Direct Lidar Odometry for a Rotating Multi-beam Lidar

Taewon Kim[1], Youngsoo Choi[2]

1. Nuclear Robot.Diagnosis Laboratory, Korea Atomic Energy Research Institute, Daejeon, Korea (traum100@gmail.com)
2. Nuclear Robot.Diagnosis Laboratory, Korea Atomic Energy Research Institute, Daejeon, Korea (yschoi1@kaeri.re.kr)

Abstract: In this paper, we propose a novel lidar odometry algorithm. It uses only a rotating multi-beam lidar without any other sensors, but is fast, precise and robust. Existing state-of-the-art methods use feature matching or finding the nearest neighbor to calculate cost between two point clouds. However, the proposed method calculates the cost directly using cylindrical projection instead of feature matching and finding the nearest neighbor. This is an extension of the direct method to a rotating multi-beam lidar. We also propose reliability weighting. When the pose estimation is finished, the reliability weight is calculated between the two cylinder images and used as the initial weight of the next step. This reduces the effects of moving objects and occlusion. Finally, we evaluate the algorithm in the KITTI odometry benchmark.

Keyword: Lidar odometry, Direct method, Motion estimation, SLAM

1 Introduction

With the development of a rotating multi-beam lidar, it has become possible to acquire dense point clouds in a fast cycle. A point cloud is suitable for the SLAM algorithm because there is a lot of overlap with previous data. However, existing lidar based SLAM algorithms have not used it effectively.

The existing lidar odometry algorithm can be classified as either a feature based method or an ICP based method. The feature based method calculates the cost between matched features[1]. The ICP based method searches for the nearest neighbors and calculates the cost. It requires a large number of computations to find the feature matching and the nearest neighbors[2][3]. Thus, they use interesting or sampled parts in the dense data.

There was a similar problem in vision-based SLAM. To solve this problem in using dense data, many vision-based SLAM algorithms use a direct method[4][5]. A direct method does not create features or find the nearest neighbors. It only optimizes the cost in a well-organized grid space called an image plane.

In this paper, we propose a direct lidar odometry algorithm. The proposed method places the point cloud into the cylinder image using a cylindrical projection. The cylinder image is a well-organized grid space such as an image plane in vision-based SLAM. Thus, we can apply the direct method to lidar data.

The proposed method calculates the cost by projecting the point cloud of the current frame onto the cylinder image of the previous frame. In addition, we use the Levenberg-Marquardt algorithm to find the pose that minimizes the cost[6]. In addition, we propose reliability weighting to reduce the influence moving objects.

The rest of this paper is organized as follows. Section 2, explains how we applied the direct method to the lidar odometry. In section 3, experimental results and a discussion regarding the KITTI odometry benchmark datasets[7] are described. Finally, conclusions and areas of future work are discussed in Section 4.

2 Approach

2.1 Cylindrical projection

A point cloud during a 360-degree rotation of the lidar is defined as one frame. When the frame is grabbed, every point is a projected cylinder image using

$$C(x,y,z)=\begin{bmatrix}u\\v\end{bmatrix}=\begin{bmatrix}-s_u\tan^{-1}\left(\dfrac{y}{x}\right)+c_u\\-s_v\dfrac{z}{\sqrt{x^2+y^2}}+c_v\end{bmatrix}, \qquad (1)$$

where x, y, z are elements of a point, u, v are coordinates of the cylinder image, s_u, s_v are scale factors, and c_u, c_v are the center of the cylinder image. Fig. 1 shows the cylinder image. Although expressed as a depth image for visibility, each grid actually contains 3D points.

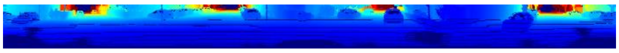

Fig.1 An example of a cylinder image

Then, a cylinder image is used to calculate the surface normal using points around the center point. It can be calculated easily because the point clouds are already arranged in the grid space. As a result, the 3D point and surface normal are stored in each grid of the cylinder image. We define this as a surfel frame $f^k = \left\{ f^k_{xyz}, f^k_{nor} \right\}$. Here, k is index of the surfel frame.

2.2 Frame to frame motion estimation

2.2.1 Direct method

Given surfel frames f^{k-1} and f^k, the pose between them is defined as $\xi^k_{k-1} \in \mathbb{R}^6$ which represents elements of **se**(3). To calculate the pose, we solve

$$\underset{\xi^k_{k-1}}{\operatorname{argmin}} \left(\sum_{\mathbf{p}_i \in f^k} \mathrm{D}\left(f^{k-1}(\mathbf{p}'_i),\ \mathrm{G}\left(f^k_{xyz}(\mathbf{p}_i), \xi^k_{k-1} \right) \right)^2 \right), \quad (2)$$

where $\mathbf{p}_i \in \mathbb{R}^2$ is a 2D index of the surfel frame f^k, function G is a rigid body transformation using **se**(3), D is the point-plane distance function, the C is the cylindrical projection function described in Equation (1) and

$$\mathbf{p}'_i = \mathrm{C}\left(\mathrm{G}\left(f^k_{xyz}(\mathbf{p}_i), \xi^k_{k-1} \right) \right). \quad (3)$$

