Visual Simultaneous Localisation and Mapping for Sewer Pipe Networks Leveraging Cylindrical Regularity*

Rui Zhang[1], Rob Worley[1], Sarah Edwards[1], Jonathan Aitken[1], Sean R. Anderson[1] and Lyudmila Mihaylova[1]

***Abstract*— This work proposes a novel visual Simultaneous Localisation and Mapping (vSLAM) approach for robots in sewer pipe networks. One problem of vSLAM in pipes is that the scale drifts and accuracy degrades. We propose the use of structural information to mitigate this problem via cylindrical regularity. The main novelty consists of an approach for cylinder detection that is more robust than previous methods in non-smooth sewer pipe environments. Cylindrical regularity is then incorporated into both local bundle adjustment and pose graph optimisation by embedding the reprojection error. The approach adopts a minimal cylinder representation with only five parameters, avoiding constraints during the optimisation in vSLAM. A further novelty is that the estimated cylinder is part of the scale drift estimation, which enables a correction to the translation estimate and this further improves the accuracy. The approach, termed *Cylindrical Regularity ORB-SLAM* (CRORB), is benchmarked and compared to leading visual SLAM algorithms ORB-SLAM2 and direct sparse odometry (DSO), as well as a vSLAM algorithm with cylindrical regularity developed for gas pipes, using real sewer pipe data and synthetic data generated with the *Gazebo* modelling software. The results demonstrate that CRORB improves substantially over the competitors, with a reduction of approximately 70% in error on real data.**

I. Introduction

Sewer pipe inspection is essential for the necessary detection of blockages and leakages. Pipe defects can potentially lead to economic losses, environmental contamination, and health problems [1]. Inspections are commonly performed by inserting a manually operated, tethered probe or rover inside the pipe with a CCTV (closed-circuit television) system for observing faults. However, autonomous, untethered robots have a huge potential to perform long-term, persistent pipe health monitoring. Localisation is a key part of this process because faults must be accurately located to be repaired. To autonomously localise the robot in real-time, and therefore any damage detected, a simultaneous localisation and mapping (SLAM) algorithm is required [2], [3]. SLAM is a challenging task in buried pipes, mainly because of the unavailability of GPS (global positioning system) [4].

Visual SLAM (vSLAM) [5], is one appealing approach for SLAM in pipes because cameras are low-cost, provide detailed information about the environment, and are commonly found on pipe inspection robots [6]–[8]. However, vSLAM algorithms tend to deteriorate in pipe environments because the scale factor decreases along with the forward motion of the robot. The point cloud output of the vSLAM algorithm resembles a cone, with the diameter ($d_A \& d_B$) shrinking as the camera moves forward: the accuracy becomes worse the farther the robot moves. Figure 1 illustrates this problem.

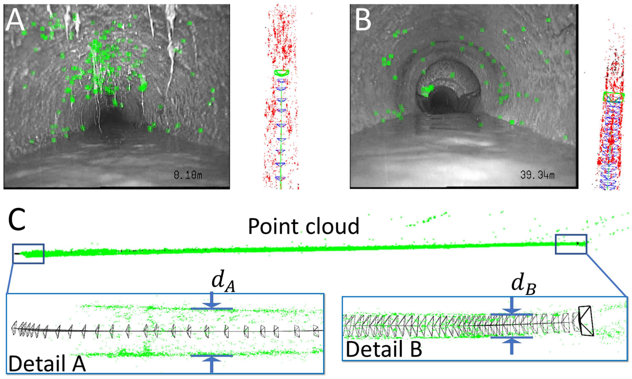

Fig. 1: Scale factor decrease over time using ORB-SLAM2 in a pipe. A,B: Image (left) and point cloud with frames (right); A: near the start of a trajectory, B: later in the trajectory. C: The overall point cloud, where the decrease in estimated pipe scale from point A to point B is seen, relative to the constant size of the illustrated frames.

In man-made environments, structural information can improve SLAM results [9]–[12]. Current research exploits structural information by building *Manhattan worlds* [13], [14], which assume that a man-made environment can be abstracted as a block with three dominant directions [15], which can be obtained in a single image. The shared Manhattan world can help estimate the frame poses. Another approach is to enforce structural regularities on the landmarks [11], [12], optimising the distance of landmarks from the expected value [11], [16]. Cylindrical structural information has been used in vSLAM for gas pipes [17] and in structure from motion (SFM) in sewer pipes [18]. However, gas pipes tend to be smooth and unobstructed, unlike sewer pipes, so cylinder estimation is simpler, and SFM is an offline problem, unlike vSLAM. Therefore, the research gap we address is using cylindrical regularity in vSLAM for sewer pipes.

One of the key challenges for vSLAM with cylindrical regularity is cylinder detection in a noisy point cloud. Shakarji et al. [19] and Lukacs et al. [20] fit a cylinder in a point cloud by minimising the point-cylinder distance, but these approaches are sensitive to noise and outliers [21],

*This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) PipeBots Programme Grant EP/S016813/1. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

[1] Authors are with Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, South Yorkshire, S10 2TN, UK rzhang47, l.s.mihaylova @sheffield.ac.uk

[22]. Lopez-Escogido et al. [23] used RANSAC (random sample consensus) and a direct method, but this approach may perform poorly when the uncertainty is unknown.

A two-step approach can be taken [21], [22], [24]–[26], first estimating the cylinder axis, and then estimating the cylinder's position and size using robust circle fitting on points projected to a plane perpendicular to the cylinder axis [22]. Estimation of the cylinder axis by principal component analysis (PCA) [21], [22], [26] is susceptible to outliers [22], and estimation of the axis by the property that vectors normal to the cylinder surface all point to the axis [24], [25] can suffer from poor quality normals. Overall, error can propagate from the first to the second step [27]. In this paper, we use a robust regression method, similar to [22], [28], to estimate the cylinder parameters, which leverages a new cylinder representation proposed here along with a new evaluation method to reject inaccurate estimations.

