Improving SLAM in Pipe Networks by Leveraging Cylindrical Regularity*

R. Zhang, M. H. Evans, R. Worley, S. R. Anderson and L. Mihaylova

[1] The University of Sheffield, Sheffield South Yorkshire S10 2TN, UK
[2] {rzhang47,mat.evans,rfworley1,s.anderson,l.s.mihaylova}@sheffield.ac.uk

Abstract. Monocular visual Simultaneous Localisation and Mapping algorithms estimate map points and frame poses simultaneously based on video data. The estimated map point locations do not contain any structural information. Due to the measurement noise, the estimated trajectory is slightly different from the ground truth. This paper improves the estimation accuracy of trajectory in a pipe network by leveraging structural regularity. An optimisation-based method is used to detect a cylinder among map points in the SLAM back-end. When the cylinder is detected, the system enforces cylindrical regularity to the points from the cylindrical pipe surface, which is named cylindrical points. The estimated trajectory and map points will benefit from this structural information. This method is verified and evaluated on both synthetic data and real-world pipe video datasets.

Keywords: SLAM · Structural regularity · Pipe networks inspection · Structure from motion.

1 Introduction

Pipe networks play an important role in transporting resources such as water, oil, and natural gas. Due to their low cost and efficiency, pipelines are widely found in cities and industry. However, pipe networks may suffer from defects, such as blockages or leakage, leading to economic losses, environmental contamination, and damage to health [2]. Nondestructive inspection or testing is an essential task, which can provide early detection of defects and avoid undesirable results.

Cameras are low-cost, effective sensors providing detailed appearance in an immersive fashion, and there is an extensive existing literature on *Structure from Motion* (SfM) and visual *Simultaneous Localisation and Mapping* (SLAM) [3]. The majority of vision-based methods focus on SfM [17], which take advantage of the structural information of pipes, but impose strong constraints on the robot motion or cannot perform in real-time. Visual SLAM methods enable an agent equipped with a camera to explore its environment and build a sparse point cloud simultaneously [4,6,16]. These can be divided into feature-based methods, which

* This work is supported by the UK's Engineering and Physical Sciences Research Council (EPSRC) Programme Grant EP/S016813/1.

track features between video frames and estimate frame poses and map points by reducing reprojection error [16], and direct methods, which track informative points and estimate the frame poses by minimizing the photometric error [4].

In the narrow space of the pipe, visual SLAM algorithms suffer from low parallax. The estimated pipe becomes less cylindrical and shapes more like a cone, and the travel distance is underestimated. In order to improve the accuracy in a man-made environment, a natural choice is structure SLAM algorithm which uses structural information such as knowledge of lines or planes. However, previous SLAM algorithms do not consider cylinders, and the SfM methods cannot be used directly for visual SLAM in pipe networks.

In this paper, we propose a new extension to the popular ORB-SLAM2 system [16] to leverage cylindrical regularity in a pipe network. The *main contributions of this work* include: 1) An iterative cylinder detection method based on sparse points, selecting cylindrical points within 95% confidence intervals; 2) In order to leverage the cylindrical information through all related frames, our method estimates the cylinder based on optimised map points, which is different from existing structure SLAM algorithms which detects structural regularities among local map points; 3) The method is verified on synthetic and real-world data. The results demonstrate that the cylindrical regularity improves the accuracy of estimated camera trajectories and sparse points clouds.

The remainder of the paper introduces related work in Section 2, and the proposed method is described in Section 3. Section 4 introduces how the system detects the cylinder. Section 5 presents the cost function and the cylindrical regularity. Section 6 describes the experiment setups and results. Finally, conclusions and plans for future work are summarised in Section 7.

2 Related Work

In man-made environments with many higher-level features and structural regularities, many robots use SLAM algorithms designed to exploit this information to improve localization accuracy. There are two main ways to incorporate structural regularities. The first approach uses the *Manhattan world* [20] which abstracts the man-made environment as a set of blocks. However, many requirements for this approach are not met in the pipe environment, such as lines in orthogonal directions and the vanishing points. The second approach is to add structural regularities to landmarks, including points, edges, and lines, in the optimisation [7, 9, 13]. These algorithms also optimise the structural parameters in the local optimisation, including other landmarks. However, these specific structural regularities are not present in the buried pipe environment.

