Planar Odometry from a Radial Laser Scanner. A Range Flow-based Approach

Mariano Jaimez, Javier G. Monroy and Javier Gonzalez-Jimenez

Abstract— **In this paper we present a fast and precise method to estimate the planar motion of a lidar from consecutive range scans. For every scanned point we formulate the range flow constraint equation in terms of the sensor velocity, and minimize a robust function of the resulting geometric constraints to obtain the motion estimate. Conversely to traditional approaches, this method does not search for correspondences but performs dense scan alignment based on the scan gradients, in the fashion of dense 3D visual odometry. The minimization problem is solved in a coarse-to-fine scheme to cope with large displacements, and a smooth filter based on the covariance of the estimate is employed to handle uncertainty in unconstraint scenarios (e.g. corridors). Simulated and real experiments have been performed to compare our approach with two prominent scan matchers and with wheel odometry. Quantitative and qualitative results demonstrate the superior performance of our approach which, along with its very low computational cost (0.9 milliseconds on a single CPU core), makes it suitable for those robotic applications that require planar odometry. For this purpose, we also provide the code so that the robotics community can benefit from it.**

I. Introduction

Odometry is an essential component for robot localization. It is commonly solved through three major techniques that are based on inertial devices, wheel encoders or visual odometry (either by feature tracking or by dense image alignment). Inertial measurement units (IMUs) are ideal to estimate spatial orientation but accumulate too much translational error over time due to their inability to cancel the gravitational component of the measurement [1]. Odometry based on enconders has extensively been used to provide fast motion estimates for wheeled or legged robots, though this approach is prone to being inaccurate due to wheel/leg slippage and the impreciseness of the kinematic robot models. Last, vision-based methods are arguably the most flexible and powerful solution to the motion estimation problem because they can be adapted to work with different types of robots (wheeled, legged, aerial) and configurations (2D-3D motion).

Our proposal here relies on laser scans and has the advantage over the aforementioned methods to be independent of the vehicle type of locomotion as well as very fast and precise, as supported by experimental validation. Thus, it turns out to be particularly suitable for those (very common) cases where the robot already uses a laser range scanner for mapping, obstacle avoidance or localization. Our approach, named RF2O (Range Flow-based 2D Odometry), builds upon [2] and represents the apparent motion of any point observed by the sensor as a function of the velocity of the sensor, assuming that the environment is static. Thus, every point defines a geometric residual which can be minimized within a dense formulation to obtain the lidar motion. To overcome the assumption of a motionless environment (i.e. to handle moving objects), we compute the Cauchy M-estimator of the geometric residuals, a more robust estimate than traditional choices like the L2 or L1 norms. Furthermore, we solve this estimation problem within a coarse-to-fine scheme, which provides finer results and allows the method to cope with larger motions.

We have conducted a varied set of experiments to compare our method against point-to-line iterative closest point (PL-ICP) [3] and the polar scan matching approach (PSM) [4]. Firstly, their performances are evaluated at different scan rates on simulated scenarios where the ground truth is available. Secondly, qualitative results are shown for a real experiment where 2D maps are built by concatenating the scanned points according to the odometry motion estimates of each method. Thirdly, we devise a real experiment to evaluate how robust the methods are against the presence of noise and moving objects. Overall, results show that RF2O is significantly more precise for both translations and rotations, and presents the lowest runtime (2 times faster than PSM and 20 times faster than PL-ICP). Besides analyzing the results presented herein, we encourage the reader to watch the demonstration video which, together with the available code, can be found at:

`http://mapir.isa.uma.es/work/rf2o`

II. Related Work

Although low-cost RGB-D cameras have recently favored the transition to 3D odometry, localization and mapping strategies, it is a matter of fact that a fair number of mobile robots move on planar surfaces and rely on laser scanners to navigate. In this context, very successful results have been achieved in the fields of 2D Localization [5][6] and SLAM [7], and many algorithms have been proposed to solve the general scan matching problem [8][3][9]. In this paper we focus on pure 2D odometry, which can be regarded as a particular case of scan matching, where the scans to be aligned are taken consecutively and are normally close to each other.

Traditionally, ICP [9] or some of its variants have been applied to solve the registration problem between consecutive scans. A very successful approach was proposed by Censi

This work has been funded by the Spanish Government under project DPI2014-55826-R and the grant program FPI-MICINN 2012.

All authors belong to the MAPIR group of the department of System Engineering and Automation, University of Malaga. {marianojt, jgmonroy, javiergonzalez}@uma.es

[3], where a point-to-line metric is used instead of the point-to-point original metric of ICP. Futhermore, the author presented an implementation which ran one order of magnitude faster than existing ICP variants, and was more precise and efficient than the pioneer point-to-segment work in [10]. More recently, Generalized-ICP [11] showed an improved performance over previous ICP versions, but has been mostly used for the registration of 3D point clouds. For this family of methods, accuracy depends on every particular version and implementation, yet they all share the same weakness: they are computationally expensive.