Cylinder representation is an important aspect of the problem for vSLAM because overparameterised approaches will overcomplicate optimisation. A cylinder is defined as a surface consisting of all the points on all the lines parallel to a given line, the cylinder axis. It can be described as a line in 3D space with a radius, and it has five degrees of freedom. A cylinder denoted by a unit vector, a point on the cylinder axis, and a radius [29] is overparameterised and will bring constraints to the optimisation. Taubin et al. [30] used a minimum number of parameters to denote a cylinder with a radius and four parameters converted to a transformation matrix but did not develop a method to perform the inverse transformation from the transformation matrix to the four cylinder parameters. Shakarji et al. [19] and Lukacs et al. [20] used different cylinder representation approaches, but both used a point closest to the origin to define the position of the cylinder axis. Shakarji et al. [19] did not introduce this point mathematically. And reducing the corresponding cost function of the latter can lead to a biased solution in optimisation. In all of these cases, it is challenging to implement cylindrical regularity in an optimisation problem. Therefore, this paper proposes the use of a cylinder representation that has a minimal set of parameters that is well-suited to online optimisation.

In summary, this paper proposes a new vSLAM algorithm that leverages cylindrical regularity termed CRORB. The approach is an extension of ORB-SLAM2 [31]. The main contributions of this paper include: 1) a new more robust cylinder detection algorithm that we combine with a minimal cylinder representation; 2) a vSLAM algorithm that leverages cylindrical information in bundle adjustment and pose graph optimisation; 3) an approach to recover the scale factor based on the isodiametric information.

The paper is organised as follows. Section II introduces the new cylinder representation approach and the cylinder evaluation approach. Section III defines the local bundle adjustment and pose graph optimisation with cylindrical regularity. Section IV gives the performance evaluation on both simulation and real-world data. The conclusions and plans for future work are given in Section VI.

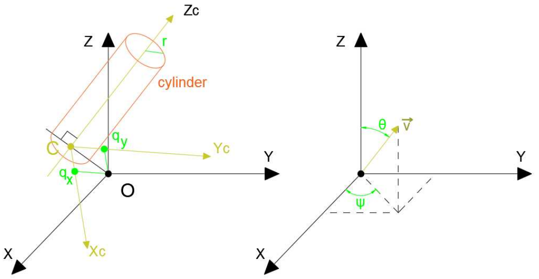

Fig. 2: A cylinder is shown in the left coordinate, and its direction is expressed in the right coordinate.

II. Cylinder representation and detection

A. Cylinder representation

This section describes an efficient cylinder representation that avoids constraints in optimisation by using a minimum number of parameters. Usually, as shown in Figure 2, a cylinder is parameterised by its radius, $r \in \mathbb{R}$ and its axis defined by a direction vector, $\vec{\mathbf{v}} \in \mathbb{R}^3$, and a position denoted by a point on the axis, $\mathbf{p} \in \mathbb{R}^3$. Therefore the set of cylinder parameters are initially defined as $\tilde{\mathcal{Y}} = \{\vec{\mathbf{v}}, \mathbf{p}, r\}$.

The cylinder direction $\vec{\mathbf{v}}$ can be defined by just two angles as shown in the right axis in Figure 2: the pitch and roll angles, $\theta \in \mathbb{R}$, $\psi \in \mathbb{R}$. The yaw angle of rotation is set to be equal to zero, since the cylinder is symmetric about this axis. The vector can be obtained as $\vec{\mathbf{v}} = \begin{bmatrix}\cos\psi\sin\theta & \sin\psi\sin\theta & \cos\theta\end{bmatrix}^\intercal$. Its rotation matrix $\mathbf{R}_{wa}$ can be obtained by transforming the Euler angle.

The key to the reduction in parameters is to choose the cylinder axis position parameter $\mathbf{p}$, defined in the world coordinate frame, as $\mathbf{p} = \mathbf{c}$, where $\mathbf{c}$ is the vector defining the closest point on the cylinder axis, C, to the origin of the world coordinate frame (see Figure 2). Thus the line segment OC is perpendicular to the cylinder axis with the perpendicular foot C. We create a cylinder coordinate a at point C with the z-axis in the same direction as the cylindrical axis. The origin O is on the xy-plane of the cylinder coordinate, and its position $\mathbf{o_a}$ in the cylinder coordinate is assumed to be $\mathbf{o}_a = \begin{bmatrix}q_x & q_y & 0\end{bmatrix}^\intercal$. We can define the position $\mathbf{c}$ of the point C in the world coordinate frame as $\mathbf{c} = -\mathbf{R}_{wa}\mathbf{o}_a$. Here $q_x \in \mathbb{R}$ and $q_y \in \mathbb{R}$ denote the transformed origin position coordinates of the world coordinates into cylinder coordinates. Setting the z-axis point in $\mathbf{o}_a$ to zero results in the parameter reduction.

Hence the model incorporating minimal cylinder information is defined by the following parameters

$$\mathcal{Y} = \{\theta, \psi, q_x, q_y, r\}. \tag{1}$$

This set of cylinder parameters consists in the ones used in the visual SLAM algorithm, presented in the later sections.