A key structural property of pipe environments are their cylindrical shape, and some Structure from Motion (SfM) methods have been designed to exploit this information. Some methods [8, 19] estimate the map point location by computing the intersection of the known cylindrical surface and the ray from the camera centre to the observation when the camera moves parallel with the pipe axis. Others methods [8,10] add cylindrical regularity to map points in the bundle

adjustment (BA) algorithm to keep the points on the known cylindrical surface. Further, this prior knowledge has been used in local pose optimisation, which can optimise the camera poses and triangulate features iteratively [12].

To date, SfM approaches have typically assumed that the robot moves forward parallel with the pipe axis, where the system benefits most from the cylindrical information [8, 12]. In some cases, the pose estimation is simplified by assuming the robot moves in a straight line [3, 8, 19]. Those assumptions do not typically hold in the SLAM problem. Cone detection among the triangular features has been used to lift restrictions on camera movement [10], which can be done without prior knowledge of the pipe axis, and is less sensitive to camera calibration [14]. This method detects multiple pipe instances with temporary map points incrementally per reconstructed model, which can not perform in real-time.

In summary, although many SfM algorithms exploit cylindrical information, SLAM algorithms cannot implement these methods directly since they incorporate some prior knowledge that is not available in the SLAM problem formulation. Meanwhile, SLAM algorithms have yet to incorporate cylindrical structure information. This paper aims to address this gap.

3 System Overview

The proposed system is derived from ORB-SLAM2 [16] which is a well-known visual SLAM algorithm. ORB-SLAM2 has three threads: tracking, local mapping and loop closing. The conventional ORB-SLAM2 selects frames with much new information as keyframes in the tracking thread. In the local mapping thread, the system triangulates features and optimises keyframe poses and map points in local optimisation. The proposed system applies cylinder-related operations in the second thread. Optimised map points are defined as map points that are optimised in the previous optimisation and excluded from current local optimisation. When enough optimised points are obtained since the beginning of mapping or the end of the last cylinder model, the system estimates a cylinder among optimised map points and select cylindrical points among local map points. The system punishes the distance of the cylindrical points from the cylindrical surface in local BA. Finally, the system culls some map points far outside of the estimated pipe model, considered outliers. Once the ratio of cylindrical points to all local map points is lower than a threshold, the system will stop using cylindrical regularity and prepare to update the cylinder model with new cumulative optimised points.

4 Cylinder Detection

Given a set of 3D map points, the cylinder is detected based on geometric properties. In order to leverage cylindrical regularity in local optimisation, it is essential to estimate the cylinder accurately and distinguish the cylindrical points clearly. This section introduces cylinder representation and cylinder detection.

4.1 Cylinder Representation and Estimation

The cylinder estimation is based on cylinder representation. The cylinder is denoted by $\vec{\pi} = \left[\vec{L_c}^{\mathrm{T}}\ \vec{O_c}^{\mathrm{T}}\ r_c\right]^{\mathrm{T}}$, where $\vec{L_c} \in R^3$ is a vector characterising the direction of the cylinder's axis, $\vec{O_c} \in R^3$ represents the 3D coordinates of the intersection point between the pipe axis and the xy plane of the world coordinate, and r_c is the radius of the cylinder. Here $\vec{P_w^i} \in R^3 (0 < i < n)$ represents a cylindrical point. These elements are illustrated in Figure 1.

The $\vec{L_c}$ vector has three elements but only two degrees of freedom, so one element can be fixed to avoid the singularity. During the initialisation, the coordinate of the first keyframe is regarded as the world coordinate shown in Figure 1.

