

# Combining LiDAR Scan Matching with Stereo Visual Odometry Using Curvefusion

Shitong Du, Xuyou Li

Department of Intelligent Science Systems and  
Engineering, Harbin Engineering University,  
Harbin, China  
e-mail: dushitong@hrbeu.edu.cn,  
lixuyou@hrbeu.edu.cn

Helge A. Lauterbach, Dorit Borrmann, Andreas  
Nüchter

Informatics VII: Robotics and Telematics,  
Julius-Maximilians-University Würzburg, Germany  
e-mail: {helge.lauterbach, dorit.borrmann,  
andreas.nuechter}@uni-wuerzburg.de

**Abstract**—In this paper, we present a novel algorithm, namely **Curvefusion for integrating LiDAR scan matching with stereo visual odometry**. First, 6-DOF pose trajectories are estimated by utilizing **SOFT** odometry, which is the state of the art stereo visual odometry based on feature selection and tracking, and the well-known ICP scan matching algorithm, respectively. Second, a deformation-based multi-sensor fusion method, namely **curvefusion** is applied. The proposed fusion method does not rely on a sensor model. As long as the trajectories of the sensors to be fused are given, we can obtain an optimized fusion trajectory, which greatly improves the computational efficiency. Experiments based on publicly available **KITTI** data set show that the proposed method outperforms or achieves similar performance compared with the state-of-the-art odometry methods.

**Keywords**—*curvefusion; SOFT; ICP; deformation-based; KITTI*

## I. INTRODUCTION

Autonomous mobile systems are widely applied in rescue, planetary exploration and autonomous driving. Localization and mapping are crucial technologies to intelligentize the robot. Simultaneous localization and mapping (SLAM) methods utilizing the LiDAR or camera to generate a globally consistent mapping and has been widely applied to the robotics community over the past decade [1]. A typical SLAM framework consists of the frontend and the backend. The front end aims to solve the data association and trajectory initialization. In the backend, a filtering or pose-graph-optimization framework is employed to further optimize the localization and mapping.

Recently, LiDAR and visual odometry have shown promising results in the 6-DOF trajectory estimation of the robot. Lidar or visual odometry often achieves accurate performance over a short distance with a few turns. However, large path even with many closed loops suffers from error accumulation and in practical applications the error grows quickly. To address this, multi-sensor fusion techniques are introduced into SLAM framework. Popular fusion methods include LiDAR, cameras and IMU. Since LiDAR and cameras have complementary strengths and weaknesses, the solution of combining LiDAR with vision cameras is often adopted [2].

In this paper, we present a novel visual-LiDAR fusion that compensates for weakness of single sensor. First, two 6-DoF pose trajectories from LiDAR and stereo visual sensor are obtained, respectively. The trajectory of LiDAR is estimated utilizing the well-known ICP (iterative closest point), while a stereo visual odometry based on feature selection and tracking algorithm (SOFT) [3] is used in visual estimation. Then, a deformation-based multi-sensor fusion method based on [4], [5] is introduced to combining LiDAR with Stereo visual odometry. However, previous work aimed to calculate shape similarity of two curves in computer vision, which is not well suitable for the trajectory fusion. To apply the previous work to our trajectory fusion, some problems have to be solved. Some extension works involving shape representation, full 6-DOF fusion are proposed. Details will be given in section III.

The remainder of the paper is organized as follows. In Section II, we summarize related works in LiDAR, visual odometry and SLAM. In Section III, the proposed algorithm is described in detail. Experimental results are presented in Section IV. The paper ends with conclusion in Section V.

## II. RELATED WORK

There is an increasing body of scholarly work regarding autonomous vehicles with different odometry and SLAM solutions. In this section, we present a brief literature review that is related to our current work.

Visual SLAM is divided into two categories: Feature based and direct methods. Feature based methods first extract the features of the input image with descriptors, such as SURF [6] and ORB [7]. Then, the transition matrix is calculated by matching the feature points between frames. Direct methods operate directly on pixel intensities without calculating feature points [8].

