



## OPEN ACCESS

## EDITED BY

Dongbing Gu,  
University of Essex, United Kingdom

## REVIEWED BY

Saber Kazeminasab Hashemabad,  
Harvard Medical School, United States  
Hui Zhou,  
Nanjing University of Science and  
Technology, China

## \*CORRESPONDENCE

Xiangyu S. Li,  
✉ sc16x2l@leeds.ac.uk  
Netta Cohen,  
✉ n.cohen@leeds.ac.uk

RECEIVED 08 April 2023

ACCEPTED 06 November 2023

PUBLISHED 23 November 2023

## CITATION

Li XS, Nguyen TL, Cohn AG, Dogar M and  
Cohen N (2023), Real-time robot  
topological localization and mapping  
with limited visual sampling in simulated  
buried pipe networks.  
*Front. Robot. AI* 10:1202568.  
doi: 10.3389/frobt.2023.1202568

## COPYRIGHT

© 2023 Li, Nguyen, Cohn, Dogar and  
Cohen. This is an open-access article  
distributed under the terms of the  
[Creative Commons Attribution License  
\(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or  
reproduction in other forums is  
permitted, provided the original author(s)  
and the copyright owner(s) are credited  
and that the original publication in this  
journal is cited, in accordance with  
accepted academic practice. No use,  
distribution or reproduction is permitted  
which does not comply with these terms.

# Real-time robot topological localization and mapping with limited visual sampling in simulated buried pipe networks

Xiangyu S. Li<sup>1\*</sup>, T. L. Nguyen<sup>1</sup>, Anthony G. Cohn<sup>1,2,3,4</sup>,  
Mehmet Dogar<sup>1</sup> and Netta Cohen<sup>1\*</sup>

<sup>1</sup>School of Computing, University of Leeds, Leeds, United Kingdom, <sup>2</sup>Alan Turing Institute, London, United Kingdom, <sup>3</sup>Tongji University, Shanghai, China, <sup>4</sup>Shangdong University, Jilin, China

**Introduction:** Our work introduces a real-time robotic localization and mapping system for buried pipe networks.

**Methods:** The system integrates *non-vision-based* exploration and navigation with an *active-vision-based* localization and topological mapping algorithm. This algorithm is selectively activated at topologically key locations, such as junctions. *Non-vision-based* sensors are employed to detect junctions, minimizing the use of visual data and limiting the number of images taken within junctions.

**Results:** The primary aim is to provide an accurate and efficient mapping of the pipe network while ensuring real-time performance and reduced computational requirements.

**Discussion:** Simulation results featuring robots with fully autonomous control in a virtual pipe network environment are presented. These simulations effectively demonstrate the feasibility of our approach in principle, offering a practical solution for mapping and localization in buried pipes.

## KEYWORDS

pipe networks, topological mapping, localization, autonomous control, robot simulation

## 1 Introduction

The development of indoor robot localization technology has gained substantial interest from researchers, industry, and a large user base (Jensfelt, 2001). Pipeline localization is also witnessing increased attention, driven by the need for more efficient and cost-effective maintenance and repair of critical infrastructure such as oil and gas pipelines, water and sewage systems, and electrical transmission lines (Mounce et al., 2021). However, unlike most indoor settings, there remain significant challenges in miniaturizing and making pipeline robots autonomous. Such autonomous robots must have accurate and dependable sensors for data collection and decision-making, as well as efficient and compact power sources and actuators. Inspection, localization and navigation algorithms must operate in real-time, while also being computationally efficient to reduce power requirements and minimize hardware needs. Existing infrastructure maps may not accurately represent the current state of pipeline networks due to changes that occur over time. As a result, relying solely on traditional maps can lead to outdated information, potentially hindering maintenance and repair operations. To overcome this challenge, autonomous mapping

techniques offer a solution by providing real-time updates, ensuring the most accurate and up-to-date information. However, accurate localization and navigation are crucial for robots to efficiently maneuver through the pipeline network and carry out tasks at specific locations. Traditional maps, which primarily focus on geometric features, may not provide sufficient information for autonomous robot navigation and decision-making. This limitation necessitates the use of topological maps that capture not only the geometric features but also the connectivity and relationships between pipeline components, such as junctions and manholes. By leveraging topological maps, robots can plan optimal paths, navigate complex networks, and perform advanced computations, such as semantic reasoning and planning. Therefore, to address these challenges and improve the efficiency and effectiveness of pipeline inspection and maintenance, we propose a topological mapping algorithm. Topological representations are compact as well as straightforward and intuitive (Kortenkamp and Weymouth, 1994), providing a clear understanding and interpretation of the environment. Indoor robots utilize mapping and localization to determine their position within an environment, even when direct or relative location sensing is unavailable. A commonly used technique is Simultaneous Localization and Mapping (SLAM), which employs geometric representations to construct maps of unknown environments while simultaneously determining the robot's location on the map (Mur-Artal et al., 2015; Rugg-Gunn and Aitken, 2022).