Alternatively, other methods were specifically designed to solve the 2D scan matching problem. Gonzalez&Gutierrez [2] formulated the "velocity constraint equation", an adaptation of the optical flow constraint for range scans, and proposed to estimate the lidar motion by imposing this restriction for every point observed in the scans. However, their method was only tested with simple simulated scenarios and provided modest results. Diosi&Kleeman presented the Polar Scan Matching approach [8], where the translation and rotation between two scans are alternately estimated until convergence. Conversely to ICP, this method avoids searching for correspondences by simply matching points with the same bearing, which leads to a higher computational performance. This approach was subsequently extended and further evaluated in [4]. A different method proposed by Olson [12] tries to find the rigid transformation that maximizes the probability of having observed the latest scan given the previous one. Additional information is used (control inputs or wheel odometry) to ease the method convergence and two different implementations, GPU and multi-resolution CPU, are presented. A thorough evaluation is performed in terms of computational performance but, surprisingly, no results for the method's accuracy are presented.

More recently, other approaches have built upon the aforementioned works. It is the case of [13] and [14], which fuse laser odometry (the Olson's laser odometry [12] and point-to-line ICP [3], respectively) with stereo vision to perform autonomous navigation with UAVs. Furthermore, the work of Pomerleau *et al.* [15] presents a fast implementation and a through evaluation of some ICP variants on real-world 2D and 3D data sets.

III. Lidar Velocity and 2D Range Flow

In this section we describe how the 2D velocity of a laser scanner can be estimated from the apparent motion that it observes, assuming that the environment is static and rigid. Let $R(t, \alpha)$ be a range scan where t is the time and $\alpha \in [0, N) \subset \mathbb{R}$ is the scan coordinate, being N the size of the scan. The position of any point P with respect to the local reference frame attached to the sensor is given by its polar coordinates (r, θ) (see Fig. 1). Provided that P is visible from the lidar, it will be observed at a scan coordinate α that is directly related to the angular coordinate of P:

$$\alpha = \frac{N-1}{FOV}\,\theta + \frac{N-1}{2} = k_\alpha \theta + \frac{N-1}{2} \tag{1}$$

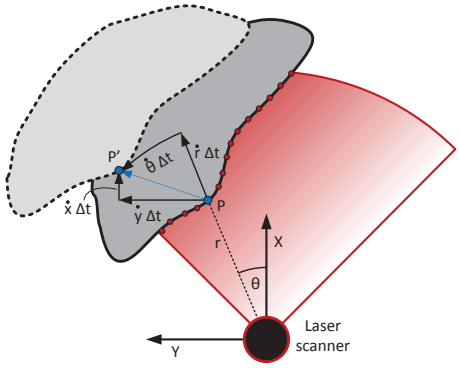

Fig. 1. Top view representation of a laser scan that intersects with a certain object. An observed point P moves with respect to the scanner to P' after an interval of time Δt.

where FOV is the scanner field of view. Similarly to the optical flow constraint equation, a linear constraint can be derived from the general expression of geometric consistency of two scan pairs. Assuming the differentiability of R, the range of any point at the second scan can be expressed as the Taylor expansion

$$R(t+\Delta t, \alpha+\Delta\alpha) = R(t,\alpha) + \frac{\partial R}{\partial t}(t,\alpha)\,\Delta t + \frac{\partial R}{\partial \alpha}(t,\alpha)\,\Delta\alpha + O(\Delta t^2, \Delta\alpha^2) \tag{2}$$

where Δt is the time lapse between consecutive scans and $\Delta\alpha$ represents the change in the scan coordinate of the point considered. Neglecting the second/higher order terms, and dividing by Δt, we can obtain a simple expression that relates the scan gradients with the change of the range and the scan coordinate of a point during the interval $[t, t+\Delta t]$:

$$\frac{\Delta R}{\Delta t} \simeq R_t + R_\alpha \frac{\Delta\alpha}{\Delta t} \tag{3}$$

with

$$\Delta R = R(t+\Delta t, \alpha+\Delta\alpha) - R(t,\alpha),$$
$$R_t = \frac{\partial R}{\partial t}(t,\alpha), \quad R_\alpha = \frac{\partial R}{\partial \alpha}(t,\alpha).$$

If we consider that $\dot r = \Delta R/\Delta t$ and $\dot\alpha = \Delta\alpha/\Delta t$ are the average velocities of a point in range and scan coordinates during the interval $[t, t+\Delta t]$, we obtain:

$$\dot r \simeq R_t + R_\alpha \dot\alpha = R_t + R_\alpha k_\alpha\, \dot\theta \tag{4}$$

Equation (4) was firstly introduced by Gonzalez&Gutierrez [2] and subsequently generalized and named as the "range flow constraint equation" in [16].