Lidar-based pose estimation is mainly divided into point-based, feature-based and distribution-based methods. The typical point-based method is ICP [9], which iteratively finds the corresponding points and estimates the pose between frames by minimizing the error function. Feature-based methods are similar to the methods in vision [2]. However, they perform poorly in environments with few feature points or low texture. The NDT methods use a series of Gaussian distributions to represent point cloud clusters which achieve high accuracy of pose estimation [10]. Recently, semantic

information is integrated into scan matching employing a deep neural network to improve registration [11].

Since multi-sensor fusion strategies take advantages of each sensor, more researchers have carried out research in this area. The combination of GPS and INS is a classic integrated positioning technology. Integrate the LIDAR or vision with inertial sensors is another hot research topic [12]. Typical fusion algorithms follow the filtering framework. Extended Kalman Filter (EKF) and Particle Filters (PF) are two most popular filtering technologies in the trajectory fusion [13,14]. However, these methods cannot guarantee the convergence of the probability distribution and the difficulty of obtaining sensor models and uncertainties also limits their applications.

Compared with the filtering-based method, the graph-based SLAM approaches have attracted more attention of researchers. A graph-based network includes nodes and edges. The nodes denote the poses of the vehicle while edges denote the mathematical relationship between nodes [15]. In [16], NDT, GPS, IMU and floor planes are added into a pose graph to obtain the accurate trajectory. Some researchers have also applied deformation into the SLAM field. In [17], a seminal map-centric method namely ElasticFusion is proposed. Although ElasticFusion achieves excellent performance, some features, such as confidence based fusion, limit his application in other sensors models beyond RGB-D.

Furthermore, Park. et al extended the ElasticFusion SLAM to LIDAR sensors [18], however, non-rigid deformation is mainly used in the backend optimization not in the odometer.

### III. METHODOLOGY

#### A. ICP and SOFT Odometry

ICP iteratively estimates the pose matrix between consecutive frames. In each iteration, point-to-point ICP finds the corresponding points with the nearest neighbor search, and the transformation  $(\mathbf{R}, \mathbf{t})$  is calculated by minimizing the following error function:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2 \quad (1)$$

where  $N_m$  and  $N_d$  are the number of the source points  $M$  and target points  $D$ .

SOFT odometry is a novel stereo odometry algorithm relying on feature tracking, which is first proposed in [19] and developed in [3]. This algorithm first extracts corner and blob masks features on the gradient image followed by non-maximum suppression. Features are then matched in circular order, using the sum of absolute differences (SAD) of the patches in the image gradient. Furthermore, these features obtained from last step are input into the RANSAC framework.

Since ICP is the most widely used algorithm for scan registration and SOFT achieves high performance on state-of-the-art vision-based odometry method, we focus on fusing two trajectories from the two methods to improve the pose estimation accuracy.

#### B. Curvefusion

The approach developed in this paper extends the prior work of [4, 5]. Previous work aimed to calculate the similarity metrics of two shape curves. To address this, they introduced a novel shape representation that represents the curve with finitely many rigid transformation matrices instead of a series of point coordinates. Then, the geodesic curve based on the shape representation between two curves is calculated, which is proved to be the intermediate curve of two curves.

Inspired by this, we propose a curve deformation-based trajectory fusion method namely curvefusion. The basic idea is: Given two trajectories, curvefusion considers these two trajectories as two curves representing the same shape. The fusion optimization of two trajectories is transformed into the problem that deforms one curve to the other in a deformation space. Thus, the intermediate curve in the deformation process is our final fusion trajectory.

To apply the shape similarity method to trajectory fusion, some improvements need to be made. In the following sections, we will re-discuss the previous algorithm. And the extension work is also presented in detail.