Compared with a map of geometric features, a topological map is more efficient in terms of storage and computation, and insensitive to error accumulation. Topological maps provide an abstraction of the physical environment and a more sophisticated spatial representation. These abstractions are well suited to built environments, both indoor and underground, allowing robotic systems to perform advanced computations such as semantic reasoning and planning. Furthermore, as suggested by Chrastil and Warren (2013), topological maps align with human spatial intuition, facilitating more natural human-computer interactions. Our work presents an algorithm for topological localization and mapping in buried pipe networks, enabling autonomous navigation for lightweight robots over extended periods.

We present a hybrid approach integrating *vision-* and *non-vision-based* strategies in a single fully autonomous framework. In particular, we use *non-vision-based* exploration and navigation while our localization and mapping rely on an active-vision algorithm, which is only activated at topologically key (landmark) locations. Nguyen et al. (2022) describe an autonomous robot for pipeline inspection which they use to demonstrate exhaustive *non-vision-based* exploration of a physical pipe network. The small size of the robot (70 g, navigating through 150 mm pipes) imposes severe limitations on sensors, motors and battery power. Instead of using a camera, the robot uses distance sensors to navigate through the pipes and to execute turns and maneuvers at junctions and dead ends. Here, we propose to add localization and mapping capabilities to this platform. We implement this robot in a simulated pipe network to develop and test our active vision, localization and mapping algorithm. In our work, we focus primarily on junctions, as these are sufficient for the topological mapping of the network. Once the robot arrives at a junction, it uses its camera to collect an image dataset of the location. The robot localizes by comparing

the images from its current location with the data in its database. This localization step includes orientation matching, i.e., calculating the robot's rotation relative to the orientation in the database. An active vision step is included to increase the accuracy and robustness of the localization. Finally, the robot either identifies its location based on a good match with an existing location in the database or defines its location as novel, adding the location information and the associated image set to the database and the topological map.

This paper is organized as follows. The remainder of this section discusses related work. In addition to the creation of topological maps such as mentioned above, we have also developed control algorithms for the fully autonomous navigation of the robot in our experiments, so that the robot can generate topological maps in real-time as it explores the pipeline. The methods are described in Section 2, including a brief overview of the autonomous control of the simulation robot presented by Nguyen et al. (2022) (Section 2.1). Although the physical robot was successful in exploring the pipe network, this study relies on a motorized camera that has yet to be implemented. Hence, the experiments with localization and topological mapping are conducted in a simulated environment. The results are presented in Section 3, including simulations of the robot recognizing junctions in real-time and constructing a topological map as it moves through the pipeline (Section 3.6). Finally, the paper concludes with a discussion in Section 4.

## 1.1 Related work

Localization in robotics involves both relative and absolute positioning methods. Absolute positioning relies on directly obtaining coordinate information through methods like GPS or satellite positioning (El-Rabbany, 2002). However, such methods are not available in the underground pipe networks and therefore are not useable. While relative positioning, also known as dead reckoning, estimates the current position relative to the initial point using sensors such as wheel encoders (Aqel et al., 2016). However, odometry, a common method for dead reckoning in wheeled robots, has limitations such as tire slippage, meandering trajectories, and intrinsic noise (Wang et al., 2015). On the other hand, lidar and camera vision positioning technologies offer better stability and accuracy, but each has its advantages and disadvantages (Nistér et al., 2004; Wang et al., 2017).

In the context of pipeline localization, various approaches have been proposed. Hertzberg and Kirchner (1996) introduced landmark features in pipelines, although the reliability of landmark classification may be questionable. Vision-based localization has been implemented in pipelines by leveraging topological features such as elbows and branches (Lee et al., 2009). These approaches utilize image analysis to extract information about landmarks and update the robot's records for localization purposes. Some studies have focused on utilizing onboard sensors, such as IMU, gyro, and leak sensors, to localize robots in water pipes (Wu et al., 2019). Other works combine visual odometry and pipeline topological mapping for localization in sewage systems (Alejo et al., 2019).

Localization techniques have been employed in diverse robotics environments. Lopes et al. (2020) developed a reconfigurable robotic miner prototype that utilizes various sensing techniques,

including tactile, conductivity, and inertial measurements, for localization and mapping during mining operations. They employ advanced probabilistic sensor fusion techniques for localization in the absence of a globally fixed coordinate frame. Topological mapping and localization have also been extensively studied in urban environments (Kortenkamp and Weymouth, 1994; Blaer and Allen, 2002).

Kalman filter-based methods have been widely utilized for sensor data fusion in pipeline navigation. Siqueira et al. (2016) proposed a sensor data fusion technique using the Kalman filter for pipe navigation. Liu and Kryz (2012) explored the use of a laser range finder for pipe inspection, while Maneewarn and Thung-od (2015) presented the ICP-EKF localization method for boiler inspection robots. These studies collectively enhance our understanding of sensor data fusion, laser range finder utilization, and localization algorithms, providing valuable insights for the development of innovative pipe inspection robotics solutions.

