

Simple But Effective Redundant Odometry for Autonomous Vehicles

Andrzej Reinke

Xieyuanli Chen

Cyrill Stachniss

Abstract—Robust and reliable ego-motion is a key component of most autonomous mobile systems. Many odometry estimation methods have been developed using different sensors such as cameras or LiDARs. In this work, we present a resilient approach that exploits the redundancy of multiple odometry algorithms using a 3D LiDAR scanner and a monocular camera to provide reliable state estimation for autonomous vehicles. Our system utilizes a stack of odometry algorithms that run in parallel. It chooses from them the most promising pose estimation considering sanity checks using dynamic and kinematic constraints of the vehicle as well as a score computed between the current LiDAR scan and a locally built point cloud map. In this way, our method can exploit the advantages of different existing ego-motion estimating approaches. We evaluate our method on the KITTI Odometry dataset. The experimental results suggest that our approach is resilient to failure cases and achieves an overall better performance than individual odometry methods employed by our system.

I. INTRODUCTION

The ability to estimate the ego-motion, also called odometry, is a vital part of most autonomous mobile systems. Nowadays, autonomous vehicles are typically equipped with multiple sensors, such as cameras or LiDARs, as different sensing modalities have individual strengths. Therefore, a large number of algorithms [35], [14], [15] have been proposed exploiting both, visual and LiDAR information to explore the advantages of both and compensate for the drawbacks of the other. In most cases, researchers focus on achieving better odometry results by designing a potentially complex multi-sensor fusion algorithm. Even if different pose estimation systems' performances are remarkable within specific datasets, such algorithms may not always operate reliably under all conditions, setups, and application domains, as we will illustrate below in Sec. IV-C. Thus, failure detection and recovery steps are useful in real-world applications. We investigate in this paper a strategy that runs multiple odometry approaches in parallel and then selects the method that appears to be the best one. Thus, newly proposed individual methods can easily be added to our approach.

Few studies are investigating redundant odometry methods running in parallel on autonomous mobile systems. One example is the work by Luo et al. [18], who propose to estimate odometry as an average of parallel odometry pipelines. This approach, however, is vulnerable since already one gross estimation error of one method will affect the

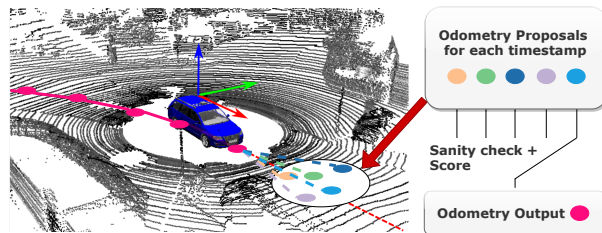


Fig. 1: Our redundant odometry system employs multiple odometry algorithms to provide proposals shown as differently colored dots for the each timestamp, exploiting LiDAR and camera data. Our approach uses sanity checks and a Chamfer distance-based score to choose the most promising pose estimation from all candidates.

overall performance. Santamaria-Navarro et al. [27] propose another solution, which runs several methods in parallel. It first sticks to one odometry method and only falls back to an alternative odometry pipeline if the initial method's divergence is detected. While this method is straightforward to realize, it does not utilize to its full extent the potential of all algorithms that run in parallel.

In this work, we investigate the possibility of building a redundant odometry system with sanity checks, and that can deal with failure cases of different odometry methods. As sketched in Fig. 1, instead of sticking to a specific odometry method, the idea is to run various odometry algorithms in parallel and choose the most promising one. We target to avoid explicit recovery behaviors for individual odometry pipelines in case of failures but propose to switch dynamically between estimation algorithms and independently of previous decisions. Furthermore, we target a computationally light-weight and flexible system.

The main contribution of this paper is a redundant odometry system to provide a more resilient ego-motion estimation than the individual odometry methods that the system consist of. Instead of sticking to one specific odometry method, the proposed system runs multiple odometry algorithms in parallel and exploits LiDAR and camera data. It combines sanity checks exploiting dynamic and kinematic constraints of the vehicle and Chamfer distance-based scores between sensor data and a local map to select the most promising pose estimation among all the odometry candidates. We make three key claims: Our approach is able to (i) use redundant odometry pipelines to provide better odometry results, (ii) generally yields the best results compared to the individual methods used, and, (iii) avoids several failure cases in different situations. The paper and our experimental evaluation back up these claims. Furthermore, the source code of our approach is available at:

<https://github.com/PRBonn/MutiverseOdometry>

All authors are with the University of Bonn, Germany.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob) and by the European Unions Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

II. RELATED WORK

Odometry estimation is a widely investigated topic in robotics. It can be achieved using different types of sensors, including wheels [32], inertial sensing [13], cameras [28], thermal cameras [9], radars [7], LiDARs [26], sonars [1] as well as various combinations [11], [34], [35], [8], [3], [20]. Although much work has been done, only a small amount of research has been devoted to making a robust and redundant system to deal with failures of individual ego-motion estimation approaches.