In [4], a curve  $\mathcal{S}$  is represented by a series of coordinate points on the curve. Here  $\mathcal{S} = (\mathbf{q}_1, \dots, \mathbf{q}_k)$ ,  $\mathbf{q}_i = (x_i, y_i, z_i)^T$ . A trajectory of the robot is not coordinate points but a fixed sequence of poses. Hence,  $\mathcal{S}$  is redefined as  $\mathcal{S} = (\mathbf{p}_1, \dots, \mathbf{p}_k)$  where  $\mathbf{p}_i$  consists of rotation matrix and translation. And the transformation between  $\mathbf{p}_i$  and  $\mathbf{p}_{i+1}$  is given as follows:

$$\hat{\mathbf{g}}_i \mathbf{p}_i = \mathbf{p}_{i+1} \quad (2)$$

Assuming that a starting reference pose  $\mathbf{p}_i$  and a fixed direction are available,  $\mathcal{S} = (\mathbf{p}_1, \dots, \mathbf{p}_k)$  is equivalent as:

$$\mathcal{S} = f^{-1}(\mathcal{G}) = \left( \mathbf{p}_1, \hat{\mathbf{g}}_1 \mathbf{p}_1, \hat{\mathbf{g}}_2 \hat{\mathbf{g}}_1 \mathbf{p}_1, \dots, \left( \prod_{i=1}^k \hat{\mathbf{g}}_i \right) \mathbf{p}_1 \right) \quad (3)$$

where  $\mathcal{G} = (\hat{\mathbf{g}}_1, \dots, \hat{\mathbf{g}}_k) \in SE(n)^k$ . Here  $SE(n)$  denotes the special Euclidean group and  $n$  defines the data dimension. As a result, we can use  $\mathcal{G}$  instead of  $\mathcal{S}$  to represent the trajectory. The specific equation is defined as:

$$f(\mathcal{S}) = \mathcal{G} = (\hat{\mathbf{g}}_1, \dots, \hat{\mathbf{g}}_k) \quad (4)$$

Consequently, a trajectory is represented by finitely many rigid transformation matrices. Here, the  $\hat{\mathbf{g}}_i \in SE(n)$  between two consecutive poses is computed by:

$$\hat{\mathbf{g}}_i = \mathbf{p}_i^{-1} \mathbf{p}_{i+1} \quad (5)$$

Next, we will discuss how to combine two trajectories to get a new trajectory. The main idea is to calculate the geodesic path between two trajectories.

Given two trajectories  $\mathcal{S}^1$  and  $\mathcal{S}^2$ , as mentioned earlier, every trajectory can be represented by finitely many rigid

transformation matrices  $\mathcal{G}$ . Hence,  $\mathcal{S}^1$  and  $\mathcal{S}^2$  are expressed as  $\mathcal{G}^1$  and  $\mathcal{G}^2$ .

$$\begin{aligned}\mathcal{G}^1 &= (\hat{\mathbf{g}}_1^1, \dots, \hat{\mathbf{g}}_k^1), \\ \mathcal{G}^2 &= (\hat{\mathbf{g}}_1^2, \dots, \hat{\mathbf{g}}_k^2)\end{aligned}\quad (6)$$

The homogeneous form of  $\hat{\mathbf{g}}_i^j$ ,  $j = 1, 2$  is defined as:

$$\hat{\mathbf{g}}_i^j = \begin{pmatrix} \mathbf{R}_i^j & \mathbf{v}_i^j \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \text{s.t., } \mathbf{R}_i^j \in SO(n), \mathbf{v}_i^j \in \mathbb{R}^n. \quad (7)$$

The matrix form in the geodesic path between  $\hat{\mathbf{g}}_i^1$  and  $\hat{\mathbf{g}}_i^2$  is calculated as:

$$\begin{aligned}\beta(t)_i &= \mathbf{R}_i^1 ((\mathbf{R}_i^1)^{-1} \mathbf{R}_i^2)^t \\ \alpha(t)_i &= \mathbf{v}_i^1 + (\mathbf{v}_i^2 - \mathbf{v}_i^1)t\end{aligned}\quad (8)$$

where  $t \in [0, 1]$ . By Eq.(8), the rigid transformation matrices representation  $\tilde{\mathbf{g}}(t)_i$  of the  $i$ -th point on geodesic path is obtained as:

$$\tilde{\mathbf{g}}(t)_i = \begin{pmatrix} \beta(t)_i & \alpha(t)_i \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \quad (9)$$