In order to describe the velocities of all points with respect to the same vector basis, we transform the radial and azimuthal velocities $(\dot r, \dot\theta)$ to a cartesian representation $(\dot x, \dot y)$, as shown in Fig. 1:

$$\dot r = \dot x \cos\theta + \dot y \sin\theta \tag{5}$$

$$r\,\dot\theta = \dot y \cos\theta - \dot x \sin\theta \tag{6}$$

As a last step, we need to impose that every apparent motion is actually caused by the lidar translation and/or rotation. In

other words, we assume that every point moves with respect to the sensor as if it was part of a rigid body whose velocity is the same but opposite in sign to that of the sensor:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{pmatrix}-v_{x,s} + y\,\omega_s\\ -v_{y,s} - x\,\omega_s\end{pmatrix} \tag{7}$$

being $\boldsymbol{\xi_s} = (v_{x,s}\,, v_{y,s}\,, \omega_s)$ a 2D twist (sensor velocity) and (x, y) the cartesian coordinates of P. If the cartesian velocities (5) (6) are substituted in (4) and the rigidity hypothesis (7) is imposed, we can transform the range flow constraint equation into a constraint for the lidar velocity:

$$\left(\cos\theta + \frac{R_\alpha k_\alpha \sin\theta}{r}\right) v_{x,s} + \left(\sin\theta - \frac{R_\alpha k_\alpha \cos\theta}{r}\right) v_{y,s} + \left(x\sin\theta - y\cos\theta - R_\alpha k_\alpha\right)\omega_s + R_t = 0 \tag{8}$$

As a result, every scanned point imposes a restriction to the sensor motion and, therefore, 3 linearly independent restrictions would theoretically suffice to estimate it.

IV. Velocity Estimation

In practice, the lidar motion cannot be estimated with only three independent restrictions because, in general, (8) is inexact due to the noise of the range measurements, the errors made by the linear approximation (3) or the presence of moving object (non-static environment). Therefore, we propose a dense formulation in which all the points of the scan contribute to the motion estimate. For every point, we define the geometric residual $\rho(\boldsymbol{\xi})$ as the evaluation of the range flow constraint (8) for a given twist $\boldsymbol{\xi}$:

$$\rho(\boldsymbol{\xi}) = R_t + \left(x\sin\theta - y\cos\theta - R_\alpha k_\alpha\right)\omega + \left(\cos\theta + \frac{R_\alpha k_\alpha \sin\theta}{r}\right) v_x + \left(\sin\theta - \frac{R_\alpha k_\alpha \cos\theta}{r}\right) v_y \tag{9}$$

To obtain an accurate estimate, the sensor motion is computed by minimizing all the geometric residuals within a robust function F:

$$\boldsymbol{\xi}_M = \arg\min_{\boldsymbol{\xi}} \sum_{i=1}^{N} F(\rho_i(\boldsymbol{\xi})) \tag{10}$$

$$F(\rho) = \frac{k^2}{2}\, ln\left(1 + \left(\frac{\rho}{k}\right)^2\right) \tag{11}$$

The function F is the Cauchy M-estimator, and k is an adjustable parameter. Conversely to the more common choices of the L2 or L1 norms, this function reduces the relevance of those points with very high residuals, and represents an effective and automatic way to deal with outliers. The optimization problem is solved with Iteratively Reweighted Least Squares (IRLS), where the weights associated to the Cauchy M-estimator are:

$$w(\rho) = \frac{1}{1 + \left(\frac{\rho}{k}\right)^2} \tag{12}$$

With IRLS, the system is iteratively solved by recomputing the residuals and subsequently the weights until convergence.

A. Pre-weighting strategy

As previously mentioned, there are some factors that can render (8) inaccurate, mainly the unfulfillment of the rigidity hypothesis (7) and the deviations from the linear approximation made in (3). Although the Cauchy M-estimator can alleviate their effect on the overall motion estimate, it does not eliminate it completely. The presence of moving objects is hard to detect before solving the motion and, therefore, we rely on the Cauchy M-estimator to downweight them during the minimization process. On the other hand, deviations from the linear approximation adopted in (3) can be detected beforehand, which helps to accelerate convergence in (10) and also leads to more accurate results. To this purpose, we propose a pre-weighting strategy to downweight the residuals of those points where the range function is nonlinear or even non-differentiable. We call it "pre-weighting" because it is applied before the minimization problem (10) is solved. In order to quantify the error associated to the linearization of (2), we expand the Taylor series to the second order:

$$\begin{aligned} \dot{r} &= R_t + R_\alpha\dot{\alpha} + R_{2o}(\Delta t, \dot{\alpha}) + O(\Delta t^2, \dot{\alpha}) \\ R_{2o}(\Delta t, \dot{\alpha}) &= \frac{\Delta t}{2}\left(R_{tt} + R_{t\alpha}\dot{\alpha} + R_{\alpha\alpha}\dot{\alpha}^2\right) \end{aligned} \tag{13}$$