Works show that redundancy can make a robotic system more robust. Rollinson et al. [25] show a robust estimation of the configuration of an articulated robot that uses a large number of redundant proprioceptive sensors (encoders, gyros, accelerometers) embedded in an unscented Kalman filter. They formulate the state estimation problem to leverage the redundancy in the proprioceptive information provided by the robot's joint angle encoders and inertial sensors. Additionally, they introduce a novel outlier detector that can identify corrupted measurements by utilizing the Mahalanobis distance matrix. Kim et al. [16] point out that redundant systems are important due to adversary attacks that corrupt measurement. They propose an observer for a nonlinear system to detect sensor attacks.

The importance of a redundant odometry system has also been shown on Mars Exploration Rovers [19]. It successfully demonstrated a switching behavior between the wheel, inertia, and visual odometry, where visual odometry was used as compensation for any unforeseen slip encountered during a drive. Another redundant odometry system was proposed by Luo et al. [18] taking an average estimate over all odometry estimates that are currently in the active mode. Others use fallback solutions and execute homing behaviors [24], sampling-based motion proposals in a particle filter [30], or combinations of dense and feature-based methods [33]. Carnevale et al. [6] propose a way to fuse the information coming from two typologies of redundant radio-frequency identification sensors with complementary characteristics using a Kalman filter framework. A switching criterion is embedded in the filtering approach where the observer uses uncertainties of the measurements and their average values to provide a switching behavior. Recently, Ebadi et al. [10] created an autonomous mapping and positioning system for the DARPA Subterranean Challenge. This work uses visual-inertial, LiDAR, and wheeled odometry as a front-end for a multi-robot mapping and positioning system to explore perceptually-degraded environments. They use the generalized iterative closest point (GICP) algorithm [29] to estimate scan-to-scan and scan-to-map matching that provides the relative transformation between the consecutive timestamps.

The related approach to our work has recently been proposed by Santamaria-Navarro et al. [27] in the context of UAVs. Their system called heterogeneous redundant odometry (HeRO) evaluates the pose estimations from each pipeline by checking gaps and jumps within the transforms sequence. In this method, the different odometries run in parallel,

and the potentially best estimation is used as the output of the system until the uncertainty of the estimation exceeds certain quality criteria. Then, the system falls back to another method while at the same time trying to reinitialize the former method. Unlike HeRO, which treats each odometry pipeline as a separate entity with a separated history of transformations and transformed point clouds, our method does not need any reinitialization procedure. It switches to an alternative method or fallbacks to a constant velocity model if all other methods fail. We focus on picking the best transform independently from the previous transforms and use a chain of transformation and transformed point clouds as a common history. HeRO assesses the best odometry based on the internal state of the odometry algorithms, while our method aims at finding a more objective metric that can choose fairly the best odometry at every timestamp. Based on HeRO, Palieri et al. [22] propose a health monitor system for the odometry estimation, which exploits LiDAR-based scan-to-scan and scan-to-map methods. While this method is similar to our approach, we do not need to manually specify the main odometry or submental ones. Our method chooses the most promising odometry by each candidate's performance scores at every timestamp without relying on any prior assumptions or knowledge about the odometry performance.

Different from all the methods mentioned above, we propose an integrated visual-LiDAR odometry system that uses multiple odometry algorithms in parallel, a set of dynamics and kinematic sanity checks that filter out potential estimation failures, and a point cloud Chamfer distance-based criterion, which chooses the most promising pose estimation at each timestamp, independently from the history.

III. OUR APPROACH

We propose a redundant odometry system enhanced with sanity checks. These steps are visualized in Fig. 2. We use the modular design to keep our system flexible. There are four modules in our system. First, the input preprocessing module exploiting multiple sensing modalities as the input to the system (see Sec. III-A). Second, the odometry candidates module employing multiple different odometry algorithms to estimate the candidate transformations at each timestamp as odometry proposals (see Sec. III-B). Third, the sanity-check module using vehicle dynamics and kinematics properties to check for failure cases in each odometry proposal (see Sec. III-C). Fourth, the scoring module exploiting the Chamfer distance to score the odometry proposals and selects the final transformation from the most recent vehicle motion (see Sec. III-D).