Subsequently, the geodesic path, i.e., the fusion trajectory  $\tilde{\mathcal{S}}(t)$  between  $\mathcal{S}^1$  and  $\mathcal{S}^2$ , is obtained as follows:

$$f(\tilde{\mathcal{S}}(t)) = \tilde{\mathcal{G}} = (\tilde{\mathbf{g}}(t)_1, \dots, \tilde{\mathbf{g}}(t)_k) \quad (10)$$

According to Eq. (4), Eq. (10) represents the intermediate fusion trajectory with finitely many rigid transformation matrixes. Substituting Eq. (10) into Eq.(3), the fusion trajectory  $\tilde{\mathcal{S}}(t) = (\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_k)$  is obtained.

Please note  $0 \leq t \leq 1$ . As shown in Eq. (8), when  $t = 0$ , the final trajectory fusion result degenerates to  $\mathcal{S}^1$ . As  $t$  increases, the fusion trajectory is closer to  $\mathcal{S}^2$ . on the contrary, the fusion curve is closer to  $\mathcal{S}^1$ . When  $t = 1$ , the fusion result becomes  $\mathcal{S}^2$ . In this paper,  $\mathcal{S}^1$  represents the trajectory from point-to-point ICP while  $\mathcal{S}^2$  is from SOFT. For  $\mathcal{S}^1$ , only one point-to-point ICP is utilized, hence, the trajectory  $\mathcal{S}^2$  is superior to  $\mathcal{S}^1$ .

To select the optimal fusion trajectory, we set the denominator of  $t$  to 50 and the step size to 1/50. Specifically, when solving Eq. (10),  $t$  is updated cyclically according to the following equation:

$$t \rightarrow t + \frac{1}{50} \quad (11)$$

The denominator of  $t$  determines the number of fusion trajectories. In this paper, we can get 51 trajectories. Therefore, selecting the optimal fusion trajectory is also a problem that has to be solved. Our selection criterion includes two aspects:

1) If the trajectory to be fused includes a loop, we calculate the position error of the loop, and the fusion

trajectory with minimized error is considered as the optimal fusion trajectory.

2) If the trajectory to be fused does not include loops, the fusion trajectory  $\tilde{\mathcal{S}}(t = \frac{49}{50})$  is selected as the optimal trajectory. This is based on the premise that  $\mathcal{S}^2$  has better performance than  $\mathcal{S}^1$ , and  $\tilde{\mathcal{S}}(t = \frac{49}{50})$  is the closest to  $\mathcal{S}^2$ .

#### IV. RESULTS AND DISCUSSION

We have evaluated our fusion method using datasets from the KITTI odometry benchmark [20]. The datasets are collected with an automobile equipped with a Velodyne laser scanner, stereo cameras and a high accuracy GPS/INS for ground truth. It contains 11 sequences which cover urban, rural and highway scenarios. In our Experiment, data from stereo cameras and the Velodyne LIDAR are used.

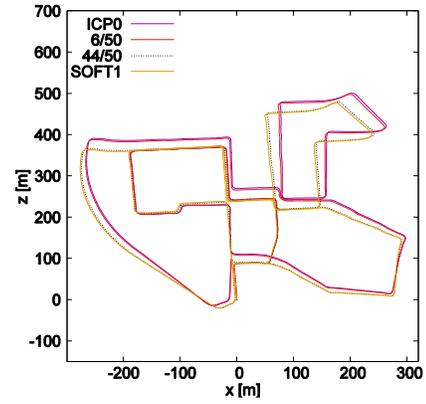


Figure 1. Fused trajectories. ICP0 corresponds to  $t = 0$ , SOFT1 corresponds to  $t = 1$ . The intermediate fusion trajectories  $t = \frac{6}{50}$  and  $t = \frac{44}{50}$  are also presented. The dataset is from sequence 00.