To obtain the optimal pose, we use the weighted Levenberg-Marquardt method. Here, ξ^k_{k-1} is updated by

$$\xi \leftarrow \xi - \left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J} + \lambda \operatorname{diag}\left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J} \right) \right)^{-1} \mathbf{J}^{\mathrm{T}}\mathbf{d}, \quad (4)$$

where λ is a damping factor and **W** is a weight matrix which is a combination of the Huber weight and the reliability weight. In addition, **d** is a vector, which is a stack of the cost

$$d_i = \mathrm{D}\left(f^{k-1}(\mathbf{p}'_i),\ \mathrm{G}\left(f^k_{xyz}(\mathbf{p}_i), \xi^k_{k-1} \right) \right). \quad (5)$$

Here, **J** is a matrix, which is the stack of $\mathbf{J}_i$. It is a Jacobian vector of Equation (5) about ξ^k_{k-1}. The global pose can be calculated through proper multiplication,

$$\xi^k_0 = \log\left(\exp\left(\xi^{k-1}_0 \right) \cdot \exp\left(\xi^k_{k-1} \right) \right). \quad (6)$$

2.2.2 Jacobian approximation

To solve Equation (2), we have to calculate **J**. Expressing $\mathbf{J}_i$ by the chain rule,

$$\mathbf{J}_i = \frac{\partial \mathrm{D}}{\partial \xi^k_{k-1}} = \frac{\partial \mathrm{D}}{\partial f^{k-1}} \frac{\partial f^{k-1}}{\partial C} \frac{\partial C}{\partial G} \frac{\partial G}{\partial \xi^k_{k-1}} + \frac{\partial D}{\partial G} \frac{\partial G}{\partial \xi^k_{k-1}}. \quad (7)$$

In Equation (7), we focus on

$$\frac{\partial f^{k-1}}{\partial C}. \quad (8)$$

Equation (8) is a surfel gradient on the surfel frame. Thus, we can assume it to be zero. because the nearby surfels are on the same plane. Thus, the final form of $\mathbf{J}_i$ is

$$\mathbf{J}_i = \frac{\partial \mathrm{D}}{\partial \xi^k_{k-1}} \approx \frac{\partial D}{\partial G} \frac{\partial G}{\partial \xi^k_{k-1}}. \quad (9)$$

2.2.3 Reliability weighting

To reduce the influence of the moving objects, we propose a reliability weighting that is used with the Huber weight to help converge Equation (4). The weight is

$$W_i = \mathrm{W}_{\mathrm{Huber}}(\mathrm{d_i}) \cdot R^{k-1}(\mathbf{p}'_i), \quad (11)$$

where $R^{k-1}(\mathbf{p}'_i) \in \mathbb{R}^1$ is the value of the reliability map at $\mathbf{p}'_i$. The reliability map is the same size as the surfel frame. It is created using the difference between the two frames, which is

$$R^k(\mathbf{p}_i) = \frac{1}{\left\| f^k_{xyz}(\mathbf{p}_i) - \mathrm{G}\left(f^{k-1}_{xyz}(\mathbf{p}_i), \xi^k_{k-t} \right) \right\|}, \quad (12)$$

where t is index of a few seconds, ξ^k_{k-t} is the pose between k-t through k. Assume that the pose estimates from k-t through k frames are correct. In the next pose estimation, the moving object has a low weight, and the static object has a high weight.

3 Experiments and discussion

We evaluated the proposed algorithm in the KITTI odometry benchmark datasets. This dataset was created by driving in an urban area with a Velodyne HDL-64E sensor installed on the car, and provides the ground truth using GPS/INS. The sensor is a 64-line multi-beam rotating lidar. The horizontal field of view (FOV) is 360 degrees and the vertical FOV is about 27 degrees.

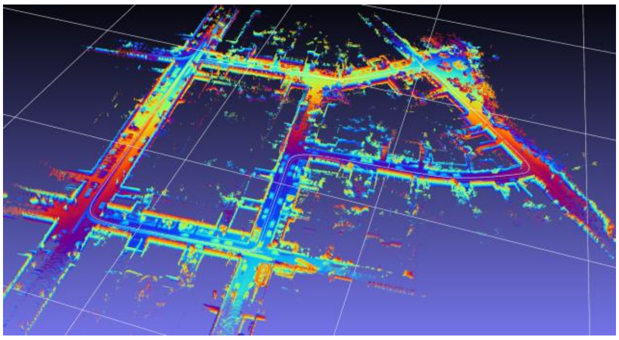

Fig.2 The point cloud map of the proposed method

The experiment was performed on sequence #7 and accumulated a delta pose between f^{k-1} and f^{k}. We did not use the loop closing or any global registration technique. Therefore, only the performance of the proposed algorithm is shown.