A. Input Preprocessing Module

In this work, we do not focus on choosing the best sensor or combination to improve odometry but provide a flexible system that allows us to use multiple sensing modalities and ego-motion algorithms in parallel to improve the robustness of the odometry. To this end, our system uses the data from a 3D LiDAR scanner and a monocular camera as an example to

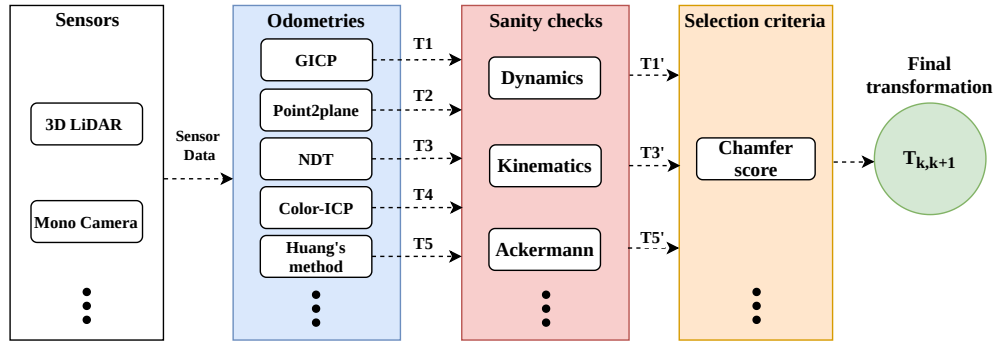


Fig. 2: System overview. Both 3D LiDAR and camera sensor data are used as the input. We first exploit multiple existing odometry algorithms to process the input data and provide transformation proposals. Those proposals will then be checked by sanity checks based on dynamics and kinematics capabilities. Only those transformations which are labeled as success cases will be evaluated in the next step. The point-to-point Chamfer distance is then calculated to score those success cases. As the result, the best transformation in terms of the calculated scores will be chosen as the final estimation. The final transformation will be saved to the odometry history and employed as an initial guess for the next iteration. Our system is flexible, every submodule could be extended by extra data, methods, or other criteria.

show that our system can exploit the redundancy of multiple sensing modalities to improve its robustness. Specifically, we use SIFT features [31] extract from images obtained by a monocular camera and voxelized point clouds from LiDAR scans as the input to different odometry methods. Such a redundancy design at the input level has the advantage that our approach is still functional even in case of sensor failures.

B. Stack of Odometry Pipelines

Our approach combines multiple odometry methods into a redundant system, which is robust to failure cases and provides better odometry results. The odometry algorithms used in this paper are generalized iterative closest point (GICP) [29], point-to-plane iterative closest point (P2P-ICP) [26], normal distributions transform (NDT) [4], Color iterative closest point (ColorICP) [23], and Huang's method [14] for combining LiDAR and camera data. Moreover, we also employ the common constant velocity model (CVM), which directly uses the last timestamp estimation without using the sensor data. Our approach is not limited only to those methods and can be easily extended.

P2P-ICP [26], [36] minimizes the sum of the squared distance between the point from the source points and the tangent plane at its corresponding point in the target point cloud for every point.

GICP [29], [17] combines the Iterative Closest Point (ICP) and point-to-plane ICP into a single probabilistic framework. It models the locally planar surface structure from both scans and minimizes the objective function in a typical ICP-based framework. This procedure may be interpreted as a plane-to-plane approach embedded in the ICP framework.

ColorICP [23], [36] optimizes jointly geometrical information (in the sense of point-to-plane ICP) together with RGB color information that is assigned to the LiDAR points which are inside the field of view of the camera(s).

NDT [4], [17] uses a representation of the target point cloud to obtain a smooth surface described as a set of local probability density functions in the subdivided space as a grid of cells. A density function is computed for each cell

based on the point distribution within the cell. To estimate the pose, NDT maximizes the likelihood that the current scan points lay on the reference scan surface.

Huang's method [14] employs the 5-point algorithm [21] within a RANSAC loop that finds 5 of the 6 transformation parameters (rotation and translation vector up to scale) between the previous camera image and the current image with the SIFT descriptor. It then uses a grid search-based 1D ICP to estimate the scale.