SLAM techniques have been successfully applied in pipeline environments. PipeSLAM, introduced by Ma et al. (2017), addresses the challenges of feature-sparse water pipes by utilizing the Rao-Blackwellized particle filter to integrate odometry and visual information. Additionally, Kryz and Najjaran (2007) developed a VSLAM system for pipe inspection robots, demonstrating the effectiveness of visual-based methods in pipeline localization. By combining visual information and SLAM techniques, the robot can simultaneously map the pipe environment and estimate its position.

Node localization in robotic sensor networks and the development of smart wireless robotic systems have been significant areas of research in pipeline inspection. Wu et al. (2015) focused on node localization in robotic sensor networks for pipeline inspection. They proposed techniques and algorithms to accurately determine the location of nodes within the network, enabling effective monitoring and control of robotic systems in pipeline environments. Kazeminasab and Banks (2022a) introduced a size-adaptable in-pipe wireless robotic system. This system incorporated a two-phase motion control algorithm to navigate and inspect water distribution systems. By integrating wireless communication capabilities, smart sensors, and adaptable motion control algorithms, the SmartCrawler system demonstrated advancements in the field of pipeline inspection, enhancing the efficiency and effectiveness of water distribution system monitoring and maintenance.

Acoustic echo localization has emerged as a promising technique for pipe inspection robots in recent years. Worley et al. (2020) focused on leveraging acoustic echoes to estimate the location and geometry of pipes accurately. By analyzing the time-delay and frequency characteristics of reflected acoustic signals, the proposed method provided valuable information for localization and mapping tasks in pipe inspection. This work offering a non-intrusive and efficient approach to gather essential information about pipe conditions and geometries.

Kazeminasab et al. (2020) proposed various techniques that integrated sensor data, such as odometry and inertial measurements, with map information to achieve reliable robot localization within the intricate pipe network. Additionally, magnetic induction communication has been utilized for localization in water distribution systems. Kazeminasab and Banks (2021) synchronized

a wireless sensor module localizer with the motion controller, enabling long-distance inspection capabilities. Kazeminasab and Banks (2022b) presented a localization method for in-pipe robots using a particle filter-based localizer synchronized with a multi-phase motion controller.

In comparison to these works, our approach aims to combine low-power exploration (using lower-power distance sensors and odometry) and vision-based topological localization for pipeline networks. We utilize low-power distance sensors for maintaining robot movement within the pipeline, while vision is employed at junctions for localization and construction of a topological map. Our focus is on lightweight underground pipeline robots, taking into consideration power and computation constraints. By integrating localization and motion control, we aim to achieve accurate and efficient smart motion for long-distance inspection in pipeline networks.

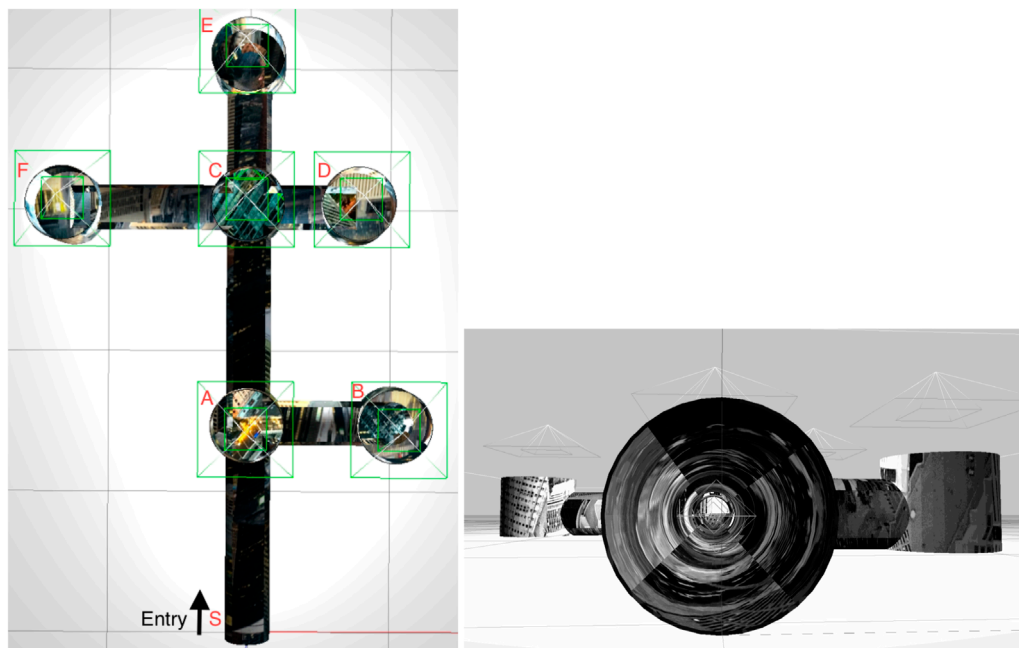
## 2 Materials and methods

The algorithm for automatic control is provided by Nguyen et al. (2022), and a detailed method overview can be found there. We present a brief discussion in Section 2.1. The junction recognition process uses normalized cross-correlation (NCC) image matching (Section 2.2). By comparing the current junction with all the junctions data collected in the database, the robot's task is to determine whether the junction has been visited before, and if so, localize (Section 2.3). Otherwise, this new junction is added to the image database and the topological map is updated (Section 2.4).

