ISPRS Journal of Photogrammetry and Remote Sensing 121 (2016) 48–59

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Localization of a mobile laser scanner via dimensional reduction

Ville V. Lehtola [a,b,*], Juho-Pekka Virtanen [a,b], Matti T. Vaaja [a], Hannu Hyyppä [a], Andreas Nüchter [c,*]

[a] Institute of Measuring and Modeling for the Built Environment, Aalto University, P.O. box 15800, 00076 Aalto, Finland
[b] National Land Survey of Finland, Finnish Geospatial Research Institute FGI, PL 84, 00521 Helsinki, Finland
[c] Julius Maximilians University Würzburg, Informatics VII – Robotics and Telematics, 97074 Würzburg, Germany

ARTICLE INFO

Article history:
Received 22 May 2016
Received in revised form 16 August 2016
Accepted 16 September 2016

Keywords:
Localization
Positioning
3D reconstruction
Lidar
Slam

ABSTRACT

We extend the concept of intrinsic localization from a theoretical one-dimensional (1D) solution onto a 2D manifold that is embedded in a 3D space, and then recover the full six degrees of freedom for a mobile laser scanner with a simultaneous localization and mapping algorithm (SLAM). By intrinsic localization, we mean that no reference coordinate system, such as global navigation satellite system (GNSS), nor inertial measurement unit (IMU) are used. Experiments are conducted with a 2D laser scanner mounted on a rolling prototype platform, VILMA. The concept offers potential in being extendable to other wheeled platforms.

1. Introduction

Localization of a mobile laser scanner (MLS) without using a global reference coordinate system, e.g., satellites, is one of the grand problems in laser scanning research (Lehtola et al., 2015, 2016; Bosse et al., 2012; Bosse and Zlot, 2009; Lauterbach et al., 2015; Liu et al., 2010; Vosselman, 2014; Kaul et al., 2016). Besides its theoretical importance, prominent solutions enable applications in indoor environments, construction and forest settings, and other areas that lack satellite coverage, e.g., planetary sites. If, in addition, the localization is done without any external sensors, such as an inertial measurement unit (IMU), we call it *intrinsic localization*.

The problem is formidable because of its inverse nature. The time-of-flight data of a laser scanner typically results to point clouds of dozens or hundreds of millions of points. Recovering the scanner trajectory by processing this data requires a sophisticated way to reduce its size by identifying invariants, or features, that describe the environment in a sufficient manner. Then the problem is preferably divided into steps, so that one and only one unknown variable is solved in each step, thus keeping the solution space computationally tracktable. Previously, we have tried to apply a 6 DoF semi-rigid SLAM method directly on a 1D trajectory solution in order to correct it into its original physical form (Lehtola et al., 2016), but this approach cannot deal with steep curves because it is based on the iterative closest point (ICP) algorithm. Specifically, when a local minimum in the *n*-scan matching is reached, the iteration gets stuck, and further computation does not help in recovering the actual trajectory.

In this paper, we set out to fix this, and to extend the concept of intrinsic localization from the one-degree-of-freedom solution (Lehtola et al., 2015) back to the six degrees of freedom. This is done in three steps in Section 2. First, local corrections on a horizontal 2D plane are introduced to the 1D trajectory. This is done not by optimizing over the whole 3D point cloud, but rather by dividing the one big problem into smaller ones by considering the trajectory in separate segments. Second, similar local corrections are introduced on a vertical 2D plane. Third, a new local filtering paradigm is introduced to compute features from the time-of-flight measurements to make the previous two steps computationally tracktable. The previously used 6 DoF semi-rigid SLAM implementation is used to bring the trajectory estimates close to the actual trajectory. Results are presented in Section 3, with data gathered using our existing platform, VILMA, and a survey-grade terrestial laser scanner[1] (TLS) is used to provide reference results. Discussion is in Section 4, and Section 5 concludes the paper.

* Corresponding authors at: Institute of Measuring and Modeling for the Built Environment, Aalto University, P.O. box 15800, 00076 Aalto, Finland (V.V. Lehtola).
E-mail address: ville.lehtola@iki.fi (V.V. Lehtola).

[1] Faro Focus 3D 120.

http://dx.doi.org/10.1016/j.isprsjprs.2016.09.004
0924-2716/

1.1. Related work

Mobile mapping systems deliver 3D data while moving profilers along a trajectory. The trajectory can be recovered by measuring the system motion, combined with extrinsic calibration, i.e., the process of estimating the position and orientation parameters of a sensor system. Recent approaches to calibration of laser scanners, which is also called boresight calibration (Skaloud and Lichti, 2006; Rieger et al., 2010) include statistical methods using sophisticated error functions (Underwood et al., 2009; Sheehan et al., 2011). However, VILMA does not need extrinsic calibration.

The other way to recover the trajectory is intrinsic. Then the focus is on the manipulation of the computational trajectory, commonly known as simultaneous localization and mapping (SLAM). SLAM has long roots in the history of robotics. Approaches include EKF-SLAM (Dissanayake et al., 2000), FastSLAM (Montemerlo et al., 2002), FastSLAM 2.0 (Montemerlo and Thrun, 2007), and GraphSLAM (Thrun and Montemerlo, 2006), including early approaches to 3D mapping (Thrun et al., 2000). Of these, GraphSLAM uses sparse matrices to represent a graph of observation interdependencies, i.e., as extended incidence matrix, and in this sense, its relative in computer vision can be thought to be the bundle adjustment, and its variations, referred broadly to as structure from motion techniques (see e.g. Triggs et al. (2000), and refs there-in). Acquisition of different 3D point clouds from the latter established the need for the well-known iterative closest point (ICP) algorithm that was developed by Besl and McKay (1992), Chen and Medioni (1992), and Zhang (1994). Meanwhile in the robotics community, Lu and Milios (1994) came up with its 2D variant. The input was scan data acquired by a robot with a horizontally mounted profiler, i.e., a 2D safety scanner. Based on this 2D ICP, Lu and Milios (1997) presented an ICP-like GraphSLAM solution, and its extension to 3D scans and poses with six degree of freedom was performed by Borrmann et al. (2008) and Nüchter et al. (2010).