Besides the odometry information provided by different algorithms based on different sensing modalities, our system also employs the so-called constant velocity model (CVM), which takes the same transformation from the previous timestamp as an additional odometry estimation with the assumption that the acceleration of the car is zero. Although the constant velocity model is simple, it will always pass the sanity checks and make the system robust.

Those odometry methods are run in parallel using the same input data and share the same odometry history. This means only the transformation that is finally selected by our approach at each timestamp will be stored and serves as the initial guess for all the odometry methods in the next timestamp.

In this paper, we exploit six different odometry methods to exploit the redundancy of multiple odometries to improve overall performance and robustness. Note that we can easily add more methods if desired. Since our method can switch dynamically for each timestamp independently from previous decisions, multiple odometry methods can be easily integrated into our system, even during online processing.

C. Sanity Checks

In robotics, we typically know the used hardware and the physical model of our robot or a typical car. Therefore, we use the dynamic and kinematic constraints of a car to check the odometry proposals provided by different algorithms, which we refer to as sanity checks. In this paper, we exploit two types of sanity checks, one based on the dynamic constraint, and the other based on the kinematic constraint.

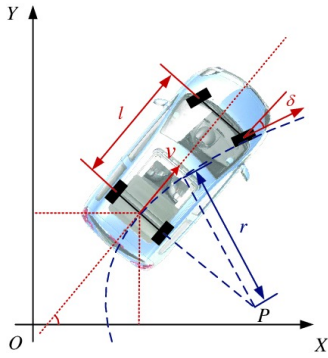


Fig. 3: Ackermann vehicle model. The variable l is the distance between the front and rear axis, v is the velocity of the rear axis center, δ is the steering angle, P is the instantaneous center of rotation, and r is the radius around P . See [37] for details.

We refer to the maximum acceleration a_{max} a typical car can reach for formulating the dynamic constraint. Our system first checks the acceleration resulting from all odometry proposals. If the estimated transformation exceeds the bound of the maximum possible acceleration, it will be rejected from further considerations.

For the kinematic constraint, we use the Ackermann vehicle model [37], which puts constraints on the kinematic possibilities of the car movement. If using a differential drive model, one obviously must use the corresponding model. As shown in Fig. 3, given the forward velocity v and the steering angle rate $\dot{\delta}$, the car's trajectory follows the Ackermann model if there is no slip or any other highly dynamic conditions such as drifting or similar. It means that in real operations, the car's sideward movements are limited to a certain range.

Applying this fact, our system uses the last pose estimation to calculate the forward velocity. Based on the forward velocity and the Ackermann model, the side velocity v_s is calculated based on the derivation [5],

$$v_s(\dot{\beta}, v) = f \frac{\frac{v}{f} + l(1 - \cos \frac{\dot{\beta}}{f})}{\sin \frac{\dot{\beta}}{f}} (1 - \cos \frac{\dot{\beta}}{f}) + l \sin \frac{\dot{\beta}}{f}, \quad (1)$$

where v is a forward velocity of the vehicle, and $\dot{\beta}$ is the turning rate, $f = \frac{1}{\Delta\tau}$, where $\Delta\tau$ is the difference between timestamps of the current frame and previous frame, l is the distance from the rear axis to the placement of the sensor. A proposal will then be rejected for future considerations if

$$|v_s - v^*| > v_{th}, \quad (2)$$

where v^* is taken from the odometry proposals and v_{th} is a maximum deviation from the side velocity that is acceptable.

Note that even if all sensors or odometry algorithms based on sensor data fail to pass the sanity checks, the constant velocity model will always pass the sanity checks and makes the proposed method fairly robust to failures.

D. Scoring Candidate Transformations

All the odometry proposals, which pass the sanity checks, will be evaluated based on the Chamfer distance metric [2]

computed on the point cloud. Thus we need at least one sensor that generates point cloud data. We use a 3D LiDAR in our system. The metric on the point cloud is defined as follow,

$$D_{\text{chamfer}}(M, N) = \frac{1}{M} \sum_{m \in M} d_N(m), \quad (3)$$

where M is a set of points in the source point cloud, N is a set of points in the target point cloud, and d_N is the minimum distance between point m and its nearest neighbor point in the target point cloud N given the transformation between point cloud M and N . In addition to the fact that point clouds after transformation should close to each other, we apply a small search radius r_s for correspondence matching and delete the points for which no correspondence could be found within the radius. The reason to use the Chamfer distance as the criterion is that it is fast to calculate and independent from all the baseline odometry methods.