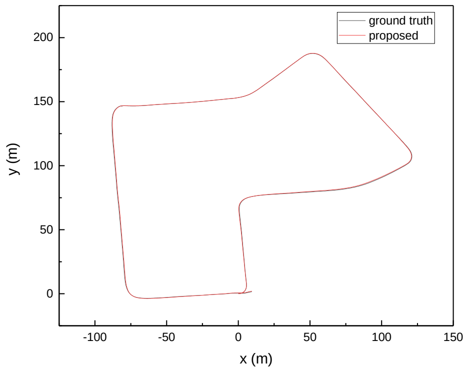

Fig.3 Comparison of the results of proposed method with the ground truth in sequence #7

The result from the x-y plane is close to the ground truth. It simply accumulates poses between frames, but there are few cumulative errors because we can effectively use a sufficient amount of nested data in the horizontal direction through the direct method.

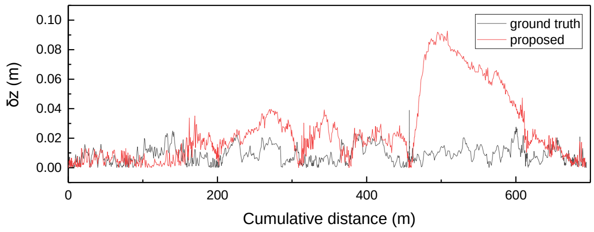

Fig.4 Comparison of the frame to frame pose estimation results for z-axis

Comparing the z-axis shows that the error is large because the vertical resolution of the sensor is lower than the horizontal resolution. The data used for vertical optimization is only 7% of the horizontal direction. In addition, since the FOV is narrow, the overlap region is small. Considering the loop closing or using the global registration techniques will obtain improved results.

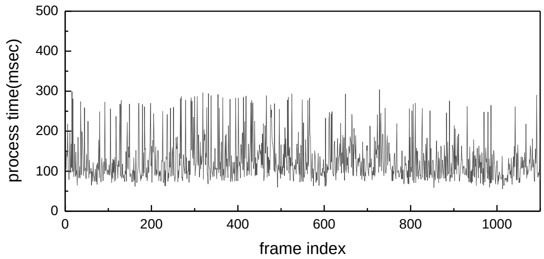

Fig.5 Process time

With the 4.2 GHz CPU, the algorithm process time is 127 ms on average. The size of the surfel frame is 900×64 and uses about 35,000 ~ 40,000 data per frame. The proposed method uses a large amount of data, but is fast because it does not find the nearest neighbors. In addition, because it has massive parallelism, it is suitable for parallel processing using GPGPU technology.

4 Conclusions and future work

In this paper, we propose a novel lidar odometry algorithm using a direct method. It uses the cylindrical projection to store the surfels in a 2D grid space, and then defines the cost function for the direct method and calculates the pose between frames through the Levenberg-Marquardt method. In addition, we also proposed the use of a simplified Jacobian and reliability weighting. The proposed method was evaluated on the KITTI odometry benchmark dataset and was proved to be an effective method for dense and overlapped data.

In the future, we will improve the accuracy of the vertical direction by considering the loop closing and the global registration technique, and apply the direct method to the continuous-time SLAM. We will also use the GPGPU technology to speed up the algorithm by taking advantage of the massive parallelism of the direct method.

References

[1] J. Zhang and S. Singh, “LOAM : Lidar Odometry and Mapping in Real-time,” Robotics: Science and Systems Conference (RSS), 2014.

[2] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in Proc. Third Int. Conf. 3-D Digit. Imaging Model., pp. 145– 152, IEEE Comput. Soc, 2001.

[3] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in Robotics: Science and Systems (RSS), pp. 435, 2009.

[4] Engel, Jakob, Thomas Schöps, and Daniel Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” In European Conference on Computer Vision, pp. 834-849. Springer, Cham, 2014.

[5] Newcombe, Richard A., Steven J. Lovegrove, and Andrew J. Davison, "DTAM: Dense tracking and mapping in real-time," Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 2320-2327, 2011.

[6] Ranganathan, Ananth. "The levenberg-marquardt algorithm," Tutoral on LM algorithm 11.1, pp. 101-110, 2004.

[7] Geiger, Andreas, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, pp. 3354-3361, 2012.

[8] Moosmann, Frank, and Christoph Stiller. "Velodyne slam," Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, pp. 393-398, 2011.

[9] Bosse, Michael, Robert Zlot, and Paul Flick. "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," IEEE Transactions on Robotics 28.5, pp. 1104-1119, 2012.

[10] Forster, Christian, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry," Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, pp. 15-22, 2014.