The camera will move along the pipe, looking along the pipe axis. Since the z axis of the world coordinate will not be perpendicular to the pipe axis, the pipe axis is likely to intersect the xy plane of the world coordinate. Thus the third element of vector $\vec{L_c}$ would not be zero and is set to a fixed nonzero number to avoid the singularity. The cylinder vector $\vec{L_c} = \left[a_c\ b_c\ 1\right]^{\mathrm{T}}$, where a_c and b_c are the elements of L_c along the x and y axes of the world coordinate, always intersects xy plane of the world coordinate in the point $\vec{O_c} = \left[x_c\ y_c\ 0\right]^{\mathrm{T}}$, at distances x_c and y_c in the x and y axes of the world coordinate. The system therefore uses five numbers to parameterise a cylinder, which is the minimum number of variables, where are estimated in unconstrained optimisation [18]. There are some other line representation methods, such as Plücker coordinates and orthonormal representation [13]. The former has a constraint to their parameters, and it is hard to illustrate the cylindrical regularity with both mathematically. The nonlinear geometry fitting is solved by g2o [1] which is used by ORB-SLAM2.

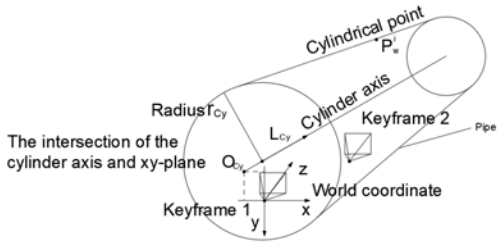

Fig. 1. Cylinder representation, with parameters describing a cylinder in space. Two image frames are shown; the world coordinates are set to those of the first frame.

Theoretically, all cylindrical points are equidistant from the cylinder axis. This property is used to estimate the cylinder and add constraints to the map points in later optimisation. For a cylindrical point $\vec{P_w^i}$, as the cross product between two vectors is the area of a parallelogram with the vectors as sides, the

following equation holds

$$S = |\vec{L_c} \times (\vec{P_w^i} - \vec{O_c})| = |\vec{L_c}| \cdot r_c \tag{1}$$

where $|\cdot|$ is the magnitude of a vector. According to this cylindrical property, an error function is obtained for a feature i as follows, where $\Delta x = x - x_c$ and $\Delta y = y - y_c$,

$$\begin{aligned} g(P_w^i) &= |\vec{L_c} \times (\vec{P_w^i} - \vec{O_c})|^2 - |\vec{L_c}|^2 |r_c|^2 \\ &= [b_c z - \Delta y]^2 + [\Delta x - a_c z]^2 + [a_c \Delta y - b_c \Delta x]^2 - {r_c}^2 (a_c^2 + b_c^2 + 1) \end{aligned} \tag{2}$$

This is minimized by the Gauss-Newton or Levenberg–Marquardt algorithm [11, 15], using the Jacobian $J(\vec{P_w^i})$ of $g(\vec{P_w^i})$ with respect to $\vec{\pi}$

$$J(\vec{P_w^i}) = \frac{\partial g}{\partial \vec{\pi}} = \begin{bmatrix} 2a_c z^2 - 2\Delta x z + 2a_c \Delta y^2 - 2b_c \Delta x \Delta y - 2a_c r_c \\ 2b_c z^2 - 2\Delta y z + 2b_c \Delta x^2 - 2a_c \Delta x \Delta y - 2b_c r_c \\ -2\Delta x + 2a_c z - 2{b_c}^2 \Delta x + 2a_c b_c \Delta y \\ -2\Delta y + 2b_c z - 2{a_c}^2 \Delta y + 2a_c b_c \Delta x \\ -2r_c({a_c}^2 + {b_c}^2 + 1) \end{bmatrix}^{\mathrm{T}} \tag{3}$$

4.2 Cylinder Detection

Cylinder detection is to estimate a cylinder among selected 3D points. Noise in the map points can lead to a poorly estimated cylinder, providing incorrect information through cylindrical regularity and causing misclassification in the future selection of cylindrical points. This sensitivity to noise makes cylinder estimation different from other landmarks. For accuracy, this cylinder is estimated from optimised map points instead of local map points observed by the current keyframe and its covisible keyframes. The local map points are noisy before optimisation. The local optimisation not only optimises the local map points but also filter out some points far outside the estimated cylinder as outliers at the end of the optimisation. Since that, the optimised map points are more reliable and accurate.