Since each odometry candidate in our system estimates the transforms from the same source point cloud to the same target point cloud, this metric allows us to assess the candidate transformations estimated by different odometry methods. Our system applies this metric to all the success odometry proposals, and the lowest score is chosen as the final odometry result.

$$T^* = \arg \min_{T \in T_1, T_2, \dots} \frac{1}{M} \sum_{m \in M} d_N(m), \quad (4)$$

The chosen transformation is stored in the odometry history and employed as the initial guess for the next iteration. We use the last N_{map} scans with the stored odometry to build a local point cloud map.

IV. EXPERIMENTAL EVALUATION

We present our experiments to showcase the capabilities of our method and to support our key claims that our approach is able to (i) use redundant odometry pipelines to provide better odometry results, (ii) generally yields the best results compared to the individual methods used, and, (iii) avoids several failure cases in different situations. For evaluation purposes, we use the KITTI Odometry Benchmark [12] which provides point clouds generated by a Velodyne HDL-64E 3D laser scanner (10 Hz, 64 laser beams, range: 100m) and color images from PointGrey Flea2 video cameras (10 Hz, resolution: 1382×512 pixels). We tested our approach on an Intel Core i9-9900K CPU with 16 cores @ 3.60GHz with 32 GB RAM. For all the experiments, we use $l = 1.0m$, $a_{\text{max}} = 6.0m/s^2$ and $v_{th} = 0.8m/s$, the search radius $r_s = 0.5m$, and the number of scans used to build the local map is $N_{\text{map}} = 10$.

A. Odometry Performance

The experiments presented in this section are designed to show our approach's odometry performance and support the claim that using our redundant system yields better overall odometry results.

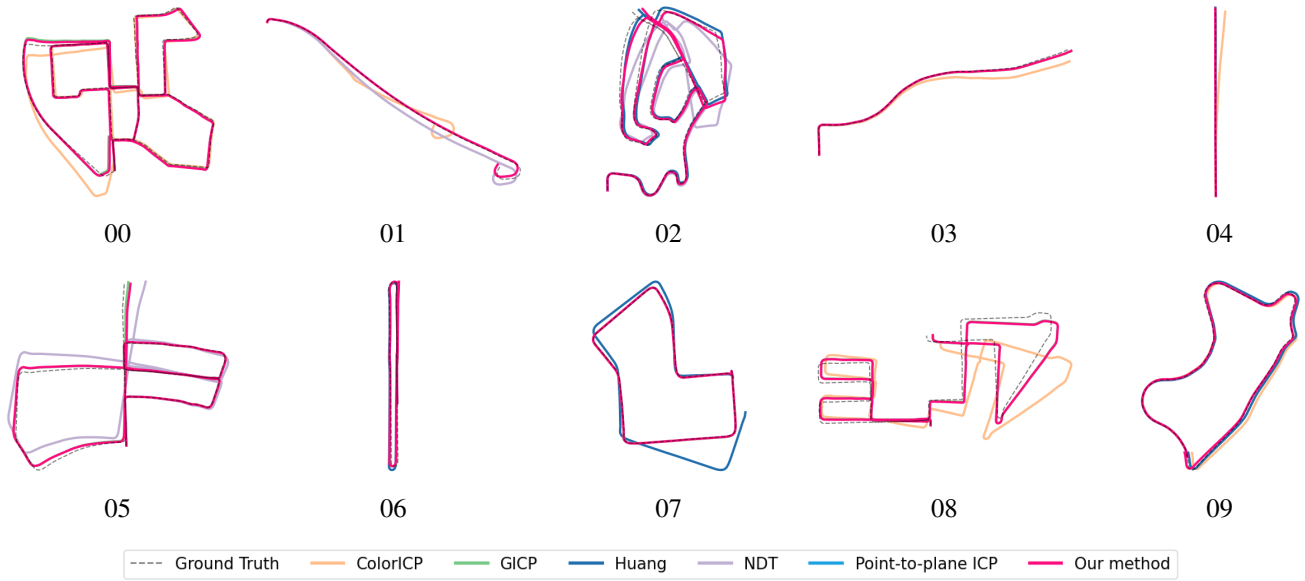


Fig. 4: Trajectories of different methods evaluated on the KITTI Dataset. For the sake of clarity, we only show the results of our method together with the best and the worst results of the baseline methods employed by our system.