The direct use of Euler angles could lead to singularities in principle. When θ is equal to 0, it incurs a singularity. It is not possible to estimate ψ from the rotation matrix of the cylinder, and its value has no effect on the direction of cylinder $\vec{\mathbf{v}}$. Therefore, we set ψ to be zero directly and update

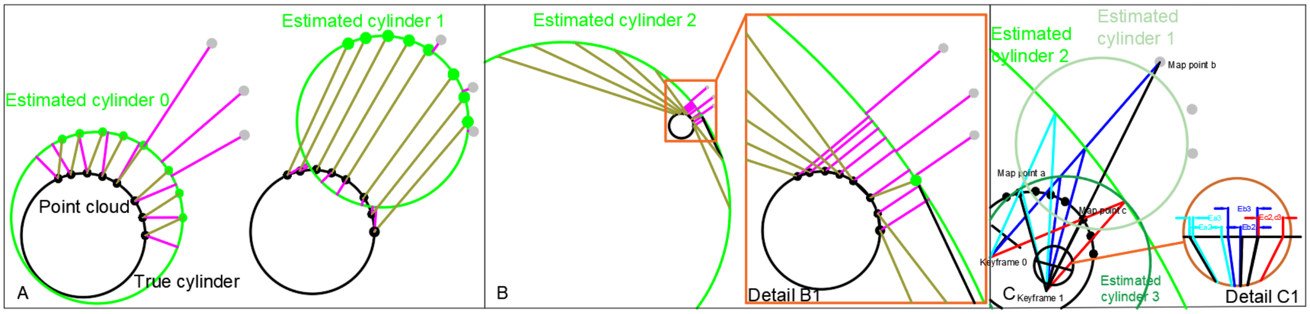

Fig. 3: The challenges in cylinder fitting and the new evaluation method. In each panel the black circle denotes the cross-section of the true pipe, and the green circle denotes an estimated cylinder. The black and grey points are the inliers and outliers respectively. The purple lines are the point-cylinder distances, and the dark yellow lines are the ideal error distances, which are the distance between map points and their corresponding positions (green points on estimated cylinders) on the estimated cylinder. A: The disadvantages of point-cylinder distance. B: The consequence of the threshold changing according to the estimated radius. C: The reprojection errors (Ea2, Ea3, etc.) of three map points given estimated cylinders.

q_x and q_y. Alternative methods for preventing singularities exist and these include quaternions [32], however, this would incur an additional parameter in the representation.

B. Cylinder detection and evaluation

This section introduces a cylinder detection approach and proposes a new cylinder evaluation approach in a sparse point cloud. We use a robust optimisation method similar to our previous work [28] to estimate the cylinder directly with the new cylinder representation approach. This approach estimates the cylinder parameters $\mathcal{Y} = \{\theta, \psi, q_x, q_y, r\}$ by minimising the L2-norm of the point-cylinder distances, $\mathbf{e}_c^n$, wrt the cylinder parameters, $\mathcal{Y}$, where the cost function is

$$F(\mathcal{Y}) = \sum_{n \in \mathcal{Q}} \|e_c^n\|^2 \tag{2}$$

where map points $\mathbf{p}^n$ belongs to the set of inliers $\mathcal{Q}$, and the point cylinder distances are defined by

$$e_c^n = |\vec{\mathbf{v}} \times (\mathbf{p}^n - \mathbf{c})| - r \tag{3}$$

The system alternately estimates the cylinder parameters $\mathcal{Y}$ and selects points from the point cloud which are inliers with respect to the estimated cylinder. This optimisation is stopped when the variance, $\sigma_e^2 = \frac{1}{N} F(\mathcal{Y})$, of point-cylinder distance is less than a threshold κ_1 or the maximum iteration is reached. The system selects a point for the next iteration if its point-cylinder distance is less than threshold κ_2. This work uses optimised points observed by at least four keyframes to exclude outliers in the point cloud. The optimised map points refer to those optimised in the previous local optimisation and excluded in the current local optimisation. The cost function is multimodal, so a good initial value is necessary, which is found using PCA on the recent trajectory.

There are a number of problems using point-cylinder distance as the cost function, which are illustrated in Figure 3. One problem is the imperfect evaluation of the estimated cylinder: in Figure 3A, Cylinder 0 fits better than Cylinder 1 in terms of position, however, the error in terms of point-cylinder distance in Cylinder 1 is lower. The ideal error cannot be computed in practice. A further problem is finding an appropriate threshold κ_2 for rejecting outliers in RANSAC or optimisation methods [23], [26]. Using the estimated cylinder radius multiplied by a constant as the threshold [26] can cause problems when the estimated radius is far from the real radius shown by Cylinder 2 in Figure 3B, where outliers are accepted as inliers since their point-cylinder distances are small compared with the large radius.

Algorithm 1 Cylinder detection

```
Input: Optimised point cloud
Output: Cylinder parameters $\mathcal{Y} = \{\theta, \psi, q_x, q_y, r\}$
Initialise cylinder parameters $\mathcal{Y}$ with PCA
Add all the map points to set $\mathcal{Q}$ of inliers
while Error variance $\sigma_e^2 < \kappa_1$ do
    Optimise cylinder parameters $\mathcal{Y}$ using (2)
    Set $\mathcal{Q}$ to the empty set
    Add map points with error $|e_c^n| < \kappa_2$ to $\mathcal{Q}$
end while
Evaluate reprojection error and count inliers
Define ratio of inliers to outliers $\alpha$
if $\alpha > \kappa_3$ then
    Return $\mathcal{Y}$
end if
```

To solve these problems, we use reprojection error to evaluate the solutions instead of point-cylinder distance, as illustrated in Figure 3C. Cylindrical points should be close to the cylindrical surface, and their positions can be approximated by computing the intersection (cyan points) of their backprojected rays (blue lines) from a selected keyframe and the estimated cylindrical surface. The algorithm projects the intersection to the rest of the keyframes and computes the reprojection error, which is the distance between projected intersections and corresponding features. Our proposed algorithm is able to detect keyframes that are outside of the estimated cylinder, as in Cylinder 1 in Figure 3A, since back-projected rays have zero or two intersections. Our method transforms 3D distance to pixel distance, which enables us to use known feature uncertainty as a threshold. The uncertainty is independent to the size of estimated cylinders. The algorithm then evaluates a cylinder by checking if the ratio α of inliers to outliers is greater than a threshold κ_3. The full procedure is given in Algorithm 1.

III. Leveraging cylindrical information

This section describes the use of cylindrical information in triangulation, local optimisation and pose-graph optimisation (PGO). The estimation can benefit from cylindrical information indirectly through the cylindrical points.

