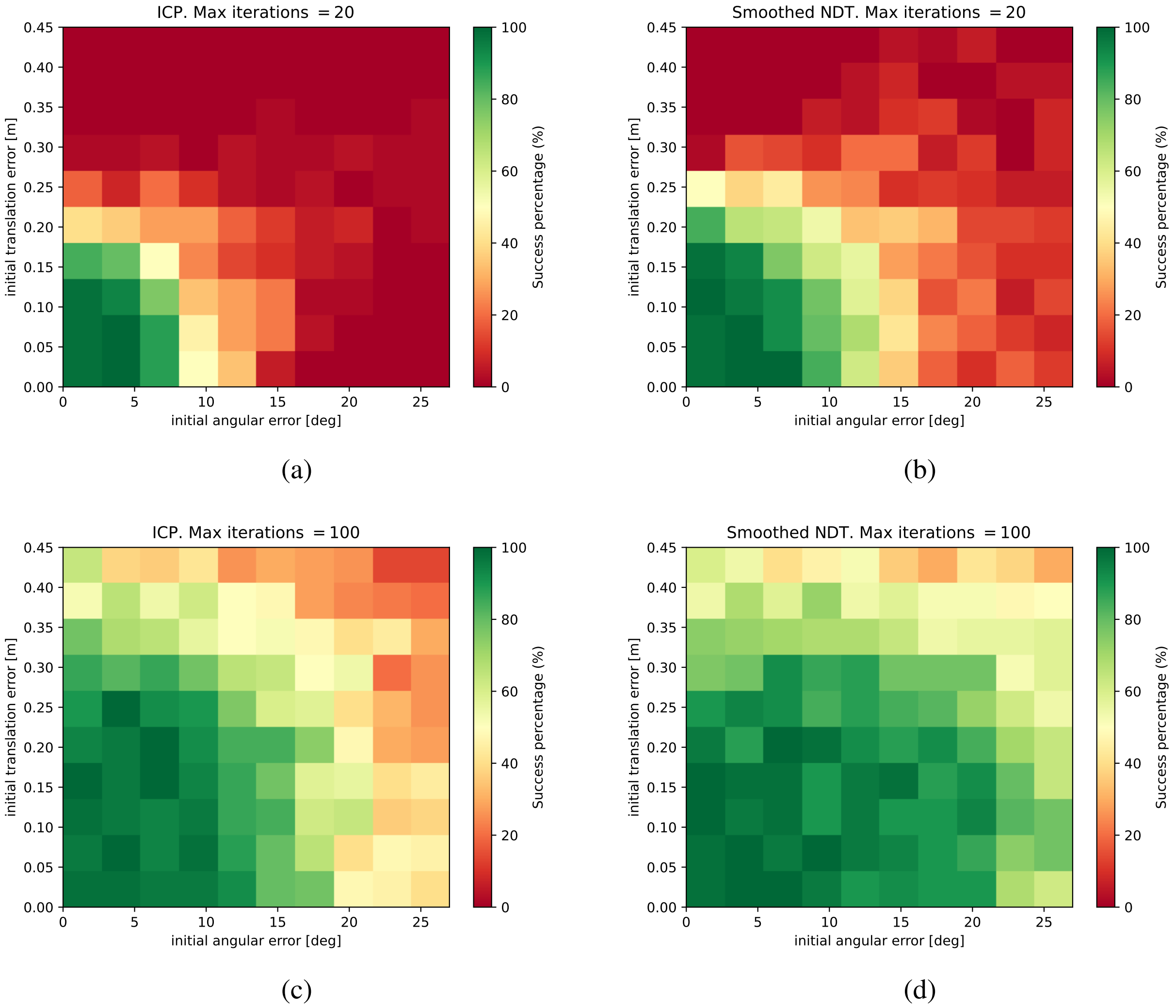
The results of this Monte-Carlo evaluation of the convergence domain of each algorithm are presented on Fig. 6. The convergence domain is represented on a 2D heatmap, where the magnitude of the initial angular and translation errors form both axes of the map, while the colors represent the fraction of registrations that were successful for each bin. When the initialization error is important, both ICP and smoothed NDT algorithms need more than 20 iterations to converge. This is why the results are presented with two different maximum number of iterations: Figs. 6a and 6b show the results obtained when setting the maximum number of iterations of both algorithms to 20. Figs 6c and 6d show the same results when the maximum number of iterations is 100.