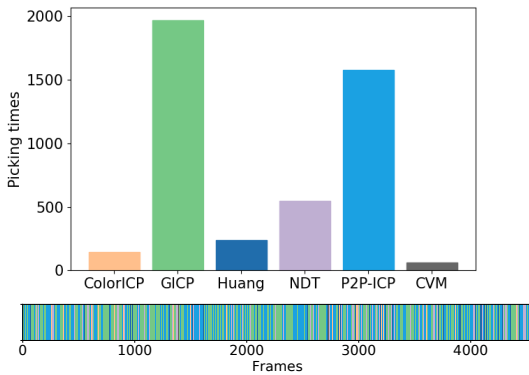


Fig. 5: Frequency of choosing different baseline methods in KITTI sequence 00. The upper figure shows the picking times of different methods, and the lower figure shows the choice of each frame.

To evaluate the odometry results, we use the KITTI odometry dataset sequences 00-10 and the KITTI odometry metric [12], which considers relative errors concerning translation and rotation averaged over different distances.

For the qualitative results, we show the estimated trajectories of our method compared to those generated by the individual odometry algorithms employed by our redundant system, see in Fig. 4. For the sake of clarity, we only show the trajectories generated by our method as well as those of the best and the worst individual odometry. We can see that, in all sequences, our method always tracks the ground truth poses well no matter how bad the worst individual odometry candidate is, and there is no individual odometry baseline that can always generate good results.

For the quantitative results, we show all evaluation results on all sequences in Tab. I. In terms of translation error, our method is clearly superior in 10 out of 11 sequences to all other methods individually and in the other one

sequence achieves a comparable performance to the best individual odometry algorithm. Overall, our method outperforms all baselines with the smallest average translation error of 1.11 %. In terms of rotation error, our method achieves the second-best average rotation error of 0.0045 deg/m.

To further insight into how our system chooses different proposals, we illustrate when and how often different methods have been selected on sequence 00 in Fig. 5. We can see that our method switches between different proposals frequently according to the sanity checks and scoring results.

B. Ablation Study on Odometry Candidates

The second experiment shows the ablation study on different odometry combinations. We use the same KITTI odometry dataset sequence 00-10 and the same metric as in the first experiment to evaluate the odometry results of different odometry input combinations. As shown in Tab. II, we can see that no matter what combination of the odometry candidate methods is provided, our system can always outperform the individual odometry methods (see results in Tab. I for the individual numbers per sequence).

C. Robustness

The third experiment illustrates our system's robustness and supports the claim that the proposed redundant odometry system is robust to different failure cases. We use four different cases to show that our system can successfully cover the failure cases where the individual candidate odometry methods fail, see Fig. 6. The failure cases refer to the situations where the odometry yields a clearly wrong estimate with respect to the ground truth poses. The failure cases can be caused by the dynamic objects, the lack of features of the environments, or the wrong data associations between different observations. In Fig. 6, we can see that in all cases when the individual entities fail, our system switches in

TABLE I: Results on KITTI Odometry

Approach	00	01	02	03	04	Sequence 05	06	07	08	09	10	Average
GICP	1.28/0.61	3.22/0.91	1.78/0.63	1.18/0.77	1.51/0.93	1.18/0.60	0.87/0.53	0.87/0.57	1.32/0.56	1.55/0.65	1.65/0.69	1.49/0.62
Huang's	2.40/0.33	6.09/0.21	1.50/0.25	1.38/0.18	0.98/0.22	2.06/0.42	1.43/0.25	6.86/4.04	2.71/0.29	1.37/0.22	2.37/0.24	2.35/0.38
P2P-ICP	1.57/0.77	5.22/1.56	1.58/0.58	1.98/1.13	1.05/0.56	1.32/0.76	1.12/0.76	0.95/0.62	1.37/0.76	1.48/0.73	1.48/0.70	1.64/0.75
ColorICP	3.09/1.60	26.08/1.68	3.17/1.45	2.93/1.40	3.11/2.00	1.74/0.99	0.99/0.63	1.60/1.51	3.14/1.65	2.44/1.33	4.04/1.66	3.89/1.44
NDT	2.88/1.66	2.42 /0.67	3.51/1.58	2.89/1.89	1.63/1.39	2.58/1.33	1.27/0.67	3.38/2.52	2.76/1.43	2.64/1.16	2.91/1.60	2.88/1.46
Our method	0.96 /0.43	2.53/0.67	1.40 /0.50	1.05 /0.67	0.97 /0.61	0.72 /0.35	0.66 /0.36	0.62 /0.45	0.99 /0.39	0.99 /0.44	1.30 /0.54	1.11 /0.45

Relative errors averaged over trajectories of 100 to 800m length: relative translational error in % / relative rotational error in degrees per 100m. Bold numbers indicate best performance in terms of translational error.