The system iteratively estimates the cylinder with updated cylindrical points close to the cylindrical surface. The points far from the cylindrical surface are regarded as non-cylindrical points. It is assumed that these distances of cylindrical points satisfy a Gaussian distribution, and the variance should be smaller than an appropriate threshold τ. This system select points whose distances are within 95% confidence intervals as cylindrical points. An alternative to this method could be to use an algorithm such as RANSAC [5], which could improve accuracy with higher computational cost. With a set of map points, the system detects the cylinder with **Algorithm 1**:

Algorithm 1 Cylinder Detection

1: Clean up the set G
2: Add all the map points to set G
3: Initialise the parameters
4: **while** $\sigma >$threshold σ_t **do**
5: Fit a cylinder to the map points from set G
6: Compute the distance of all map points from the cylinder axis
7: Compute the mean μ and variance σ of the distances $d_j (j \in G)$
8: Clean up the set G, and add all points close to the mean μ, given by $|d_j - \mu| \leq 1.96\sigma$, which is a 95% confidence interval
9: **end while**

After cylinder detection, the estimated cylinder is fixed during the subsequent local bundle adjustment until the low ratio of current cylindrical points to all local map points. The estimated cylinder changes slightly if the system optimises it in every local bundle adjustment due to the measurement noise. This instability will give the system a false impression that the robot moves in a curve pipe, which contradicts reality. With the estimated cylinder, the map points close to the cylindrical surface are classified as cylindrical points, which have a cylindrical regularity in the local bundle adjustment.

5 Bundle Adjustment with Cylindrical Regularity

Formulation The state of the proposed augmented ORB-SLAM2 system includes local frame poses, local map points, and the estimated cylinder parameters. When the kth keyframe is accepted, the full state set is defined as follows:

$$X = \{T_{wi}, \vec{p_j}, \vec{\pi}\}_{i \in \alpha_k, j \in \beta_k} \tag{4}$$

where the variable α_k denotes the covisible keyframes for the current keyframe k, the set β_k contains the map points observed by frames α_k, and $\vec{\pi}$ is the estimated cylinder. Covisible keyframes share more than a certain number of map points. The following cost function

$$f = \arg\min_X \sum_{i \in \alpha_k} \sum_{j \in \beta_k} \rho({e_v^{ij}}^2_{\Sigma_v}) + \sum_{l \in \gamma_k} \rho({|e_c^l|}^2_{\Sigma_c}) \tag{5}$$

is optimised via the local bundle adjustment method, with respect to X and with a fixed $\vec{\pi}$, using the g2o solver [1]. Here e_v^{ij} is the reprojection error of feature j observed by the keyframe i, and e_c^l is the cylindrical regularity of cylindrical point l from the set γ_k. The kernel function ρ is used to suppress outliers. Σ_v is the covariance matrix of a feature observation and is set to be an identity matrix; Σ_v is the variance of the point-to-surface constraint.

Reprojection Error The reprojection error is the distance between the projected map point and observed feature in the image plane, for frame i and feature j, this is given by

$$e_v^{ij} = z_{ij} - \kappa(T_{iw}P_{wj}) \tag{6}$$

where $\kappa(\cdot)$ is used to map a 3D map point P_{ij} to a 2D pixel coordinate f_{ij}, T_{iw} is the transformation matrix between the camera pose and the world coordinate.

Cylindrical Regularity The cylindrical regularity is same as equation (2). Given a cylindrical point P_j and the cylinder parameters L_c, o_c and r_c, the cylindrical regularity is given by

$$e_c^j = |L_c \times (P_w^i - o_c)|^2 - |L_c|^2 {r_c}^2 \tag{7}$$

In the following experiments, it is assumed that the distances of cylindrical points are within the interval $[-0.05r_c, 0.05r_c]$ from the cylindrical surface. The uncertainty of the regularity is set to $0.00065r_c^2$.

6 Performance Validation and Evaluation

The proposed algorithm is evaluated over different testing scenarios on synthetic and real data. ORB-SLAM2 and ORB-SLAM2 with cylindrical regularity (CRORB) are compared in terms of trajectory accuracy and running time on a computer with Intel Core i7-8700 @ 3.2GHz, 16GB memory.