### 2.1 Robot control for navigation and exploration

The miniature robot is equipped with three range sensors, an inertia measurement unit (IMU), two wheel-leg encoders, and a camera (800 × 800 Pixels) for sensing. The robot starts at the entrance of the pipe network and exhaustively explores all the sections before returning to the starting point. While exploring the pipe network, the robot encounters all junctions, dead-ends, and obstacles at least once, using its sensors to autonomously navigate and maneuver to deal with these conditions based on its estimated states. In our work and Nguyen et al. (2022), we do not use the camera for autonomous control of the robot but only fuse the data of the other three types of sensors to estimate the robot's state. Eleven robot-in-pipe states capture different positions at T-junction, at left/right branches, at left/right corners, at straight pipe centerline, inclined in straight pipe, at a dead-end, at a cross junction, at open-space, and finally, to detect when the robot is approaching a collision. Three time-of-flight range sensors are installed at the front, front-left, and front-right of the robot to measure distances from the robot to the surrounding environment. These distances, combined with IMU data and their historical data, provide sufficient information for the algorithm to calculate the current robot's local state (Nguyen et al., 2022).

Once the robot confirms its estimated local state, it makes a high-level decision to turn right or left at an angle, go straight,



**FIGURE 1**

The simulated pipe network in the Gazebo simulator (Koenig and Howard, 2004). Pipe diameter 200 mm. Left: External view with the six junctions and entry point manually labeled. Right: inside view of a pipe with added texture.

or turn around. By default, the robot decides to always take the right most unexplored branch at any junction to guarantee an exhaustive exploration of all sections of the pipe network. Specifically, the robot turns right at a cross junction, T-junction, right corner, and right branch. It goes straight at a left branch and in a straight pipe. It turns left at a left corner and turns around at a dead-end or upon encountering a significant obstacle (that blocks passage through the pipe cross-section). Depending on the direction of an impending collision, the robot turns left or right to move away from the pipe walls or obstacles. A detailed table of actions taken regarding the robot state was explained in Nguyen et al. (2022). After making a high-level decision, a low-level motor controller is called to fine-tune the robot's direction and velocity in the pipes. Encoder values and their historical data are used to calculate the maneuver's speed and estimated turning angles.

As described above, the robot uses three distance sensors mounted on the front to recognize a junction ahead. At this point, the robot faces the center of the junction ahead. In our work, we augment the control algorithm in Nguyen et al. (2022) to command the robot to move forward from the entrance to a junction by a set distance (equal to the estimated radius of the junction). Once at the approximate center of a junction, sampling, localization, and mapping take place, as described in the following sections. Once complete, the robot returns to the entrance of the junction, and the original control algorithm resumes.

We noted that the IMU, odometry and laser data are not used for localization but only for navigation and low-level control. Our localization method as described in this paper relies on vision information only.

## 2.2 Identification of known/unknown junctions

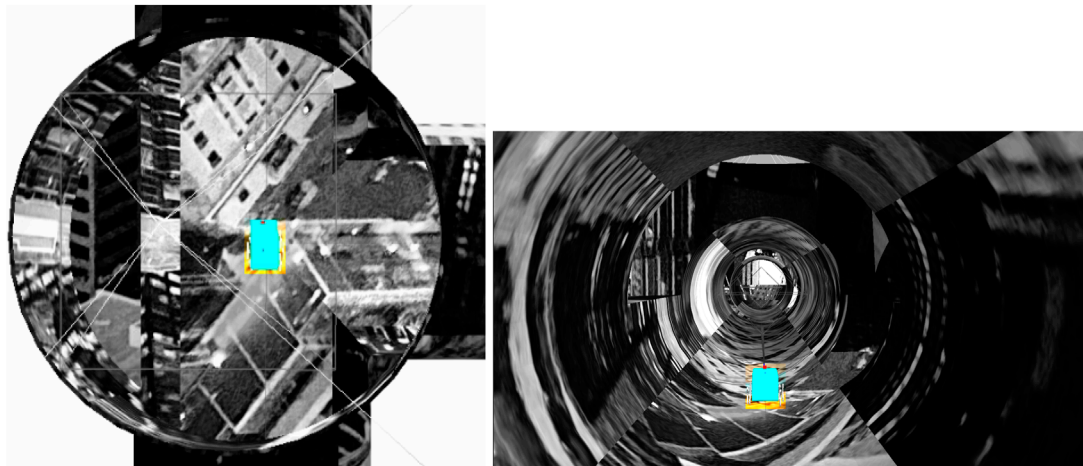
The robot visits different junctions as it explores the pipe network (Figure 1). During this exploration, the robot may visit the same junction multiple times. We call such previously visited junctions *known junctions*. We call junctions that are encountered by the robot for the first time *unknown junctions*. This section describes how the robot can identify whether it is currently in a known or unknown junction. During autonomous exploration and topological mapping, the database consists only of known junctions. As more junctions are encountered, the database and map are updated.