Fig.1 shows the intermediate process of  $\mathcal{S}^1$  (ICP0) deforms to  $\mathcal{S}^2$  (SOFT1). Note that only two intermediate fusion trajectories are presented. As shown in Fig. 1, when  $t = \frac{44}{50}$ , the position error of the loop of the trajectory is the smallest, therefore, we elect  $t = \frac{44}{50}$  as the optimal fusion trajectory in sequence 00.

Our curvefusion is evaluated using the absolute metric [21] and KITTI metric [20], respectively. The absolute metric calculates the absolute root-mean-square error of the translation and absolute average rotation errors over the all scans according to Eq. (12) to (14).

$$\Delta \mathbf{T}_{abs,i} = \begin{pmatrix} \Delta \mathbf{R}_{abs,i} & \Delta \mathbf{t}_{abs,i} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} = \mathbf{T}_{r,i} \mathbf{T}_{e,i}^{-1}, \quad (12)$$

where  $\mathbf{T}_{r,i}$  and  $\mathbf{T}_{e,i}$  denote the pose matrix of the ground truth and estimated fusion trajectory, respectively in the  $i$ -th frame. Subsequently, RMSE of the translation and the absolute average rotation errors are computed as follows:

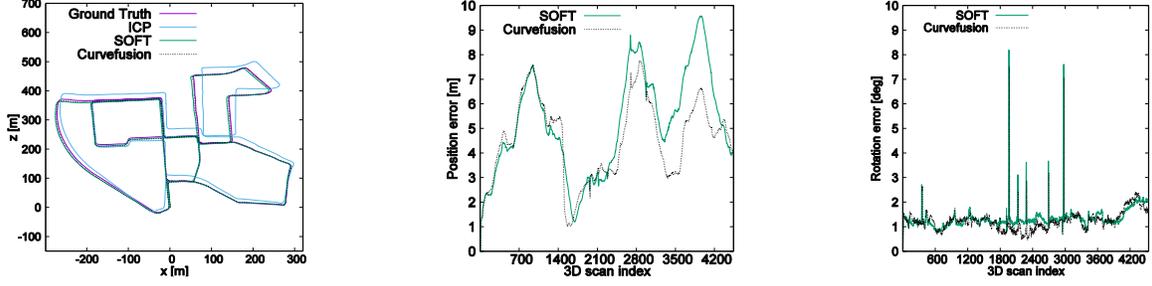


Figure 2. Trajectory and translation as well as rotational error comparison of seq00.

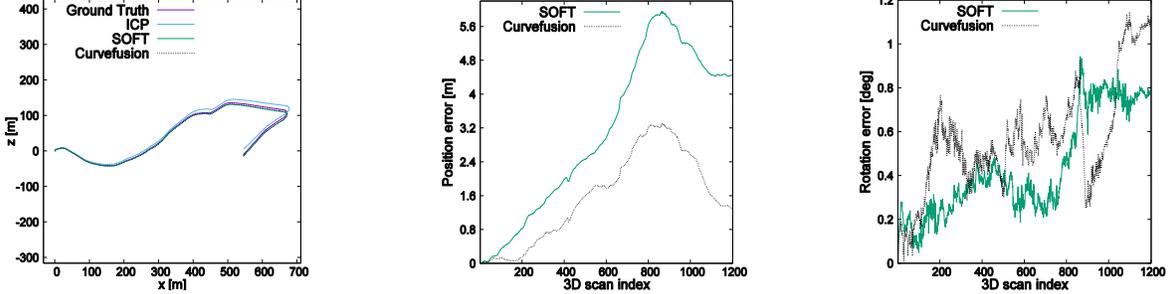


Figure 3. Trajectory and translation as well as rotational error comparison of seq10.

$$\sigma = \sqrt{\frac{1}{n+1} \sum_{i=0}^n \|\Delta \mathbf{t}_{abs,i}\|^2} \quad (13)$$

$$r_m = \sum_{i=0}^n f_{\theta}(\Delta \mathbf{R}_{abs,i}) \quad (14)$$

where  $\|\cdot\|$  represents the Euclidean metric.  $f_{\theta}(\cdot)$  is defined in [21].