TABLE II: Ablation study on input odometry methods.

P2P-ICP	NDT	ColorICP	Huang's	GICP	Average
✓	✓				1.23/0.54
✓	✓	✓			1.21/0.52
✓	✓	✓	✓		1.17/0.50
✓	✓	✓	✓	✓	1.11/0.45

Here we use the same KITTI metric as used in Tab. I. We show the average results of different setups evaluated with all sequences.

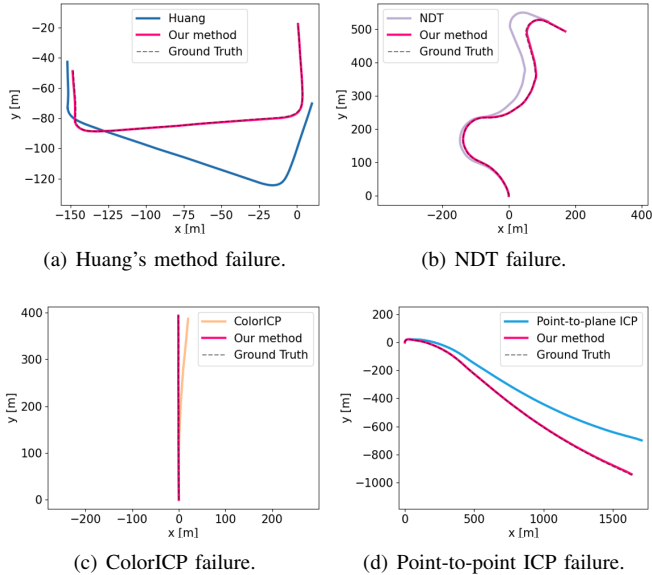


Fig. 6: Robustness evaluation of our system. Here we show failure cases of different methods employed by our system (Huang's method, NDT, ColorICP and Point-to-plane ICP respectively). We see that in all failure cases our system work properly.

between the odometry candidates at the right time, choosing the correct estimate and provide proper odometry results.

To gain further insight, we provide an analysis of the sanity checks. If we do not apply sanity checks, our KITTI 00-10 overall score drops from 1.11 % to 1.37 % in terms of the average translation error. The results of the acceleration sanity checks on KITTI sequence 01 is shown in Fig. 7. We can see in Fig. 7(a) that there are some large accelerations that indicate the failures of the odometry estimations. After using sanity checks, see Fig. 7(b), our system provides smoother velocity estimates, see Fig. 7(c), and provide better overall results, see Fig. 7(d).

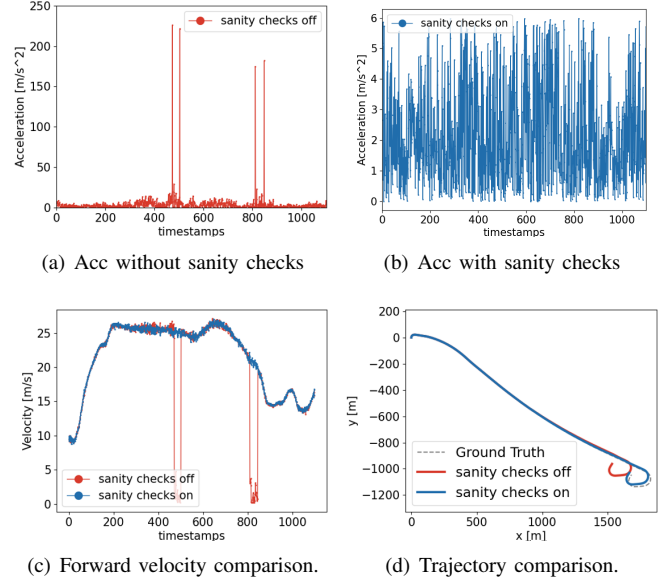


Fig. 7: Analysis of sanity checks for our system on dataset 01. (a) shows the noise of acceleration estimation. (b) shows the results after our approach filter out the noise using sanity checks. (c) shows that with sanity checks our method can generate smoother velocity and provide also a better trajectory shown in (d).

V. CONCLUSION

In this paper, we presented a robust and resilient odometry system. Our method is simple, effective, easy to implement, and we provide the source code for further use. Our system exploits the redundancy of running multiple odometry pipelines, including multiple streams, in parallel. It uses the dynamic and kinematic constraints-based sanity checks and the Chamfer distance-based criteria to avoid failure cases and select the most promising odometry estimation among all proposals. For that, at least one sensor that generates point clouds, such as a LiDAR, is required. We evaluated our approach on the KITTI dataset and supported all claims made in this paper. The experiments suggest that the proposed system is resilient and robust to odometry failures and achieves better performance than all the baselines in terms of odometry accuracy. Despite these encouraging results, there is space for further improvements. In future work, we plan to use learning-based approaches to design better selection criteria for choosing the most promising odometry estimate.