When the robot is approximately at the center of a junction (Figure 2), it collects images all around the junction (See example in Figure 3). The 360° view of the current junction is obtained by rotating the camera that is mounted above the robot (Figure 5) in discrete steps. Given the diameter of the simulated pipe, each image covers an angular width of about 90°. We want a significant overlap between images and found that a small set of 6 images (corresponding to 60-degree rotations between images) is sufficient for robust localization (see Section 3.2 in the Results). As the robot explores the environment, it collects images from each junction, and the stored sets of images are given by:

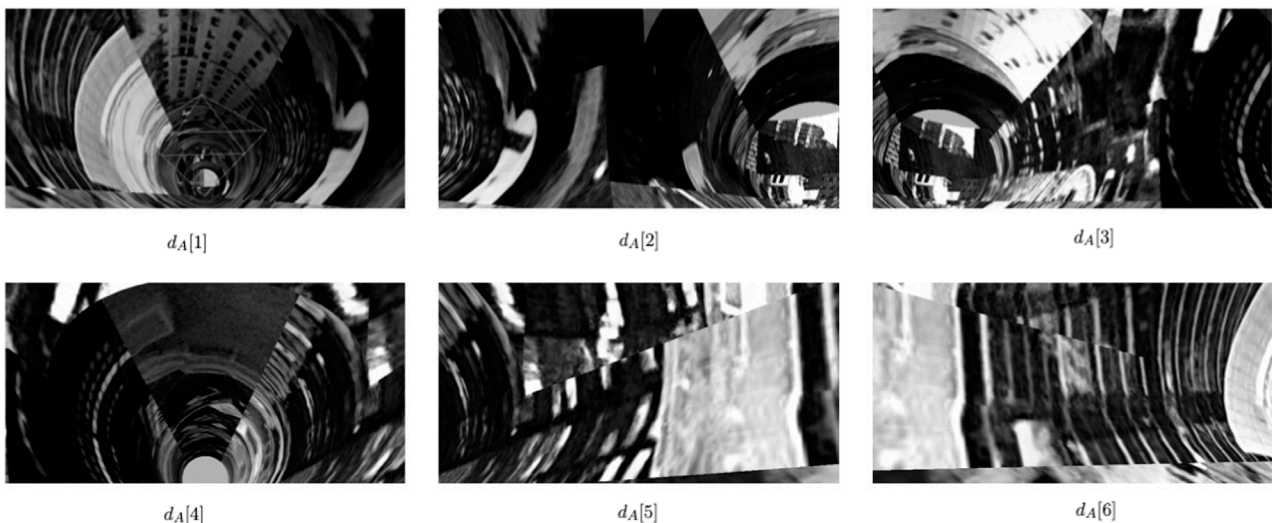
$$D = \{d_i\}_{i=1}^{\text{NumJunctions}}$$

where  $D$  is the database, and NumJunctions is the number of known junctions at that point in the exploration.

To identify a junction, the robot compares the image set collected at the current junction,  $d_c$ , with the image sets of all known junctions. Given two image sets  $d_c$  and  $d_j$ , we perform image



**FIGURE 2**  
Robot (appearing as a light blue rectangle in a top-down and rear view) at the approximate center of junction A.



**FIGURE 3**  
Collected image set from junction A (example of  $d_A$ ).

matching to define the similarity  $S(d_c, d_i)$  between the two junctions (see Algorithm 2 below). We then compute the maximum similarity score,  $S_{max}$ , across all possible known junctions, to obtain the best candidate for a known junction, here labeled with index  $j$ , using:

$$(S_{max}, j) = \left( \max(S(d_c, d_i)), \underset{S}{\operatorname{argmax}} \right)$$