Recent development that begun with approaches that cut the trajectory into segments and performed some globally consistent scan matching on the segments (Stoyanov and Lilienthal, 2009; Bosse and Zlot, 2009), has led towards continuous-time SLAM (Anderson and Barfoot, 2013; Anderson et al., 2014). However, the realization of these often focuses on cameras with rolling shutters instead of event-based vision sensors (Mueggler et al., 2011). Furthermore, the methods of Alismail et al. (2014) and Anderson et al. (2015) are designed for scanners that differ from the ones intended for VILMA, both in terms of the data rate and the modality of the data. To our best knowledge no other intrinsic localization solutions exist. Otherwise, the work closest to ours may be the one of Zhang and Singh (2014) on Lidar Odometry and Mapping (LOAM) that employs external angular measures to perform localization.

1.2. Initial straight trajectory estimate (1 DoF)

The localization of the laser data requires a successful reconstruction of the sensor trajectory. The trajectory $j(t)$ is time-dependent with six degrees of freedom, namely, three from location and three from orientation. We write out

$$j(t) = [\theta(t), \psi(t), \phi(t), x(t), y(t), z(t)]^T \tag{1}$$

where θ is the pitch, ψ is the roll, and ϕ is the yaw angle. Time is denoted by t. Without any reference coordinate system, the successful reconstruction of the trajectory requires that these degrees of freedom are eliminated. Previously, this was done for a holonomic system in one dimension (1D) (Lehtola et al., 2015). We briefly outline this solution here.

To capture a 3D environment with a 2D laser scanner, the scanner has to be rotated about at least one axis. The 2D scanner is mechanically attached onto a round platform, see Fig. 1, so that it can only rotate about one axis of rotation, namely θ. Therefore rotational degrees of freedom are reduced by two, i.e., φ and ψ are constant. The platform does not slip against the floor, and so x, y, and z all become direct functions of θ. This assumption gets relaxed during the SLAM step of Section 2.3.

The main angle of rotation $\theta(t)$ is the path parameter that describes the scanner trajectory, and obtaining it, as follows, solves localization in 1D. The scanner sits on the hypotenuse at a distance of $R_0 - R_1$ from VILMA's central axis, where $R_1 = 0.13$ m, and $R_0 = 0.25$ m is the radius of the metal disk. Assuming that the floor is flat, simple trigonometry is employed to write

$$\theta = \arccos(R_0/d) \tag{2}$$

where $d = d_m + (R_0 - R_1)$, and d_m is the minimum measured distance to the floor over one full 2D circle observation, a so-called slice. Considering the minimum distance to the floor, the scanner's position on disk radius varies between two values, depending on whether the scanner is upside down, Eq. (2), or upside up, in which case $\theta = \pi - \arccos(R_0/d)$, with $d = R_0 + \cos 27.5\,\mathrm{deg}(d_m - R_2)$, and $R_2 = 0.42$ m. Here, the 27.5deg is half of the dead angle of the scanner. The angle θ is incremented by 2π for each cycle that the platform rolls. Each time the 2D scanner is perpendicular towards the floor (PTF), $\theta(t) = \pi + 2\pi n,\ n = 0, 1, 2, \ldots$, the scanning distance reduces to the minimum R_1. We call this a PTF-observation, and keep track of these occurrences in the laser data series obtaining a time series. The PTF observation is robust to error, since data points from a large field of view can be used to interpolate the floor point precisely below the sensor. Also, stochastic errors in PTF observations do not cumulate with time as long as the no-slip condition with the floor applies. Once $\theta(t)$ is obtained, the coordinate transform for the 2D sensor data (X, Z) is obtained considering the trajectory of a contracted cycloid,

$$\begin{cases} x = X \\ y = R_0\theta + (R_0 - R_1)\sin\theta + \sin(\theta)Z\,, \\ z = R_0 + (R_0 - R_1)\cos\theta + \cos(\theta)Z \end{cases} \tag{3}$$

where (x, y, z) are the coordinates of the resulting 3D point cloud, and the platform trajectory is

$$\vec{j}_{1\mathrm{D}}(t) = \begin{pmatrix} 0 \\ R_0\theta(t) \\ 0 \end{pmatrix}. \tag{4}$$

2. Back to the six degrees of freedom

In 1D, the localization is done knowing only the initial and the current state of the system, as is obvious from Eq. (4). In two or more dimensions, however, the trajectory reconstruction by path integration requires accurate measuring of the position all along the path. Therefore, we use 6 DoF SLAM. Still, this requires a relatively good initial trajectory estimate, and next we concentrate on how to obtain that. The overall algorithm, and the contribution of this paper, are illustrated in Fig. 2.

Each time-of-flight measurement, or point, is connected to the trajectory, at a position $s(t)$ from where the measure was made at time t. These positions are discretized with respect to time into N slices, where each slice contains one 2D profile, i.e., points from the full rotation of the mirror of the scanner. From the 1D solution of Eq. (4), we have the trajectory divided into parts of length

$$l_n = (\theta_n - \theta_{n-1})R_0, \tag{5}$$