REFERENCES

- [1] A. Bahr, J.J. Leonard, and M.F. Fallon. Cooperative localization for autonomous underwater vehicles. *Intl. Journal of Robotics Research (IJRR)*, 28(6):714–728, 2009.
- [2] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI)*, pages 659–663, 1977.
- [3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.
- [4] P. Bieber and W. Strasser. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743 – 2748, 11 2003.
- [5] M. Buczko, V. Willert, J. Schwehr, and J. Adamy. Self-validation for automotive visual odometry. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, 2018.
- [6] D. Carnevale and F. Martinelli. State estimation for robots with complementary redundant sensors. *Intl. Journal of Advanced Robotic Systems*, 12(10):138, 2015.
- [7] S.H. Cen and P. Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.
- [8] X. Chen, A. Milioto, E. Palazzolo, P. Gigore, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.
- [9] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies. Thermal-inertial odometry for autonomous flight throughout the night. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1122–1128, 2019.
- [10] K. Ebadi, Y. Chang, M. , A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, L. Carlone, and A.A. Agha-mohammadi. Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 80–86, 2020.
- [11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *Intl. Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013.
- [12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.
- [13] P. Groves. Navigation using inertial sensors [tutorial]. *IEEE Aerospace and Electronic Systems Magazine*, 30:42–69, 02 2015.
- [14] K. Huang and C. Stachniss. Joint Ego-motion Estimation Using a Laser Scanner and a Monocular Camera Through Relative Orientation Estimation and 1-DoF ICP. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.
- [15] K. Huang and C. Stachniss. Accurate Direct Visual-Laser Odometry with Explicit Occlusion Handling and Plane Detection. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.
- [16] J. Kim, C. Lee, H. Shim, Y. Eun, and J.H. Seo. Detection of sensor attack and resilient state estimation for uniformly observable nonlinear systems having redundant sensors. *IEEE Trans. on Automatic Control*, 64(3):1162–1169, 2019.
- [17] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. Voxelized gicp for fast and accurate 3d point cloud registration. EasyChair Preprint no. 2703, EasyChair, 2020.
- [18] R.C. Luo, C.C. Yih, and K.L. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, 2002.
- [19] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics (JFR)*, 24(3):169–186, 2007.
- [20] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.
- [21] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(6):756–770, 2004.
- [22] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A. Agha-mohammadi. Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robotics and Automation Letters (RA-L)*, 2020.
- [23] J. Park, Q. Zhou, and V. Koltun. Colored Point Cloud Registration Revisited. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.
- [24] D. Perea-Ström, I. Bogoslavskyi, and C. Stachniss. Robust Exploration and Homing for Autonomous Robots. In *Journal on Robotics and Autonomous Systems (RAS)*, 2016.
- [25] D. Rollinson, H. Choset, and S. Tully. Robust state estimation with redundant proprioceptive sensors. In *Proc. of ASME Dynamic Systems and Control Conference (DSCC)*, October 2013.
- [26] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. of the Intl. Conf. on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.
- [27] A. Santamaria-Navarro, R. Thakker, D.D. Fan, B. Morrell, and A. Agha-Mohammadi. Towards resilient autonomous navigation of drones. *arXiv preprint*, abs/2008.09679, 2020.
- [28] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics and Automation Magazine (RAM)*, 18(4):80–92, 2011.
- [29] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.
- [30] C. Stachniss, G. Grisetti, N. Roy, and W. Burgard. Evaluation of Gaussian Proposal Distributions for Mapping with Rao-Blackwellized Particle Filters. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.
- [31] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2531–2538, 2008.
- [32] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [33] K.M. Wurm, C. Stachniss, and G. Grisetti. Bridging the Gap Between Feature- and Grid-based SLAM. *Journal on Robotics and Autonomous Systems (RAS)*, 58(2):140 – 148, 2010.
- [34] J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.
- [35] J. Zhang and S. Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2174–2181, 2015.
- [36] Q.Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
- [37] W. Zong and Q. Chen. A robust method for detecting parking areas in both indoor and outdoor environments. *IEEE Sensors Journal*, 18(6):1903, 2018.