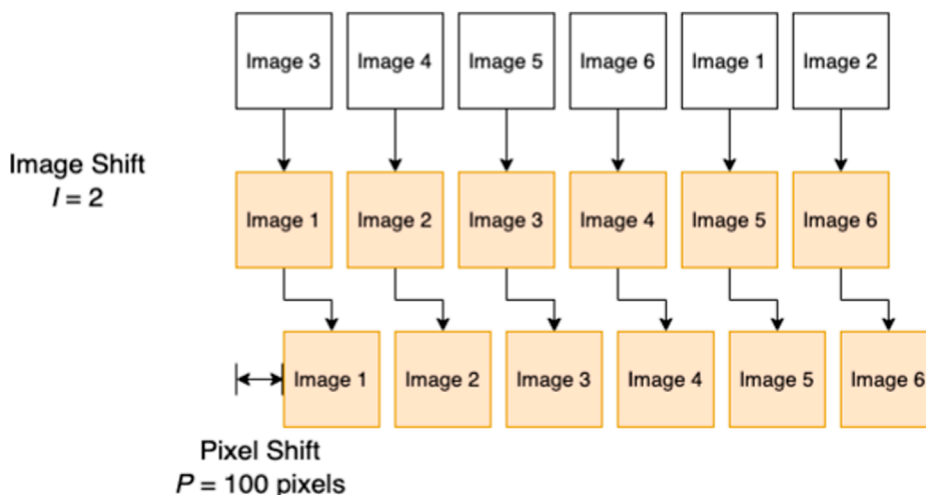
Finally, we determine whether the current junction is a known junction or an unknown junction by thresholding this maximum similarity score  $S_{max}$ , shown in Algorithm 1. If the similarity is higher than the threshold, we identify the current location as a known junction  $j$  in the database. Otherwise, we perform active vision  $A(d_c, d_j)$  to better align the image sets (see Section 2.3; Algorithm 3) and repeat the image-set comparison with junction  $j$  in the database. If the similarity is still lower than the threshold, we define this junction as an unknown junction and add the current

```

1 Known/Unknown junction  $K(d_c, d_j)$ ;
   Input : Current image set  $d_c$ , Corresponding max similarity image set  $d_j$  in the database
   Output : known/unknown junction True/False
2 threshold can be determined by Figure 7;
3 if  $S_{max} > \text{similarity\_threshold}$  then
4   | return known;
5 else
6   |  $S_{max} = A(d_c, d_j)$ ;
7   | if  $S_{max} > \text{similarity\_threshold}$  then
8     | return known;
9   | else
10  | return unknown;
11  | end
12 end
    
```

Algorithm 1. Algorithm for identification of known/unknown junctions.

image set  $d_c$  to the database as a new junction. We also add this junction to the topological map, as described in Section 2.4.



**FIGURE 4**  
 A illustration for the image shift  $I$  and pixel shift  $P$ . (white: currently sampled junction, orange: image set of a known junction from the stored dataset).

Normalized cross-correlation is used to calculate the maximum similarity,  $S(d_c, d_i)$ , and corresponding offset between two different image sets. For greater realism, only grayscale information is used for image analysis. Image matching is performed for every possible junction pairing (one corresponding to the robot’s present location and the other for each ‘target’ junction in the database). There are six images in each image set, and hence six possible offsets between two image sets. For a given offset, the normalized cross-correlation is computed (in the Fourier domain) between all images in the current location with the candidate matching junction in the database. For each candidate junction, after finding the image offset with the maximum similarity, the horizontal pixel offset is set to the position of the maximum cross-correlation value. Looping through all possible candidate junctions, the pairwise similarity and the corresponding image and pixel shifts (offsets) are obtained. Finally, the highest similarity score is used to pin down a single candidate junction. The image-matching data for this junction pair is then used as a basis for the active vision step.

The normalized cross-correlation (NCC) is a commonly used region-based method in image feature matching. In the absence of a single 360° panoramic view of the entire junction, here we apply NCC to one image-pair at a time and increment the possible image shifts to allow for the robot entering the junction from different pipe sections, as illustrated in Figure 4. We match the similarity of gray pixel values in the whole image field in the two images ( $I_1$  and  $I_2$ ). For  $I_1$ , we calculate the normalized cross-correlation coefficient between it and the  $I_2$ , and the corresponding pixel shift that maximizes the match.

We exploit the similarity in the formulation of the cross-correlation to a convolution function, which can be implemented efficiently with a Fast Fourier Transform (FFT). To perform a cross-correlation, we perform a convolution between one image and a second conjugate image (see Algorithm 2). The implementation uses python3.6, opencv2, Numpy, and fftconvolve function from scipy.signal.