A. Triangulation

With the cylindrical information, the system performs triangulation and verifies the cylindrical assumptions in new map points created in triangulation. During the triangulation, the system assumes new map points belong to the previous cylinder. The system computes the reprojection error introduced in Section II and triangulates map points whose reprojection errors exceed a threshold. If fewer than 65% of points are cylindrical points, the system drops the cylindrical assumption and triangulates all cylindrical points. Otherwise, the system accepts positions of intersections as the positions of cylindrical points and marks the current keyframe as a cylindrical keyframe. This approach completes the triangulation and cylindrical point selection together. It prevents the cylindrical points from shrinking towards the cylinder axis as well.

B. Local bundle adjustment with cylindrical regularity

This section introduces local bundle adjustment (BA) leveraging the cylindrical regularities.

1) Formulation: The local BA includes map points $\mathcal{P}_k$ and keyframes $\mathcal{L}_k$ according to the covisibility graph [33]. The optimisation also involves cylinders $\mathcal{I}_k$ whose cylindrical points join this optimisation. The full state vector contains the current keyframe k, its covisible keyframes $\mathcal{V}_k$, map points $\mathcal{P}_k$ and cylinders $\mathcal{I}_k$. The states at time t are

$$\mathcal{X}_t = \{\mathbf{T}_{iw}, \mathbf{p}^j, \mathcal{Y}^m\}_{i\in\mathcal{V}_k, j\in\mathcal{P}_k, m\in\mathcal{I}_k} \tag{4}$$

where $\mathbf{T}_{iw} \in \mathrm{SE}(3)$ denotes the frame poses. The cost function is

$$\begin{aligned} J(\mathcal{X}_t) = &\sum_{j\in\mathcal{L}_k}\sum_{i\in\mathcal{P}_k} \rho\left(\|\mathbf{e}_v^{ij}\|^2_{\boldsymbol{\Sigma}_v}\right) \\ &+ \sum_{m\in\mathcal{I}_k}\sum_{n\in\mathcal{Q}_m\cap\mathcal{P}_k} \rho\left(\|e_c^{mn}\|^2_{\boldsymbol{\Sigma}_c}\right) \end{aligned} \tag{5}$$

where $\mathbf{e}_v^{ij} \in \mathbb{R}^2$ is the reprojection error of feature j observed by keyframe i, and $e_c^{mn} \in \mathbb{R}$ is the point-cylinder distance of cylindrical point n on cylinder m. $\mathcal{Q}_m$ is a set of the corresponding cylindrical points on cylinder m. The Huber kernel function ρ is used to improve robustness to outliers. $\boldsymbol{\Sigma}_v$ is the covariance matrix of a feature observation, and $\boldsymbol{\Sigma}_c$ is the variance of the point-to-surface constraint. We model the point-cylinder distance with Gaussian distribution. In the following experiments, it is assumed that the distances of cylindrical points are within the interval $[-0.05r, 0.05r]$ from the cylindrical surface. The uncertainty of the regularity $\boldsymbol{\Sigma}_c$ is set to $\frac{25r^2}{38416}$. Also, a Chi-squared test is used to reject outliers, and the threshold is 3.814.

2) Reprojection error: The reprojection error is the distance between the projected map point and the observed feature in the image frame.

3) Cylindrical regularity: Given a cylindrical point $\mathbf{p}^n$ and the cylinder parameters $\{\mathbf{c}_m, \vec{\mathbf{v}}_m, r_m\}$, the cylindrical regularity term e_c^{mn} is given by (3). The next subsection III-C introduces pose graph optimisation leveraging a new regularity.

C. Pose graph optimisation with cylindrical regularity

This part introduces a new cylindrical regularity in pose graph optimisation according to the structure of pipes. Pose graph optimisation is used to correct keyframe poses in the detected loop. It enforces a Sim(3) constraint between two keyframes, mainly according to the covisibility graph. Thus, the keyframes that share few features with others have few constraints. They may change more than expected after the loop closing. We build Sim(3) constraints between estimated cylinders and the corresponding cylindrical keyframes, which can help maintain the relative poses. It can prevent cylindrical keyframes from moving out of the corresponding pipe during optimisation.

1) Formulation: The full state vector in the pose graph optimisation contains all keyframe poses $\mathcal{F}$ and cylinders $\mathcal{C}$. The set of states is defined as:

$$\mathcal{X} = \{\mathbf{S}_{wi}, \mathbf{S}_{wm}\}_{i\in\mathcal{F}, m\in\mathcal{C}}, \tag{6}$$

where $\mathbf{S}_{wi} \in \mathrm{Sim}(3)$ is the similarity transformation matrix of keyframe i, which has 7 degrees of freedom. Cylinders m join the pose graph optimisation in the form of a similarity transformation matrix $\mathbf{S}_{mw} \in \mathrm{Sim}(3)$ also. The transformation matrix of the cylinder can be obtained by converting the cylinder parameters, described in Section II. The scale factor is set to 1.

The system minimises the following cost function with respect to all parameters $\mathcal{X}$

$$J(\mathcal{X}) = \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{G}_i} \|\boldsymbol{\epsilon}_p^{ij}\|^2_{\boldsymbol{\Sigma}_p} + \sum_{m\in\mathcal{C}}\sum_{n\in\mathcal{H}_m} \|\boldsymbol{\epsilon}_c^{mn}\|^2_{\boldsymbol{\Sigma}_p}, \tag{7}$$

where $\boldsymbol{\epsilon}_p$ is the relative pose error between keyframe i and related keyframes in set $\mathcal{G}_i$ [33]. Cylindrical relative pose error $\boldsymbol{\epsilon}_c \in \mathfrak{sim}(3)$ is given between cylinder m and its cylindrical keyframes in set $\mathcal{H}_m$. $\boldsymbol{\Sigma}_\mathbf{p}$ is assumed to be an identity matrix in this case.