It can be noticed that, neglecting higher order terms, the second order derivatives in $R_{2o}(\Delta t, \dot{\alpha})$ can be used to detect the deviations from linearity. One special case is the second order derivative with respect to time (R_{tt}), which cannot be computed in a coarse-to-fine scheme because the warped images are timeless and, therefore, the concept of second temporal derivative makes no sense (coarse-to-fine is described in Section V). Moreover, it is important to detect those regions of the scans where the range function is not only nonlinear but non-differentiable. These regions are mainly the edges of the different objects observed, and are typically characterized by very high values of the first order derivatives (R_t and/or R_α). To penalize these two effects, nonlinearities and discontinuities, we define the following pre-weighting function for each scanned point:

$$\bar{w} = \frac{1}{\sqrt{\epsilon + {R_\alpha}^2 + \Delta t^2 {R_t}^2 + K_d\left({R_{\alpha\alpha}}^2 + \Delta t^2 {R_{t\alpha}}^2\right)}} \tag{14}$$

The parameter K_d marks the relative importance of first order and second order derivatives, and ϵ is a small constant to avoid the singular case.

Thus, we initially compute a pre-weighted set of residuals

$$\rho_i^w(\boldsymbol{\xi}) = \bar{w}_i\ \rho_i(\boldsymbol{\xi}) \qquad\qquad i \in \{1, 2...N\} \tag{15}$$

which are subsequently minimized according to (10) (11). Although we do not show comparisons in the paper, this strategy provides better results than standard IRLS minimization without pre-weighting and converges faster (approximately by a factor of 2).

V. Coarse-to-Fine Scheme and Scan Warping

The linearization presented in (3) holds either for small displacements between consecutive scans or at areas with

constant range gradients (which, in the case of a lidar, would occur for a very unusual geometry: an Archimedean spiral). To overcome this limitation, we estimate motion in a coarse-to-fine scheme, where the coarser levels provide a rough estimate which is improved subsequently in finer levels. The coarse-to-fine scheme was introduced by Battiti *et al.* [17] to solve the optical flow problem for large displacements, and has commonly been adopted ever since [18][19].

Let R_0, R_1 be two consecutive laser scans. Initially, two Gaussian pyramids are to be created by successively downsampling (typically by 2) the original scans R_0, R_1. Normally, a Gaussian mask is applied to downsample RGB or grayscale images but, in the case of range data, a standard Gaussian filter is not the best choice since it creates artifacts on the filtered scans. As an alternative, we employ a bilateral filter [20] that does not mix distant points which are likely to belong to different objects of the scene. Once the pyramids are built, the velocity estimation problem is iteratively solved from the coarsest to the finest level. At every transition to a finer level, one of the two input scans must be warped against the other according to the overall velocity estimated in previous levels ($\boldsymbol{\xi}_p$). This warping process is always divided into two steps and, in our formulation, is applied over the second scan R_1. Firstly, every point P observed in R_1 is spatially transformed using the rigid body motion associated to the twist $\boldsymbol{\xi}_p$:

$$\begin{pmatrix} x^w \\ y^w \\ 1 \end{pmatrix} = e^{\hat{\boldsymbol{\xi}}_p} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \; \hat{\boldsymbol{\xi}}_p = \Delta t \begin{pmatrix} 0 & -\omega_p & v_{x,p} \\ \omega_p & 0 & v_{y,p} \\ 0 & 0 & 0 \end{pmatrix} \quad (16)$$

Secondly, the transformed points must be reprojected onto R_1 to build the warped scan R_1^w so that:

$$R_1^w(\alpha^w) = \sqrt{(x^w)^2 + (y^w)^2}, \quad (17)$$

$$\alpha^w = k_\alpha \arctan\left(\frac{y^w}{x^w}\right) + \frac{N-1}{2} \quad (18)$$

Several points could be warped to the same coordinate α^w, in which case the closest one is preserved (the others would be occluded). If $\boldsymbol{\xi}_p$ is converging to the real velocity, then the warped scan R_1^w will be considerably closer to the first scan R_0 than the original R_1, which allows us to apply the linear approximation in (2) with a finer resolution.

VI. Implementation

Our algorithm pays special attention to the computation of the range gradients. Normally, a fixed discrete formula is employed to approximate either scan or image gradients. In the case of range data, this strategy leads to very high values of the gradients at the object borders, which do not represent the real gradients over those objects. As an alternative, we make use of an adaptive formula that regards the geometry of the environment. This formula weights forward and backward derivatives in the scan with the 2D distances between contiguous observations (points):

$$\begin{aligned} R_\alpha(\alpha) &= \frac{d(\alpha+1)\, R_\alpha^-(\alpha) + d(\alpha)\, R_\alpha^+(\alpha)}{d(\alpha+1) + d(\alpha)} \quad (19) \\ R_\alpha^- &= R(\alpha) - R(\alpha-1),\; R_\alpha^+ = R(\alpha+1) - R(\alpha) \\ d(\alpha) &= \|((x(\alpha) - x(\alpha-1), y(\alpha) - y(\alpha-1))\| \end{aligned}$$

Thus, the closest neighbour is always contributing more to the gradient computation while distant points barely affect it. In the case that both neighbours are approximately equidistant, the presented formula is equivalent to a centered finite difference approximation. More details about the gradient computation can be found in [19].