6.1 Synthetic Data

A synthetic environment is shown in Figure 2. There is a straight 20-meter long pipe with an inner diameter of 1 meter. The lower part of the straight pipe is embedded in the ground. The virtual robot moves horizontally in the pipe, and the camera observes the inner pipe surface. To provide features for SLAM, a colourful image covers the inner pipe surface (Figure 2). Every second, the virtual camera collects 30 images of 1280×720 pixels, with added Gaussian noise. The true trajectory of the robot is exported from ROS Kinetic.

6.2 Real Data

The real data is acquired from a small unmanned ground vehicle (UGV) which moved along a long-buried straight pipe. The UGV (Figure 3) is equipped with a pin-hole camera collecting images of 720×576 at 20 FPS (Figure 3). A length of rope from a rope drum attaches to the UGV. When the robot moves, the rope drum turns and calculates the distance travelled shown on the images. This recorded travel distance is used as a reference instead of frame poses.

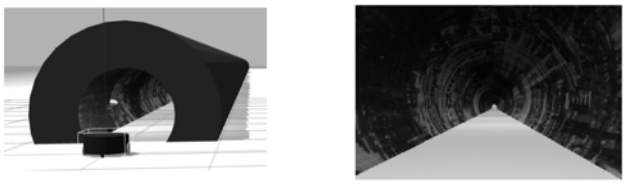

Fig. 2. Left: The synthetic pipe environment, with a flattened bottom surface to allow easy robot motion, and a cylindrical shape to represent the real pipe environment. The robot is pictured. Right: The pipe's inner surface with a synthetic texture to allow visual SLAM algorithms to detect features.

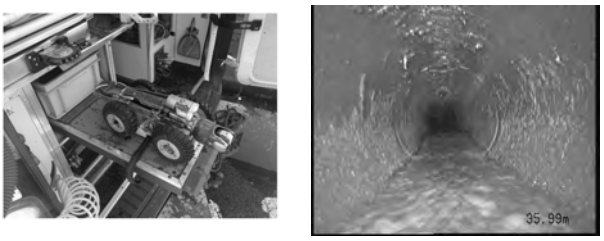

Fig. 3. Left: The UGV with camera used to collect data. Right: An example image from the pipe environment. The texture on the sides of the pipe can be seen.

Fig. 4. The 3D points from actual data and the cylinder fit to them. The *non-cylindrical* points can be seen which are not fit to the cylinder, and the *cylindrical* points which will be fit to the estimated cylinder are seen ahead of the most recent keyframe.

Dataset	Synthetic Data		Real Data	
Method	ORB-SLAM2	CRORB	ORB-SLAM2	CRORB
rmse	0.026m	0.007m	0.378m	0.285m
Cylinder Detection		0.0158s		0.00533s
BA	0.199s	0.324s	0.059s	0.073s
Local optimisation	0.411s	0.505s	0.08s	0.104s

Table 1. Comparison between the two SLAM algorithms

6.3 Discussion

The experiments proved that the CRORB is more accurate than the ORB-SLAM2 in a pipe at the cost of additional computation. The figure 4 show mapping with actual data, and the cylinder fit to the data points. The error between the ground truth and the estimated trajectory is difficult to visualise. Table 1 compared accuracy and computation between two SLAM algorithms. The root means squared error (RMSE) of the absolute pose error from two estimated trajectories are provided. These numbers indicate the performance of the estimation. Also, the running time of local optimisation, BA and cylinder detection are compared, each of which is the average time over ten runs on the same data. The local optimisation includes cylindrical points classification, optimiser initialisation and BA.

The running time of local optimisation and BA differs significantly between the two algorithms. The difference is because the BA in CRORB has additional cylindrical regularity. Also, the algorithms in actual data cost less computation. In the scale-free map, the robot's speed is faster in the actual data than in synthetic data. The local optimisation in synthetic data involves more covisible keyframes, and they share more common map points, which means a larger optimisation problem with more variables and more constraints. So the running time of synthetic data is longer. The conventional ORB-SLAM2 tends to underestimate the trajectory. The cylindrical regularity cannot solve this problem, but it can slow down this trend.