2) Cylindrical relative pose error: Cylindrical relative pose error is the difference between the original relative pose $\mathbf{S}'_{im} \in \mathrm{Sim}(3)$ and the updated relative pose between the cylinder and the corresponding cylindrical keyframes. It is defined as follows:

$$\epsilon_c^{mn} = \log_{Sim(3)}(\mathbf{S}_{mw}\mathbf{S}_{wi}\mathbf{S}'_{im}), \tag{8}$$

After pose-graph optimisation, the system optimises all related keyframes, map points and cylinders in global bundle adjustment.

D. Isodiametric information

One problem encountered with monocular SLAM is that the scale drifts over time. This is particularly a problem in pipes, as has been observed here empirically and in the literature [18]. We can use the cylinder model to mitigate this scale drift problem. In sewer pipe networks, pipes tend to have a constant diameter in between manholes [34]. If we make the assumption that the pipe diameter is constant between manholes, we can estimate the scale from the estimated cylinders and use it to correct the estimated translations.

To correct for scale drift, we assume the first cylinder detected in the CRORB algorithm is unaffected by scale drift, and its radius r_r is accepted as the reference. The scale factor of a cylindrical keyframe is equal to the ratio of the radius r_i of its observed cylinder to the reference radius, and therefore the corrected translation estimate $\mathbf{t}^*_{ij}$ is given by

$$\mathbf{t}^*_{ij} = \mathbf{t}_{ij} \frac{r_r}{r_i}, \tag{9}$$

where $\mathbf{t}_{ij}$ is the uncorrected translation estimate between keyframe i and the next keyframe j. For non-cylindrical keyframes, the scale can be obtained by interpolation. We refer to this version of the algorithm as CRORB with isodiametric correction (CRORBic).

IV. Experiments

To evaluate the proposed algorithms CRORB and CRORBic, we conducted experiments on:

1) Synthetic data from a 3D pipe network simulation modelled on a real sewer network constructed using robot operating system [35] and Gazebo [36].
2) Real-world data, from live CCTV inspection data of a sewer pipe.

To demonstrate the advantages of CRORB/CRORBic we benchmarked and compared it to ORB-SLAM2, DSO [37], and a *Benchmark* vSLAM method with cylindrical regularity where cylinder detection using point-cylinder distance as used in [17] is integrated with ORB-SLAM2 implementation. In these experiments, the algorithms were implemented on a computer with Intel Core i7-8700 @ 3.2GHz, 32GB memory.

The algorithms are evaluated and compared in terms of trajectory accuracy and running time. The open-source software `evo_traj` was used to align the estimated trajectories with ground truth [38]. The absolute pose error (APE) is the difference between the estimated poses and the ground truth. The main metric used was the root mean square error (RMSE) of the APE, $e_{RMSE} = \sqrt{\sum_{i=1}^{n} e_i^2/n}$, where e_i is the APE for pose i and there are n estimated poses in total. The scale is estimated as the ratio of the true travel distance to the estimated distance between the first two keyframes created in the initialisation.

A. Synthetic data

A synthetic 3D pipe network environment based on a real-world sewer pipe network in the UK (Figure 4A) was created in Gazebo [36] (Figure 4B) and simulated with ROS [35]. Manholes connect several straight pipe sections with a constant 1 metre diameter to form a branched pipe network similar in layout to the real sewer pipe network. A virtual robot moves through the pipes and the front-facing camera observes the inner pipe surface. The inner pipe surface had colour variation emulating the inside of a sewer pipe. The virtual camera collects images of 800×800 pixels at 30 frames per second, with added Gaussian noise.

We tested the vSLAM algorithms for three different trajectories. Table I shows the results of different algorithms on synthetic data. The last column shows the results obtained from algorithms without and with loop closing separately. The corresponding trajectories are shown in Figure 4C-F. The orientations of the frame poses are similar to the true value, and the errors are in translation which mainly stems from the drift of the scale factor. The cylindrical regularity slows down the decrease of the scale factor, and the trajectories from CRORB are all more accurate than those from other algorithms. With the isodiametric information (CRORBic), the algorithm can recover the scale factor even if the system detected cylinders only in some keyframes. The results show that cylindrical regularities improve estimation at a small computation cost. Compared with ORB-SLAM2, Benchmark has a more significant error due to poor cylinder fitting among noisy point clouds (Figure 4 and Table I). DSO was also tested on the synthetic data. The results show that DSO has substantially worse accuracy than both ORB-SLAM2 and our own proposed methods since the inaccurate initialisation (Figure 4 and Table I). The scale factor decreases slightly after the initial response.

Note that the RMSE across the data sets does not correlate with distance travelled as might be expected - this is because, although errors do accumulate, the *rate* of error accumulation tends to be different in separate experiments due to a number of other factors, such as lighting conditions, texture variability and robot motion - a systematic study of these effects is beyond the scope of this paper.

B. Real data

Real data was acquired from a CCTV rover inspection of two straight sewer pipes, 40 and 50 metres long respectively, connected by a manhole (Figure 5A): the inspection of the first pipe segment is referred to here as Expt 1 and the second pipe segment as Expt 2. The robot (Figure 5B) was equipped with a forward pinhole camera collecting images of 720×576 resolution at 25 FPS (Figure 5C). The robot moved slowly at an average rate of 0.18 m/s down the centre of the pipe. In Expt 2, due to disturbances (camera shake and severe flowing water), ORB-SLAM2 only works on the final 20 metres.

To establish ground truth for localisation, we used a tether attached to the robot to measure this one-dimensional distance. Note that ground truth for the full 6DOF pose is not available in the real sewer pipe. The metric evaluated was the root mean square of the error of travel distance. The camera intrinsic parameters were obtained by calibration with a checkerboard. We used the default parameters from ORB-SLAM2, except the number of features per image which was set to 3,000 [39].