Last, it is important to remark that there are some geometric configurations of the environment from which the sensor motion cannot be recovered. The most common case arises when the lidar only observes a wall. Under this circumstance, the motion parallel to the wall is undetermined and therefore the solver would provide an arbitrary solution for it (not only our method but any approach based purely on geometry). In order to mitigate this problem, we apply a low-pass filter in the eigenspace of the velocity $\boldsymbol{\xi}$ which works as explained next. First, the eigenvalues of the covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ of the IRLS solution are analyzed to detect which motion (or combinations of motions) are undetermined and which are perfectly constrained. In the space of the eigenvectors, the velocity $\boldsymbol{\xi}_M^t$ provided by (10) is weighted with that of the previous time interval $\boldsymbol{\xi}^{t-1}$ to obtain the new filtered velocity $\boldsymbol{\xi}^t$:

$$[(1+k_l)I + k_e E]\, \boldsymbol{\xi}^t = \boldsymbol{\xi}_M^t + (k_l I + k_e E)\boldsymbol{\xi}^{t-1} \quad (20)$$

where E is a diagonal matrix containing the eigenvalues and k_l, k_e are parameters of the filter. Concretely, k_l imposes a constant weighting between the solution from the solver and the previous estimate while k_e defines how the eigenvalues affect the final estimate. These parameters are set to the following values:

$$k_l = 0.05 e^{-(l-1)}, \quad k_e = 15 \times 10^3 e^{-(l-1)} \quad (21)$$

where l is the pyramid level that ranges from 1 (coarsest) to the number of levels considered. Please refer to [19] for a more detailed explanation on this filter and how it is applied.

VII. Experiments

This section is composed of a set of three different experiments. The two first experiments address the evaluation of the proposed RF2O algorithm and its comparison with other approaches in simulated and real environments, respectively. The third experiment is carried out to analyze the robustness of the motion estimates against noise and the presence of moving objects. For comparison, two state-of-the-art scan matchers are selected: Point-to-Line ICP (PL-ICP) [3], and Polar Scan Matching (PSM) [4]. In both cases, we use the implementations that their authors published online. For quantitative evaluation, the relative pose errors as described in [21] will be considered. Both translational and rotational deviations per second will be evaluated with the root mean

TABLE I

SIMULATED EXPERIMENT - TRANSLATIONAL AND ROTATIONAL DEVIATIONS PER SECOND, AND EXECUTION TIMES.

		Translational RMSE (cm/s)			Rotational RMSE (deg/s)			Runtime (ms)		
	Scan rate (Hz)	RF2O	PSM	PL-ICP	RF2O	PSM	PL-ICP	RF2O	PSM	PL-ICP
Scen. 1	10	**0.425**	14.82	1.860	**0.108**	2.412	0.524	**0.941**	1.837	15.98
	5	**0.308**	7.363	0.759	**0.054**	1.572	0.321	**0.933**	1.979	18.51
	2	**0.248**	3.071	0.584	**0.043**	0.598	0.281	**0.904**	2.205	23.79
	1	**0.273**	12.27	0.396	0.372	2.290	**0.108**	**0.900**	2.675	27.58
Scen. 2	10	**0.398**	19.56	1.904	**0.121**	4.725	0.473	**0.951**	1.994	19.02
	5	**0.346**	18.60	1.084	**0.084**	4.370	0.268	**0.935**	2.642	23.84
	2	**0.785**	18.13	10.14	**0.339**	4.155	3.042	**0.931**	3.351	28.59
	1	**5.250**	42.67	24.07	**3.669**	15.67	7.282	**0.892**	3.656	35.56
Scen. 3	10	**0.461**	4.940	18.44	**0.071**	1.469	0.246	**0.922**	1.826	19.55
	5	**0.382**	5.499	39.74	**0.054**	2.027	0.129	**0.940**	2.296	15.25
	2	**0.249**	7.138	38.48	**0.033**	2.328	0.071	**0.900**	2.911	17.54
	1	**0.439**	33.51	40.19	0.106	3.693	**0.068**	**0.875**	3.677	26.74

squared error (RMSE), which corresponds to a performance measure independent of the experiment duration.

For real experiments, a Hokuyo URG-04LX-UG01 laser scanner mounted on a Giraff mobile robot [22] is used to gather the laser scans at a maximum frequency of 10 Hz. For the case of simulated experiments, the laser characteristics have been imitated (ray number = 682, FOV = 240 °, max distance = 5.5 m). Moreover, a Gaussian noise with σ = 1 cm is added to the simulated scans to make them more realistic.