Due to space limitations, we only show the error graphs of sequence 00 and sequence 10. As Fig. 2 and Fig. 3 show, by fusing SOFT and ICP using curvefusion, the accuracy of fusion trajectory is improved. Note that since the errors of ICP are larger than SOFT, ICP is not evaluated in this paper. Fig. 2 and Fig. 3 also demonstrate that when one trajectory has high accuracy, while the other trajectory suffers a large error, curvefusion can still obtain excellent fusion trajectory.

TABLE I. RESULTS OF OUR CURVEFUSION METHOD COMPARED WITH SOFT ON THE KITTI DATASET USING ABSOLUTE METRIC TABLE TYPE STYLES

Sequences	Curvefusion			SOFT	
	$\lambda$	$t_{err}$ [m]	$r_{err}$ [deg]	$t_{err}$ [m]	$r_{err}$ [deg]
seq00	44	<b>4.8396</b>	<b>1.2808</b>	5.5927	1.3556
seq01	49	<b>5.0246</b>	<b>0.4347</b>	5.9222	0.5235
seq02	n.a.	n.a.	n.a.	n.a.	n.a.
seq03	49	<b>1.4035</b>	<b>0.7175</b>	1.5077	0.7312
seq04	49	<b>0.3365</b>	0.0972	0.3546	<b>0.0909</b>
seq05	49	<b>1.7893</b>	0.4873	1.8155	<b>0.4611</b>
seq06	47	<b>1.6604</b>	<b>0.7298</b>	1.8186	0.8209
seq07	49	1.2342	0.5485	<b>1.2135</b>	<b>0.5069</b>
seq08	49	<b>18.6710</b>	<b>2.9776</b>	18.6779	3.0186
seq09	49	5.7377	<b>0.5047</b>	<b>5.7188</b>	0.5175
Seq10	49	<b>1.9063</b>	0.5694	3.6449	0.4403

The absolute translation and rotation error to ground truth based on equation (13) and (14) are given in Table I. Note that ICP fails in sequence 02, hence, Curvefusion is not applied in sequence 02.  $\lambda$  represents the numerator of  $t$ .  $t_{err}$  denotes translation error, while  $r_{err}$  is rotation error.

It can be seen from Table I that parameter  $t$  of the fusion trajectory of most sequences is selected as 49. This is based on the fact that the accuracy of the SOFT trajectory outperforms the simple point-to-point ICP, thus, the optimal fusion trajectory is naturally very close to SOFT. Table I also demonstrates that our fusion algorithm achieved better and similar results before fusion compared with SOFT.

TABLE II. RESULTS ON KITTI ODOMETRY

Sequences	Curvefusion	SOFT	LOAM	IMLS-SLAM
seq00	0.61/0.37	<b>0.50/0.28</b>	0.78/-	<b>0.50/-</b>
seq01	0.46/ <b>0.17</b>	<b>0.40/0.18</b>	1.43/-	0.82/-
seq02	n.a.	n.a.	0.92/-	<b>0.53/-</b>
seq03	<b>0.53/0.45</b>	0.57/ <b>0.44</b>	0.86/-	0.68/-
seq04	<b>0.30/0.26</b>	0.30/0.24	0.71/-	0.33/-
seq05	<b>0.29/0.19</b>	0.33/0.18	0.57/-	0.32/-
seq06	0.20/0.17	<b>0.17/0.15</b>	0.65/-	0.33/-
seq07	0.38/0.33	0.38/0.32	0.63/-	<b>0.33/-</b>
seq08	0.88/ <b>0.44</b>	0.90/0.45	1.12/-	<b>0.80/-</b>
seq09	<b>0.54/0.23</b>	0.54/0.23	0.77/-	0.55/-
Seq10	0.66/0.39	0.61/ <b>0.26</b>	0.79/-	<b>0.53/-</b>

relative rotational error in degrees per 100m / relative translational error in %.