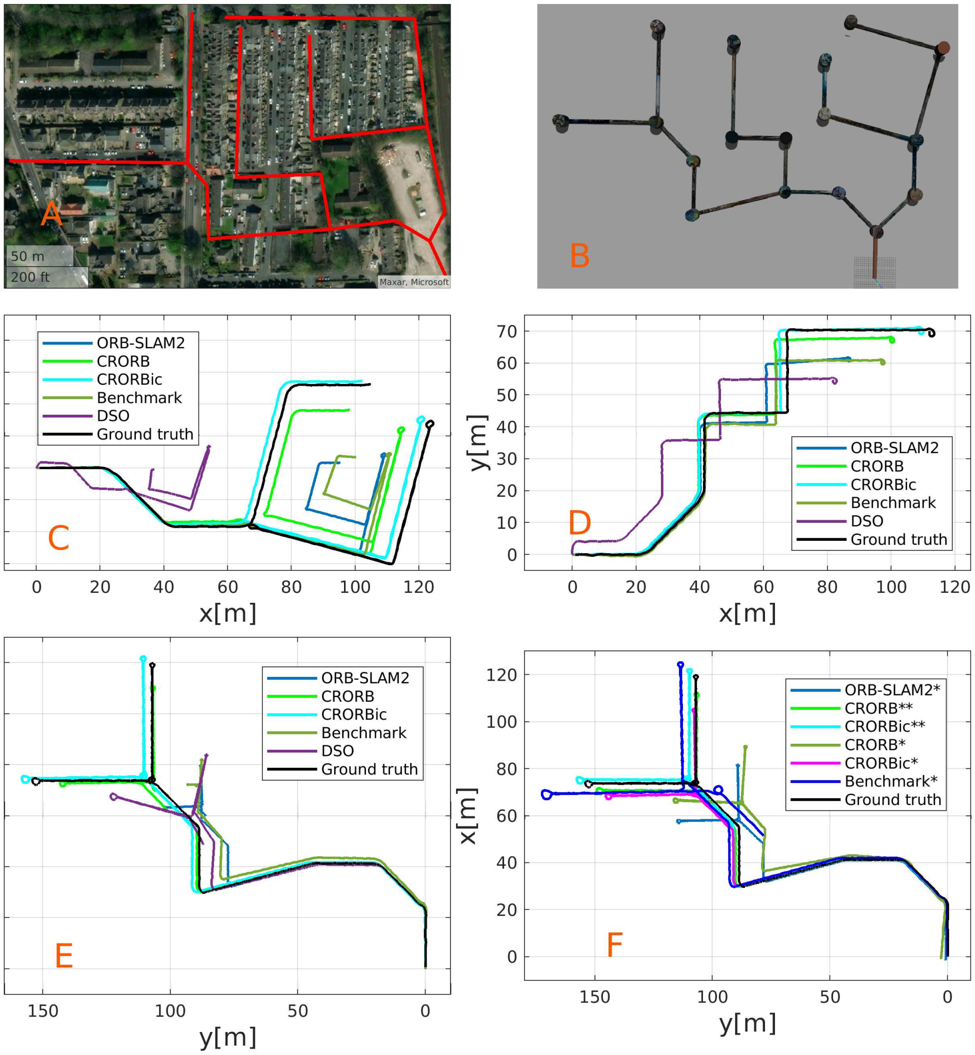

Fig. 4: Simulation results. A: Real-world sewer pipe network. B: Synthetic pipe network created in ROS-Gazebo to emulate the real network in panel A. C-F: Estimated trajectories around the simulated pipe network from the ORB-SLAM2, CRORB and CRORBic algorithms against the ground-truth trajectory. Superscript* means the algorithm runs loop closing without cylindrical regularity. Superscript** means the algorithm runs loop closing leveraging the cylindrical regularity.

Dataset	Synthetic Data 1		Synthetic Data 2		Synthetic Data 3				
Travel Distance	333.6		175.5		395.6				
Method	ORB2	CRORB(ic)	ORB2	CRORB(ic)	ORB2	CRORB(ic)	ORB2*	CRORB(ic)*	CRORB(ic)**
RMSE(m)	13.79	7.84(3.29)	12.01	6.82(2.84)	26.70	7.59(4.34)	24.92	6.21(7.78)	5.63(4.66)
Triangulation(ms)	57.6	56.2	48.2	47.0	53.0	52.1	54.0	52.4	54.8
Local optimisation(ms)	150.3	169.4	132.9	153.7	170.5	194.4	166.8	190.8	205.1
Cylinder Detection(ms)	—	31.0	—	28.5	—	35.7	—	39.6	45.9
Method	DSO	Benchmark	DSO	Benchmark	DSO	Benchmark	Benchmark*		
RMSE(m)	43.7	23.88	27.5	25.54	43.5	51.96	47.43		
Local optimisation(ms)	—	145.8	—	133.8	—	167.9	174.3		

TABLE I: Comparison between the SLAM algorithms on synthetic data. Note that ORB2 is an abbreviation for ORB-SLAM2. Superscript* means the algorithm runs loop closing without cylindrical regularity. Superscript** means the algorithm runs loop closing leveraging the cylindrical regularity. Symbol '—' means the statistics are not available or not applicable.