A. *Comparison in a Simulated Environment*

In this experiment, the compared methods estimate the planar motion of a laser scanner that moves in a simulated environment. We use the precise ground truth available in simulation to perform a quantitative evaluation of the different approaches. The simulated environment is divided into three distinct scenarios: a room containing only objects formed by straight line segments (Scen. 1), a room that contains only curved obstacles and curved walls (Scen. 2) and a straight corridor with scattered small objects (Scen. 3). During the experiment, the lidar travels along a predefined path, covering a distance of 43.47 meters at an average speed of 0.398 m/s. Furthermore, four different scan rates (10, 5, 2 and 1 Hz) are tested to analyze the influence of the frequency of execution in the odometry estimates. Table I depicts the relative pose errors in the form of translational and rotational deviations per second, together with their runtimes for the three compared methods. Fig. 2 plots the simulated environment with the best estimated trajectory of each method. That is, from all the execution rates, only the one with overall minimum error is plotted for qualitative assessment. As can be noticed, RF2O exceeds the other two approaches for all the scenarios in the experiment, providing much more accurate estimates. PL-ICP presents relatively good estimates for the room scenarios, but it drastically fails at the corridor (specially for translations). On the other hand, PSM presents much higher relative errors in general, and concretely at the second scenario where only curved objects can be found. Furthermore, it presents important problems at narrow places such as doors.

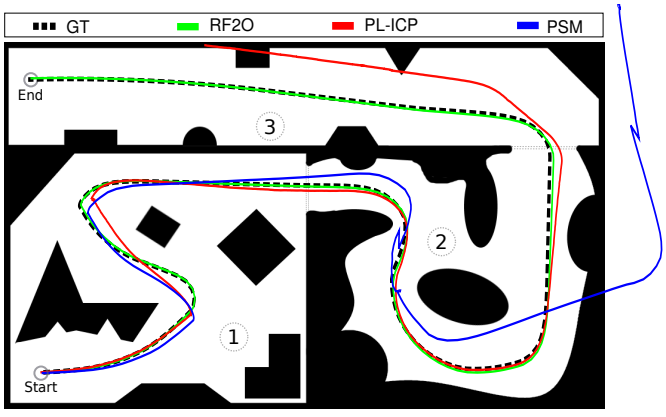

Fig. 2. Simulated environment and the best path as estimated by each method (RF2O@5Hz, PL-ICP@10Hz, PSM@2Hz). Numbers indicates the different scenarios of the environment.

It is interesting to notice that the best results are not obtained with the highest frequency. Experiments at 10 Hz provide worse results than those at 5 or 2 Hz, which indicates that data oversampling leads to error accumulation. On the other hand, a too little frequency implies that consecutive scans are too separate and more difficult to align (as occurs at 1 Hz). Thus, the optimal frequency is not always the highest available one and depends on the average (or maximum) linear and angular speeds of the lidar.

An alternative and helpful way to compare these methods is to calculate their RMS errors per segment length, as described in [23]. Fig. 3 depicts these average translational errors as a percentage of the segment length considered, and computed independently for the three scenes of the experiment. It can be seen that our approach is in all cases superior to the other two methods, being always under 1% RMSE for all three scenes. PL-ICP is the second best candidate, having around 5% RMSE, except for the long corridor (Scen. 3) where it completely fails.

Finally, from the computational point of view, the last columns on Table I show the runtimes in milliseconds measured on an AMD Phenom II X6 1035T CPU at 2.6 GHz. Overall, RF2O takes less than 1 ms, followed by PSM with 2.85 ms and PL-ICP with more than 19 ms. Taking this into account, the presented approach not only provides more accurate estimates but it is also much faster than its competitors.

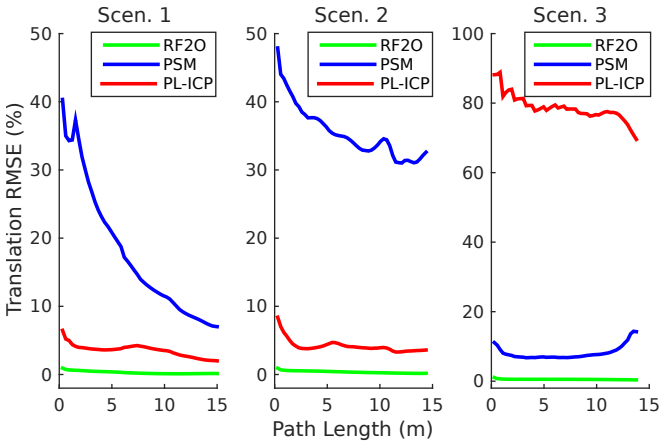

Fig. 3. Translation errors averaged over all sub-sequences of a given length for the three scenes of the simulated experiment.

B. Real Experiment

To validate the results obtained in the simulated experiments, we employ a real mobile robot equipped with a Hokuyo laser scanner which navigates in an office-like environment. Making use of a mobile robot allows us to include the odometry estimates from the onboard encoders (a pair of low-cost AMT102-v incremental encoders from CUI Inc.), but prevents us from performing a quantitative comparison given the lack of a precise ground truth. Therefore, in this section the different methods are compared just qualitatively by plotting 2D maps built purely from the odometry pose estimates. In other words, for each method we present maps built as a concatenation of 2D point clouds along their estimated trajectories, without resorting to global consistency or any other mapping strategy.