We note that a preprocessing step is taken before performing the cross-correlation. Because the lower part of the image contains the

```

1 Image Set Cross Correlation  $S(d_a, d_b)$ ;
2 Input : Two Image Dataset  $d_a$  and  $d_b$ 
3 Output : Similarity  $S$ , Image Shifts  $I$ , Pixel Shifts  $P$ 
4 Iterate through all the images and cycle through different image shifts;
5 for  $j \leftarrow 0$  to  $\text{sizeof}(d_a)$   $J$  do
6   for  $k \leftarrow 0$  to  $\text{sizeof}(d_b)$   $K$  do
7     Read image from junction dataset  $a$  and  $b$ ;
8     Pre-processing, Read the image in the gray channel and crop the bottom half;
9     Image1  $\leftarrow d_a[j]$ ;
10    Image2  $\leftarrow d_b[(j+k)\%K]$ ;
11    Start Cross-Correlation between two images;
12    Calculate the difference between each pixel and the mean of every pixel of one image;
13    Image1  $\leftarrow$  (Image1 - Mean(Image1));
14    Image2  $\leftarrow$  (Image2 - Mean(Image2));
15    Faster to flip up down and left right then use fftconvolve;
16    Flip1  $\leftarrow$  FlipUpDown(FlipLeftRight(Image1));
17    OutMatrix  $\leftarrow$  fftconvolve(Image2, Flip1);
18    Finally, normalize the resulting matrix;
19    similarity  $s \leftarrow$  Max value in OutMatrix;
20    pixel shifts  $p \leftarrow$  Index of  $s$ ;
21    Add  $s$  to the  $s_{list}$ ;
22    Add  $p$  to the  $p_{list}$ ;
23     $s_{list} \leftarrow s$ ;
24     $p_{list} \leftarrow p$ ;
25  end
26 Sum  $s$  and calculate the mean;
27  $s = \text{Mean}(\text{Sum}(s_{list}))$ ;
28 Sum  $p$  together and calculate the mean;
29  $p = \text{Mean}(\text{Sum}(p_{list}))$ ;
30 end
31 Find the max  $s$  and  $p$  based on different offsets  $j$  for a single candidate junction;
32  $S \leftarrow s$ ;
33  $P \leftarrow p$ ;
34  $I \leftarrow j$ ;
35 return  $S, P, I$ ;
    
```

**Algorithm 2.** Algorithm for computing the similarity of two image sets and corresponding image shifts, pixel shifts.

robot, which will affect the accuracy of the comparison results, we crop out the lower half of the image. Next, we subtract the mean pixel intensity.

### 2.3 Pipeline robot localization using active vision

As previously mentioned, a minimal change to this control algorithm was implemented to allow the robot to approach the approximate center of the junction. In this section, we describe a second addition to the control algorithm: the implementation of active vision.

```

1 Active Vision  $A(d_c, d_j)$ ;
   Input : Current image set  $d_c$ , Corresponding max similarity image set  $d_j$ 
   Output : Similarity  $S$ 
2 Using Algorithm 2 to calculate the similarity  $S$ , image shifts  $I$ , pixel shifts  $P$ ;
3  $S, P, I \leftarrow S(d_c, d_j)$ ;
4 RelativeAngle  $\leftarrow (I \times 2\pi/6 + P \times cam\_width/pixel\_width)$ ;
5 Rotate the camera according to RelativeAngle;
6 Initialize  $S_1$ ;
7 Move the robot a little distance forward.;
8 MoveRobot();
9  $d_c \leftarrow RecordImageSet()$ ;
10  $S_1 = S(d_c, d_j)$ ;
11 if  $S_1 > S$  then
12 | return  $S_1$ ;
13 else
14 | MoveBackCenter();
15 end
16 Move the robot a little distance backward.;
17 MoveRobot();
18  $d_c \leftarrow RecordImageSet()$ ;
19  $S_1 = S(d_c, d_j)$ ;
20 if  $S_1 > S$  then
21 | return  $S_1$ ;
22 else
23 | MoveBackCenter();
24 end
25 Move the robot a little distance left.;
26 MoveRobot();
27  $d_c \leftarrow RecordImageSet()$ ;
28  $S_1 = S(d_c, d_j)$ ;
29 if  $S_1 > S$  then
30 | return  $S_1$ ;
31 else
32 | MoveBackCenter();
33 end
34 Move the robot a little distance right.;
35 MoveRobot();
36  $d_c \leftarrow RecordImageSet()$ ;
37  $S_1 = S(d_c, d_j)$ ;
38 if  $S_1 > S$  then
39 | return  $S_1$ ;
40 return  $S$ ;

```

Algorithm 3. Algorithm for using the active vision to find the maximum similarity of two image sets based on image shifts, pixel shifts.

Given the image shift  $I$ , pixel shift  $P$ , as well as the angle between successive images ( $2\pi/6$  radians), camera width ( $2\pi/4.5$ ), and pixel widths (800 pixels), the relative angle between the two image sets, is  $(I \times 2\pi/6 + P \times cam\_width/pixel\_width)$ . To execute the active vision, we rotate the camera according to this angle, and then move the robot a little distance forward, backward, left, and right, in turn from its position (roughly in the center of the junction). We continue cross-correlating with the same dataset in the database until we find a direction that increases the similarity. If active vision increases the similarity sufficiently to exceed to similarity threshold, the junction can be identified (it is a known junction). Otherwise, it is classified as an unknown (i.e., new) junction, as described in Section 2.2; (Algorithm 3).