The results demonstrate the effectiveness of the proposed approach in estimating the robot's travel distance. Table II shows that CRORB obtains better accuracy than ORB-SLAM2, reducing the RMSE by 48% with experimental data set 1 and an 18% reduction in RMSE in data set 2. CRORB with isodiametric correction (CRORBic) can use intermittent estimated cylinders to correct the scale drift, and Table II shows that it works much more accurately, giving an 86% reduction in RMSE compared to ORB-SLAM2 in data set 2. Performance results for all algorithms are compared in

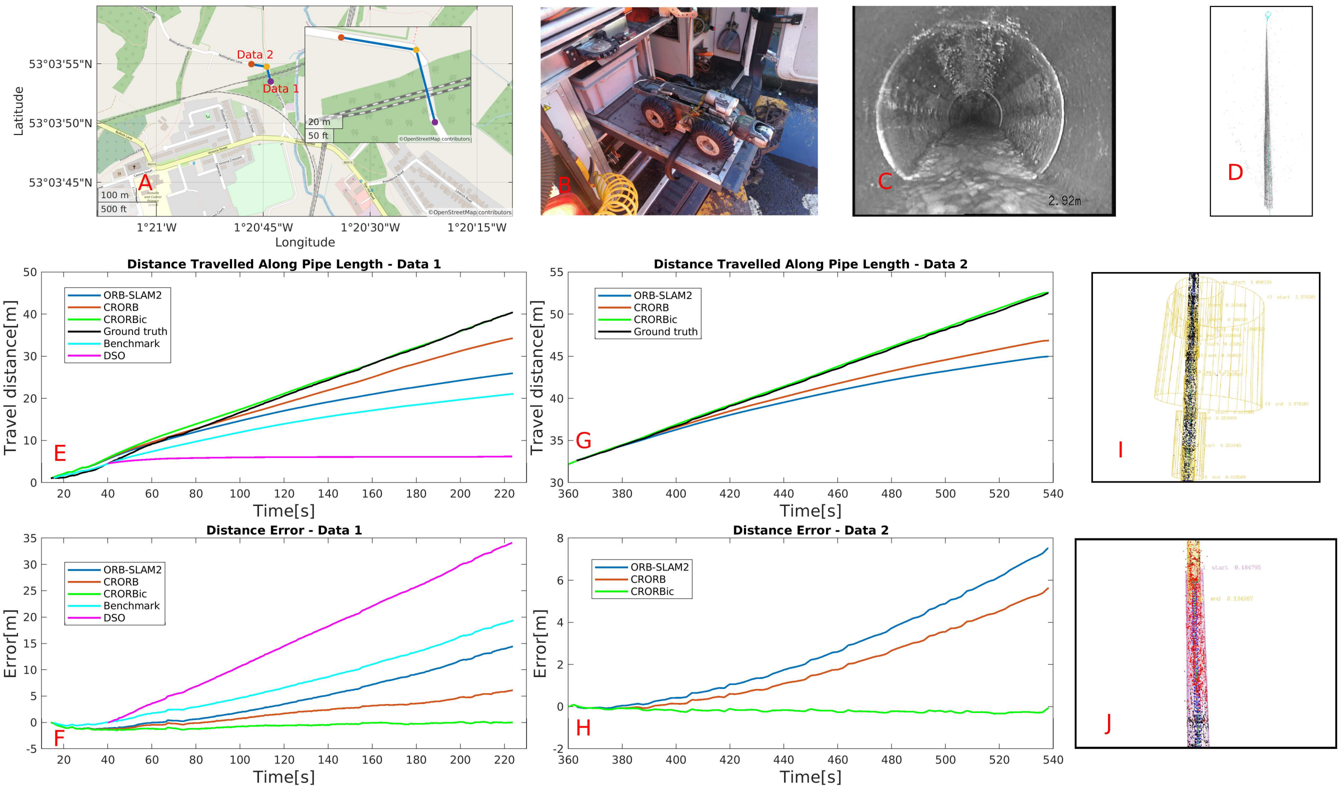

Fig. 5: Real-world experimental results. A: Aerial view of the manholes (dots) and two sewer pipe locations (blue lines) corresponding to Experiment 1 and Experiment 2. B: Sewer pipe inspection rover. C: View inside the sewer pipe from the rover camera. D: The top view of the point cloud created by DSO on Data 1. E&G: Comparison of estimated travel distance and ground truth over time for pipes 1 and 2 respectively. F&H: Travel distance error over time for pipes 1 and 2 respectively. I: Cylinder detection using the Benchmark method. J: Cylinder detection using our proposed method in Algorithm 1.

Dataset	Experimental Data 1				Experimental Data 2		
Travel distance (m)	39.5				20.3		
Approach	ORB-SLAM2	CRORB(-ic)	Benchmark	DSO	ORB-SLAM2	CRORB(-ic)	Benchmark
RMSE (m)	6.41	3.30(1.62)	10.29	18.55	3.33	2.71(0.44)	5.04
Triangulation (ms)	68.6	60.5	—	—	64.9	54.0	—
Local BA (ms)	76.5	98.4	82.2	—	75.8	83.3	76.5

TABLE II: Experimental results. Symbol '—' means the statistics are not available or not applicable.

Figure 5E-H and the problems with cylinder detection from [17] compared to our method is illustrated in Figure 5I-J. Point cloud in Figure 5D is conical, which means that the scale factor shrinks rapidly on Data 1. The execution time for triangulation, local optimisation, and cylinder detection is shown in Table II averaged over ten runs.

V. Discussion

This paper develops a vSLAM algorithm that leverages structural information regarding cylindrical regularity and solves the problem of scale drift and gives accurate mapping and localisation. The results show substantial improvements over ORB-SLAM2 in both simulations and real data.

There is still scope for future work. In certain regions of sewer pipes, there may be insufficient features for the vSLAM algorithm to match between keyframes, which could cause the algorithm to fail. In practice, multisensor SLAM is needed to deal with occasions with insufficient visual features. There are times during operation when there are significant amounts of flowing water in the pipe, which will cover the lower part of the pipe and disrupt the vSLAM algorithm. Future work in path planning algorithms will consider flow conditions and schedule the robot to navigate through the pipe at periods of low flow and rest or recharge during periods of high flow. Analysis of the impact of different uncertainties on the algorithm performance in changeable lighting environments and variable pipe textures is another area of future work.