The path covered by the robot during this experiment is roughly 49 meters long, and is travelled at an average speed of 0.535 m/s (max speed of 0.6 m/s). For all methods, we set the scan rate at 10 Hz. Fig. 4 plots the maps built from the trajectory estimates of the different approaches. As a reference, we plot the map built from the accurate localization provided by [6], which does not compute odometry but finds the pose of the robot within a previously built map. As can be seen, the map derived from our odometry estimation is noticeably closer to the reference map than any of the others. PL-ICP provides the second best estimation after RF2O, failing mostly in the corridor areas (see Section VII-A), which results in a shortening of the map and overlapping of scan points in such areas. PSM and the encoder-based maps follow the comparison, being the latter the worst of all of them, with difference.

C. Robustness Against Noise and Non-static Environments

Finally, we analyze how noise and moving objects affect the motion estimation of the proposed method, i.e., when the assumption of a static environment is violated. Therefore, this section is composed of two experiments. The first one aims to evaluate the drift of the compared methods caused by the noise of the measurements. To this end, a real experiment is conducted where a lidar working at 10 Hz is kept still in a static environment for a time lapse of three minutes. Under this setup, since the only error involved is the sensor noise, the outcome represents how noise affects the motion estimates of the different methods. Table II shows the relative deviations per second of the compared methods. We have also considered a simplified version of our approach (RF2O-NC), where we remove the Cauchy M-estimator and simply minimize the squared residuals (see (11)). From these results we can conclude that both RF2O and PL-ICP are equally good at translations, being marginally worse than the non-robust RF2O-NC, while, at rotations, PL-ICP is slightly superior than the others.

Then, a second experiment is conducted in the same scenario but, in this case, several moving objects are introduced. During the experiment two persons are walking around the robot, opening and closing a door and displacing a cardboard box (see Fig. 5). The reader is encouraged to watch the demonstration video where the experiment is shown in detail (`http://mapir.isa.uma.es/work/rf2o`).

TABLE II

TRANSLATIONAL AND ROTATIONAL DEVIATIONS PER SECOND: ROBUSTNESS AGAINST NOISE AND MOVING OBJECTS.

Static Experiment

RMSE	RF2O	RF2O-NC	PSM	PL-ICP
Translation (cm/s)	0.125	**0.113**	0.268	0.125
Rotation (deg/s)	0.075	0.064	0.216	**0.043**

Moving Objects Experiment

RMSE	RF2O	RF2O-NC	PSM	PL-ICP
Translation (cm/s)	0.636	0.879	3.548	**0.412**
Rotation (deg/s)	0.267	0.321	1.091	**0.082**

As can be seen in the second part of Table II, PL-ICP is the most robust method in such situations, followed by the proposed RF2O. It is important to notice that, although PL-ICP is between two and three times better than RF2O, the magnitude of the errors is still pretty small for both methods, unlike the PSM estimates, which show important translational and rotational drifts. Finally, comparing the two versions of our approach, it can be noticed that under the presence of moving objects, the Cauchy M-estimator provides results that are 25% more accurate than those obtained with standard quadratic minimization.

VIII. CONCLUSIONS

We have presented a novel approach named RF2O to estimate the planar motion of a lidar by imposing the range flow constraint equation on consecutive scan pairs. Extensive experiments have been carried out to demonstrate the accuracy of our method, and comparisons with point-to-line ICP, Polar Scan Matching and the standard wheel odometry have been performed under different scenarios and frame rates. Results show that RF2O provides the most accurate estimates for both translations and rotations, except for non-static environments, where PL-ICP is slightly superior. With a reported runtime of barely 1 millisecond, planar motion can be easily estimated with almost no computational cost, which makes this method attractive for many robotic applications