Table II shows the relative translational and rotational errors on KITTI metrics. LOAM and IMLS-SLAM [22] are two of the state-of-the-art Lidar-based odometry or SLAM algorithm. We can see that curvefusion achieves better or similar results compared to the popular pose estimation

approaches. However, by comparing Table I and Table II, our method is better in terms of absolute error. This reason can be the ICP suffers from a large relative error, which is retained in the fusion trajectory.

## V. CONCLUSION

We present a novel approach called curvefusion for integrating Lidar scan matching with stereo visual odometry. Different from filtering-based or pose graphs optimization-based fusion methods, we propose a deformation-based trajectory method.

We evaluated the absolute and relative errors on the KITTI benchmark, and the results show that curvefusion outperforms or achieves similar achievement compared with single the state-of-the-art odometry methods.

Nevertheless, some existing problems need to be solved. First, simple point-to-point ICP does not perform well in some harsh environments, e.g., seq02, which ultimately leads to a limited improvement in the accuracy of our fusion trajectory. A solution is to replace the ICP with a better-performing LIDAR odometry, such as LOAM. Besides, for the optimal curve selection criteria, we simply judge by loop position error which not applies to all situations. Probability statistics is expected to be used to solve our problems.

## REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (slam): Part i the essential algorithms," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [2] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 2174–2181.
- [3] I. Cvitic, J. Cestic, I. Markovic and I. Petrovic, "Soft-slam: Computationally efficient stereo visual slam for autonomous uavs," *Journal of Field Robotics (JFR)*, 2017.
- [4] G. G. Demisse, D. Aouada, and B. Ottersten, "Similarity metric for curved shapes in euclidean space," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5042–5050.
- [5] G. G. Demisse, D. Aouada, and B. Ottersten "Deformation based curved shape representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1338–1351, 2017.
- [6] H. Bay, T. Tuytelaars and L. Van Gool SURF: Speeded up robust features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; 2006:3951
- [7] E. Rublee, V. Rabaud, K. Konolige. ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*; 2011:2564–2571.
- [8] M. Pizzoli, C. Forster, D. Scaramuzza REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE International Conference on Robotics and Automation*; 2014:2609–2616.
- [9] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proceedings of Robotics: Science and Systems*, 2009.
- [10] M. Magnusson, N. Vaskevicius et al., "Beyond points: Evaluating recent 3d scan-matching algorithms," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp.3631–3637.
- [11] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak, "Integrating deep semantic segmentation into 3d point cloud registration," *IEEE Robotics and automation letters*, vol. 3, no. 4, pp. 2942–2949, 2018.
- [12] Z. Zhang, S. Liu, G. Tsai et al., "Pirvs: An advanced visual-inertial slam system with flexible sensor fusion and hardware co-design," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp.3826–3832.
- [13] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 298–304.
- [14] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–36, 2007.
- [15] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [16] K. Koide, J. Miura, E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement". *International Journal of Advanced Robotic Systems*. March 2019.
- [17] T. Whelan, R. F. Salas-Moreno, B. Glocker et al., "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.
- [18] C. Park, P. Moghadam, S. Kim, A. Elfes et al., "Elastic lidar fusion: Dense map-centric continuous-time slam," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp.1206–1213.
- [19] I. Cvitic and I. Petrovic, "Stereo odometry based on careful feature selection and tracking," in *In European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.
- [20] A. Geiger, P. Lenz, and R. Urtasum, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3354–3361.
- [21] S. May, D. Droschel et al., "Three-dimensional mapping with time-of-flight cameras," *Journal of Field Robotics*, vol. 26, no. 11, pp. 934–965, 2009.
- [22] J. Deschaud. Imls-slam: scan-to-model matching based on 3d data. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.