7 Conclusion

In this paper, a novel SLAM framework is proposed to leverage cylindrical regularity in a straight pipe. The cylindrical regularity can improve the localisation accuracy. In the future, we plan to use a new flexible cylinder representation method and include other structural information from a pipe network.

References

1. Burgard, R.: g2o - General Graph Optimization (2011), https://github.com/RainerKuemmerle/g2o
2. Chuang, T.Y., Sung, C.C.: Learning and SLAM based decision support platform for sewer inspection. Remote Sensing **12**(6) (2020). https://doi.org/10.3390/rs12060968
3. El Kahi, S., Asmar, D., Fakih, A., Nieto, J., Nebot, E.: A vison-based system for mapping the inside of a pipe. Proc. 2011 IEEE International Conference on Robotics and Biomimetics (1), 2605–2611 (2011). https://doi.org/10.1109/ROBIO.2011.6181697
4. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: Proc. European conference on computer vision. pp. 834–849. Springer (2014)

5. Fischler, M.A., Bolles, R.C.: Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM **24**(6), 381–395 (1981). https://doi.org/10.1145/358669.358692
6. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: Proc. IEEE International Conference on Robotics and Automation. pp. 15–22. No. May (2014). https://doi.org/10.1109/ICRA.2014.6906584
7. Gee, A.P., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering higher level structure in visual SLAM. IEEE Transactions on Robotics **24**(5), 980–990 (2008). https://doi.org/10.1109/TRO.2008.2004641
8. Hansen, P., Alismail, H., Rander, P., Browning, B.: Visual mapping for natural gas pipe inspection. International Journal of Robotics Research **34**(4-5), 532–538 (2015). https://doi.org/10.1177/0278364914550133
9. He, Y., Zhao, J., Guo, Y., He, W., Yuan, K.: PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features. Sensors (Switzerland) **18**(4), 1–25 (2018). https://doi.org/10.3390/s18041159
10. Kagami, S., Taira, H., Miyashita, N., Torii, A., Okutomi, M.: 3D Pipe Network Reconstruction Based on Structure from Motion with Incremental Conic Shape Detection and Cylindrical Constraint. In: Proc. IEEE International Symposium on Industrial Electronics. vol. 2020-June, pp. 1345–1352 (2020). https://doi.org/10.1109/ISIE45063.2020.9152377
11. Kenneth Levenberg: A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics **2**(2), 164–168 (1944)
12. Künzel, J., Werner, T., Eisert, P., Waschnewski, J., Möller, R., Hilpert, R.: Automatic Analysis of Sewer Pipes Based on Unrolled Monocular Fisheye Images. In: Proc. 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018. vol. 2018-Janua, pp. 2019–2027 (2018). https://doi.org/10.1109/WACV.2018.00223
13. Li, X., He, Y., Lin, J., Liu, X.: Leveraging Planar Regularities for Point Line Visual-Inertial Odometry (2020), http://arxiv.org/abs/2004.11969
14. Lopez-Escogido, D., De La Fraga, L.G.: Automatic extraction of geometric models from 3D point cloud datasets. In: Proc. 014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE (2014). https://doi.org/10.1109/ICEEE.2014.6978316
15. Marquardt, D.W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the society for Industrial and Applied Mathematics **11**(2), 431–441 (1963)
16. Mur-Artal, R., Tardos, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
17. Özyeşil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion. Acta Numerica **26**, 305–364 (2017). https://doi.org/10.1017/S096249291700006X
18. Shakarji, C.M.: Least-Squares Fitting Algorithms of the NIST Algorithm Testing System. Journal of Research of the National Institute of Standards and Technology **103**(6), 633–641 (1998). https://doi.org/10.6028/jres.103.043
19. Zhang, Y., Hartley, R., Mashford, J., Wang, L., Burn, S.: Pipeline reconstruction from fisheye images. Journal of WSCG **19**, 49–57 (2011)
20. Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., Yu, W.: StructSLAM: Visual SLAM with building structure lines. IEEE Transactions on Vehicular Technology **64**(4), 1364–1375 (2015). https://doi.org/10.1109/TVT.2015.2388780