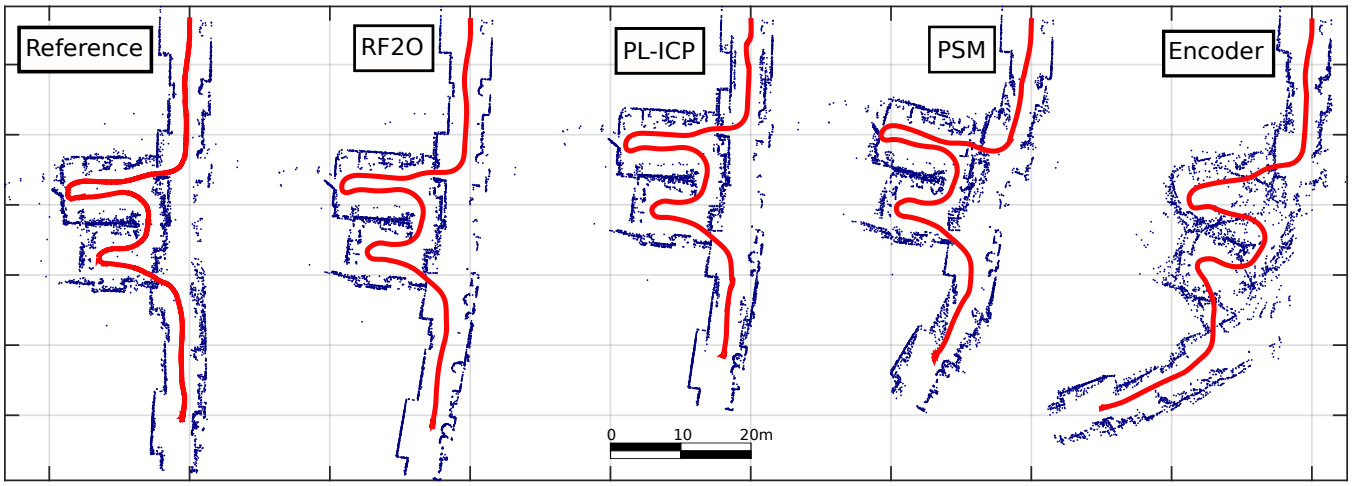

Fig. 4. Maps built as a concatenation of 2D point clouds along the estimated trajectories for different methods. The reference map is built using the accurate localization of a particle filter-based approach. Trajectories are shown in red and the scanned points in blue.

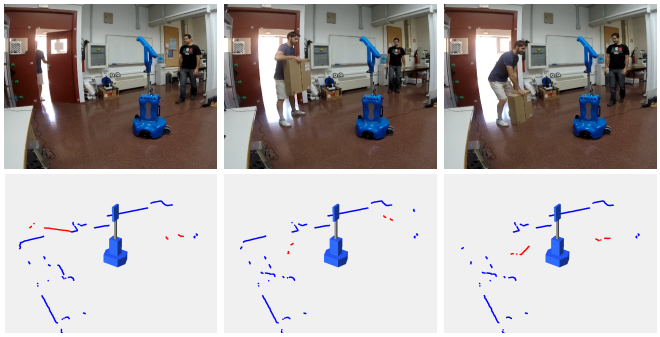

Fig. 5. First row: Sequence of images taken during the second experiment described in the section VII-B. Second row: 3D representation of the scans and the robot at those particular instants, where the non-static points are shown in red.

that are computationally demanding and require real-time performance. For future work, we plan to analyze the effect of small deviations from planar motion, which might be useful if this method is applied to estimate the motion of a quadcopter or a vehicle with strong dynamics.

References

[1] O. J. Woodman, "An introduction to inertial navigation," *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, vol. 14, 2007.

[2] J. Gonzalez-Jimenez and R. Gutierrez, "Direct motion estimation from a range scan sequence," *Journal of Robotic Systems*, vol. 16, no. 2, pp. 73–80, 1999.

[3] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE Int. Conference on Robotics and Automation (ICRA)*, 2008, pp. 19–25.

[4] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125–1153, 2007.

[5] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[6] J. L. Blanco, J. Gonzalez-Jimenez, and J. A. Fernandez-Madrigal, "Optimal filtering for non-parametric observation models: applications to localization and SLAM," *The International Journal of Robotics Research*, vol. 29, no. 14, pp. 1726–1742, dec 2010.

[7] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[8] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to SLAM," in *IEEE Int. Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 3317 – 3322.

[9] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL tentative*, 1992, pp. 586–606.

[10] J. Gonzalez, A. Stentz, and A. Ollero, "A mobile robot iconic position estimator using a radial laser scanner," *Journal of Intelligent and Robotic Systems*, vol. 13, no. 2, pp. 161–179, 1995.

[11] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[12] E. B. Olson, "Real-time correlative scan matching," in *IEEE Int. Conference on Robotics and Automation (ICRA)*, 2009, pp. 4387–4393.

[13] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," in *Unmanned Systems Technology XI, SPIE*, 2009, pp. 733 219–733 219.

[14] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[15] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[16] H. Spies, B. Jahne, and J. L. Barron, "Range flow estimation," *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 209–231, 2002.

[17] R. Battiti, E. Amaldi, and C. Koch, "Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy," *International Journal of Computer Vision*, vol. 6, no. 2, pp. 133–145, 1991.

[18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conference on Computer Vision (ECCV)*, 2004, pp. 25–36.

[19] M. Jaimez and J. Gonzalez-Jimenez, "Fast visual odometry for 3-D range sensors," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 809 – 822, 2015.

[20] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE Int. Conference on Intelligent Robot Systems (IROS)*, 2012.

[22] J. Gonzalez-Jimenez, C. Galindo, and J. R. Ruiz-Sarmiento, "Technical improvements of the Giraff telepresence robot based on users evaluation," in *IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. Int. Conference on Computer Vision (ICCV)*, 2012, pp. 3354–3361.