For a maximum number of iterations of 20, both algorithms show a low success rate for initialization errors above 15° and 30cm (Figs. 6a and 6b). However, the results indicate that the convergence domain of smoothed NDT is more important than the one of ICP, with a higher proportion of total successful registrations. Setting the maximum number of iterations to 100 leads to an important increase in the convergence domain of both methods (Figs. 6c and 6d). Again, smoothed NDT has a bigger convergence domain than ICP, with in particular more tolerance for initial angular errors.

These results indicate that smoothed NDT is a robust registration methods, which requires less iterations than ICP to converge towards the correct result. In addition, this experiment can be used to define the convergence domain of smoothed NDT for the development of a pose initialization method. When the maximum number of iterations is set to 100, Fig. 6d indicates that initial errors up to 20° and 25cm still lead to a success probability of the method above roughly 80%.

5 CONCLUSION

It can be challenging to track the pose of a spacecraft on consecutive lidar point clouds in real-time when the point clouds are dense. Therefore, we investigated the use of NDT, an efficient alternative



(a)

(b)

(c)

(d)

Figure 6: Monte-Carlo evaluation of the convergence domain of both algorithms. Different maximum number of iterations max_iter were set. Green colors indicates high success percentages, red colors low success. **(a)** ICP convergence domain for $max\_iter = 20$. **(b)** Smoothed NDT convergence domain for $max\_iter = 20$. **(c)** ICP convergence domain for $max\_iter = 100$. **(d)** Smoothed NDT convergence domain for $max\_iter = 100$.

to ICP, for the problem of uncooperative spacecraft pose estimation. A smoothing method of the NDT map was introduced in order to increase the robustness of the algorithm. Lidar point clouds also present motion blur when the observed target spacecraft is tumbling. This work proposed a motion compensation technique used in conjunction with a Kalman filter. In consequence, recorded lidar point clouds are deblurred prior to being processed by the registration algorithm.

The modified smoothed NDT algorithm was tested at the European Proximity Operations Simulator, and compared to ICP. For the experiment consisting in tracking the pose of a spinning satellite, smoothed NDT shows better precision than ICP in terms of angular and translational error. The proposed method is also significantly faster than ICP, by a factor 3 to 4. To isolate the impact of motion compensation, a test without this strategy was run. This test indicates that the application of the deblurring method leads to a precision improvement in the tracking of a few degrees in terms of angular error. To compare the convergence basin of ICP and smoothed NDT, a Monte-Carlo evaluation was performed on a second EPOS dataset. This evaluation shows that smoothed NDT is more robust than ICP to an error in the initial estimate of the relative pose.

The conducted tests demonstrate that smoothed NDT is a precise, efficient and robust registration method for pose tracking of an uncooperative spacecraft using a scanning lidar. In addition, lidar motion blur can be effectively mitigated by the application of the proposed deblurring strategy. However, for a rendezvous mission, a pose initialization method is needed in order to initialize the tracking. Further work will focus on the development of an initialization technique, as well as the evaluation of the proposed method with computing hardware and sensors that are realistic for space use.

REFERENCES

[1] L. P. Cassinis, R. Fonod, and E. Gill, "Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft," *Progress in Aerospace Sciences*, vol. 110, p. 100 548, 2019. DOI: 10.1016/j.paerosci.2019.05.008.

[2] K. Klionovska and M. Burri, "Hardware-in-the-loop simulations with umbra conditions for spacecraft rendezvous with pmd visual sensors," *Sensors*, vol. 21, no. 4, p. 1455, 2021. DOI: 10.3390/s21041455.

[3] D. Sun, L. Hu, H. Duan, and H. Pei, "Relative pose estimation of non-cooperative space targets using a tof camera," *Remote Sensing*, vol. 14, no. 23, p. 6100, 2022. DOI: 10.3390/rs14236100.

[4] S. Ruel, T. Luu, and A. Berube, "Space shuttle testing of the tridar 3d rendezvous and docking sensor," *Journal of Field robotics*, vol. 29, no. 4, pp. 535–553, 2012. DOI: 10.1002/rob.20420.

[5] J. O. Woods and J. A. Christian, "Lidar-based relative navigation with respect to non-cooperative objects," *Acta Astronautica*, vol. 126, pp. 298–311, 2016. DOI: 10.1016/j.actaastro.2016.05.007.

[6] F. Yin, W. Chou, Y. Wu, G. Yang, and S. Xu, "Sparse unorganized point cloud based relative pose estimation for uncooperative space target," *Sensors*, vol. 18, no. 4, p. 1009, 2018. DOI: 10.3390/s18041009.