VI. Conclusion

This paper proposed a new approach for vSLAM in sewer pipes, leveraging structural information in the form of cylindrical regularity, including a minimal cylinder representation suited to online optimisation, a robust approach for cylinder evaluation and associated approaches for local bundle adjustment and pose graph optimisation. The experimental results from both 3D simulations and real-world data demonstrated the effectiveness of the approach versus a standard vSLAM approach, ORB-SLAM2. The comparison showed that the cylindrical information not only improved the accuracy of the trajectory but also helped recognise the pipes.

REFERENCES

[1] American Society of Civil Engineers, "Failure to Act: The Economic Impact of Current Investment Trends in Water and Wastewater Treatment Infrastructure," 2011, [Online] Available at: http://www.asce.org/uploadedfiles/issues_and_advocacy/our_initiatives/infrastructure/content_pieces/failure-to-act-waterwastewater-report.pdf (accessed 29/09/2020).

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[4] J. M. Aitken, M. H. Evans, R. Worley, S. Edwards, R. Zhang, T. Dodd, L. Mihaylova, and S. R. Anderson, "Simultaneous Localization and Mapping for Inspection Robots in Water and Sewer Pipe Networks: A Review," *IEEE Access*, vol. 9, pp. 140 173–140 198, 2021.

[5] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual SLAM algorithms," *Robotics*, vol. 11, no. 1, p. 24, 2022.

[6] P. Debenest, M. Guarnieri, and S. Hirose, "PipeTron series - Robots for pipe inspection," in *Proceedings of the 3rd International Conference on Applied Robotics for the Power Industry (CARPI)*, 2014, pp. 1–6.

[7] A. A. F. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, "A new approach to the sewer pipe inspection: Fully autonomous mobile robot "KANTARO"," in *Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics (IECON)*. IEEE, 2006, pp. 4088–4093.

[8] E. Rome, J. Hertzberg, F. Kirchner, U. Licht, and T. Christaller, "Towards autonomous sewer robots: the MAKRO project," *Urban Water*, vol. 1, no. 1, pp. 57–70, 1999.

[9] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-Inertial Odometry with Structural Regularity of Man-Made Environments," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 999–1013, 2019.

[10] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6583–6590, 2020.

[11] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging Planar Regularities for Point Line Visual-Inertial Odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. [Online]. Available: http://arxiv.org/abs/2004.11969

[12] X. Li, Y. Li, E. P. Ornek, J. Lin, and F. Tombari, "Co-Planar Parametrization for Stereo-SLAM and Visual-Inertial Odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6972–6979, 2020.

[13] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," in *Proc. IEEE International Conference on Robotics and Automation*, 2014, pp. 1540–1545.

[14] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.

[15] J. M. Coughlan and A. L. Yuille, "Manhattan World: Compass direction from a single image by Bayesian inference," *Proceedings of the IEEE International Conf. on Computer Vision*, vol. 2, pp. 941–947, 1999.

[16] Y. Lu and D. Song, "Visual Navigation Using Heterogeneous Landmarks and Unsupervised Geometric Constraints," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 736–749, 2015.

[17] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Visual mapping for natural gas pipe inspection," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 532–538, 2015.

[18] S. Kagami, H. Taira, N. Miyashita, A. Torii, and M. Okutomi, "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, 2020, pp. 1345–1352.

[19] C. M. Shakarji, "Least-Squares Fitting Algorithms of the NIST Algorithm Testing System," *Journal of Research of the National Institute of Standards and Technology*, vol. 103, no. 6, pp. 633–641, 1998.

[20] G. Lukács, R. Martin, and D. Marshall, "Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1406, no. June, pp. 671–686, 1998.

[21] J.-F. Lalonde, N. Vandapel, and M. Hebert, "Automatic Three-Dimensional Point Cloud Processing for Forest Inventory Automatic Three-Dimensional Point Cloud Processing for Forest Inventory," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-06-21, 2006.

[22] A. Nurunnabi, Y. Sadahiro, R. Lindenbergh, and D. Belton, "Robust cylinder fitting in laser scanning point cloud data," *Measurement*, vol. 138, pp. 632–651, 2019. [Online]. Available: https://doi.org/10.1016/j.measurement.2019.01.095

[23] D. Lopez-Escogido and L. G. De La Fraga, "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.

[24] T. Rabbani and F. van den Heuvel, "Efficient Hough transform for automatic detection of cylinders in point clouds," pp. 60–65, 2005.

[25] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[26] T. T. Tran, V. T. Cao, and D. Laurendeau, "Extraction of cylinders and estimation of their parameters from point clouds," *Computers and Graphics (Pergamon)*, vol. 46, pp. 345–357, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cag.2014.09.027

[27] A. M. Araújo and M. M. Oliveira, "Connectivity-based cylinder detection in unorganized point clouds," *Pattern Recognition*, vol. 100, 2020.

[28] R. Zhang, M. H. Evans, R. Worley, S. R. Anderson, and L. Mihaylova, "Improving SLAM in Pipe Networks by Leveraging Cylindrical Regularity," in *Proc. Towards Autonomous Robotic Systems Conference (TAROS)*, 2021, pp. 56–65.

[29] T. Chaperon and F. Goulette, "Extracting cylinders in full 3D data using a random sampling method and the Gaussian image," *Vmv 2001*, pp. 35–42, 2001.

[30] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[31] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[32] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, Eds., *Applications of quaternions in robotics*. London: Springer London, 1998, pp. 99–111. [Online]. Available: https://doi.org/10.1007/BFb0047690

[33] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[34] D. Butler, C. James Digman, C. Makropoulos, and J. Davies, *Urban Drainage*, 4th ed. Boca Raton: CRC Press, 2018.

[35] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[36] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2149–2154.

[37] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[38] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

[39] R. Mur-Artal, J. D. Tardos, J. M. M. Montiel, and D. Galvez-Lopez, "ORB-SLAM2," 2016. [Online]. Available: https://github.com/raulmur/ORB-SLAM2