[7] W. Guo, W. Hu, C. Liu, and T. Lu, "Pose initialization of uncooperative spacecraft by template matching with sparse point cloud," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 9, pp. 1707–1720, 2021. DOI: 10.2514/1.G005042.

[8] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, Spie, vol. 1611, 1992, pp. 586–606. DOI: 10.1117/12.57955.

[9] G. Zhao, S. Xu, and Y. Bo, "Lidar-based non-cooperative tumbling spacecraft pose tracking by fusing depth maps and point clouds," *Sensors*, vol. 18, no. 10, p. 3432, 2018. DOI: 10.3390/s18103432.

[10] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, IEEE, vol. 3, 2003, pp. 2743–2748. DOI: 10.1109/IROS.2003.1249285.

[11] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3d registration reliability and speed-a comparison of icp and ndt," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 3907–3912. DOI: 10.1109/ROBOT.2009.5152538.

[12] S. Pang, D. Kent, X. Cai, H. Al-Qassab, D. Morris, and H. Radha, "3d scan registration based localization for autonomous vehicles-a comparison of ndt and icp under realistic conditions," in *2018 IEEE 88th vehicular technology conference (VTC-Fall)*, IEEE, 2018, pp. 1–5. DOI: 10.1109/VTCFall.2018.8690819.

[13] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992. DOI: 10.1016/0262-8856(92)90066-C.

[14] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *2002 International Conference on Pattern Recognition*, IEEE, vol. 3, 2002, pp. 545–548. DOI: 10.1109/ICPR.2002.1047997.

[15] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Pose estimation for spacecraft relative navigation using model-based algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 431–447, 2017. DOI: 10.1109/TAES.2017.2650785.

[16] L. Liu, G. Zhao, and Y. Bo, "Point cloud based relative pose estimation of a satellite in close range," *Sensors*, vol. 16, no. 6, p. 824, 2016. DOI: 10.3390/s16060824.

[17] Y. Li, Y. Wang, and Y. Xie, "Using consecutive point clouds for pose and motion estimation of tumbling non-cooperative target," *Advances in Space Research*, vol. 63, no. 5, pp. 1576–1587, 2019. DOI: 10.1016/j.asr.2018.11.024.

[18] Q. Wang, T. Lei, X. Liu, *et al.*, "Pose estimation of non-cooperative target coated with mli," *IEEE Access*, vol. 7, pp. 153 958–153 968, 2019. DOI: 10.1109/ACCESS.2019.2946346.

[19] R. Opromolla and A. Nocerino, "Uncooperative spacecraft relative navigation with lidar-based unscented kalman filter," *IEEE Access*, vol. 7, pp. 180 012–180 026, 2019. DOI: 10.1109/ACCESS.2019.2959438.

[20] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007. DOI: 10.1002/rob.20204.

[21] C. Schulz, R. Hanten, and A. Zell, "Efficient map representations for multi-dimensional normal distributions transforms," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2679–2686. DOI: 10.1109/IROS.2018.8593602.

[22] C. Ulaş and H. Temeltaş, "3d multi-layered normal distribution transform for fast and long range scan matching," *Journal of Intelligent & Robotic Systems*, vol. 71, pp. 85–108, 2013. DOI: 10.1007/s10846-012-9780-8.

[23] L. Renaut, H. Frei, and A. Nüchter, "Lidar pose tracking of a tumbling spacecraft using the smoothed normal distribution transform," *Remote Sensing*, vol. 15, no. 9, p. 2286, 2023. DOI: 10.3390/rs15092286.

[24] L. Renaut., H. Frei., and A. Nüchter., "Smoothed normal distribution transform for efficient point cloud registration during space rendezvous," in *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, SciTePress, 2023, pp. 919–930. DOI: 10.5220/0011616300003417.

[25] A. Barrau and S. Bonnabel, "Invariant kalman filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018. DOI: 10.1146/annurev-control-060117-105010.

[26] H. Benninghoff, F. Rems, E.-A. Risse, and C. Mietner, “European proximity operations simulator 2.0 (epos)-a robotic-based rendezvous and docking simulator,” *Journal of large-scale research facilities JLSRF*, 2017. DOI: `10.17815/jlsrf-3-155`.

[27] 3DTK, *The 3D toolkit*, [Online]. Available: `https://slam6d.sourceforge.io/index.html`, 2023.