## 2.4 Topological mapping

Topological maps capture the relationships between elements of a map that are represented as nodes (or vertices), connected by edges (Garcia-Fidalgo and Ortiz, 2015). In our case, the map captures connectivity between pipe segments and the correct order of these edges in each junction. While distance information can be important for some purposes, the adjacency of pipe segments and junction information suffice for path planning in the network. The key to constructing topological maps is identifying appropriate spatial points of interest in the environment as vertices and extracting sufficient semantic information from these locations. In addition to their compactness and elegance, topological maps may be more stable than spatial maps in the face of closed loops (in the absence of absolute position information). It is possible to consider a more spatially grounded topological map, in which edges (i.e., pipe

```

1 Topological Mapping  $T(d_c, D, V_{last}, W, G)$ ;
   Input : Current image set  $d_c$ , Database  $D$ , last vertex  $V_{last}$ , Label  $W$ , and Graph  $G$ 
   Output : Graph  $G$ 
2 Iterate through all the image sets in the database  $D$ ;
3 Initialize  $S_{max}, P_{max}, I_{max}$ ;
4 for  $i \leftarrow 0$  to sizeof( $D$ ) do
5 | Using Algorithm 2 to calculate the similarity  $S$ , image shifts  $I$ , pixel shifts  $P$ ;
6 |  $S, P, I \leftarrow S(d_c, d_i)$ ;
7 | if  $S > S_{max}$  then
8 | |  $S_{max} = S$ ;
9 | |  $P_{max} = P$ ;
10 | |  $I_{max} = I$ ;
11 end
12 Find the max similarity  $S_{max}$  through all the comparisons and corresponding  $d_{max}$ , image shifts  $I_{max}$  and pixel shifts  $P_{max}$ ;
13 Determines if the current junction is a known or known junction.;
14 result =  $K(d_c, d_{max})$ ;
15 if result == True then
16 | It's a known junction. The vertex is the same as the vertex represented by  $d_{max}$ ;
17  $V_c \leftarrow GetVertex(d_{max})$ ;
18 last vertex  $V_{last}$ ;
19 if Edge( $V_{last}, V_c$ ) doesn't exist then
20 | Create a new Edge  $E$ ;
21 |  $E \leftarrow NewEdge(V_{last}, V_c)$ ;
22 | Add the label  $W$  to the edge  $E$ ;
23 | AddLabel( $E, W$ );
24 |  $W \leftarrow W + 1$ ;
25 | Add Edge  $E$  to Graph  $G$ ;
26 |  $G \leftarrow AddEdge(E, G)$ ;
27 | return  $G$ ;
28 else
29 | It's an unknown junction;
30 | Add the current image set  $d_c$  to the database  $D$ ;
31 |  $d_c \rightarrow D$ ;
32 | Add a new vertex  $V_n$ ;
33 |  $V_n \leftarrow NewVertex(V_n, d_c)$ ;
34 |  $G \leftarrow AddVertex(V_n, G)$ ;
35 | Create a new Edge  $E$ ;
36 |  $E \leftarrow NewEdge(V_{last}, V_n)$ ;
37 | Add the label  $W$  to the edge  $E$ ;
38 | AddLabel( $E, W$ );
39 |  $W \leftarrow W + 1$ ;
40 | Add Edge  $E$  to Graph  $G$ ;
41 |  $G \leftarrow AddEdge(E, G)$ ;
42 | return  $G$ ;
43 end

```

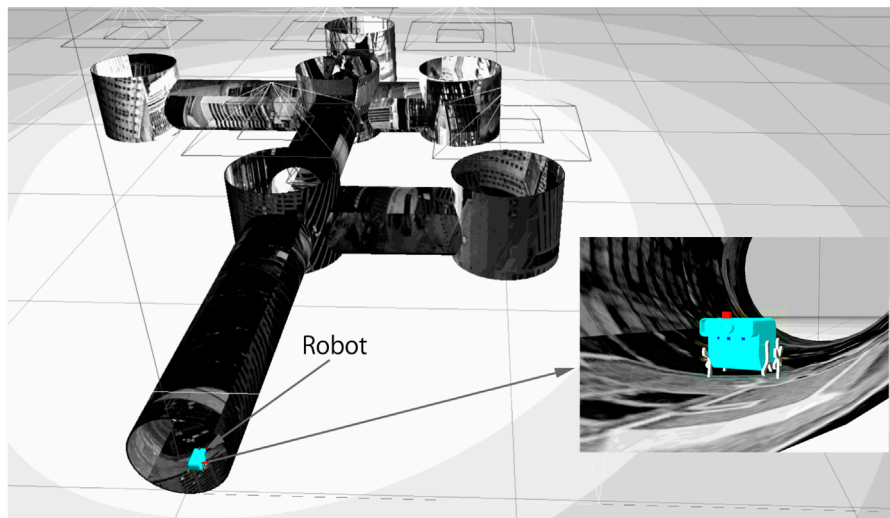
Algorithm 4. Algorithm for processing a junction during topological map building.

sections) are assigned information (e.g., a distance estimate from dead reckoning, the time taken to move between two vertices, or the energy consumed). In our case, however, junctions contain sufficient semantic information about the scene.

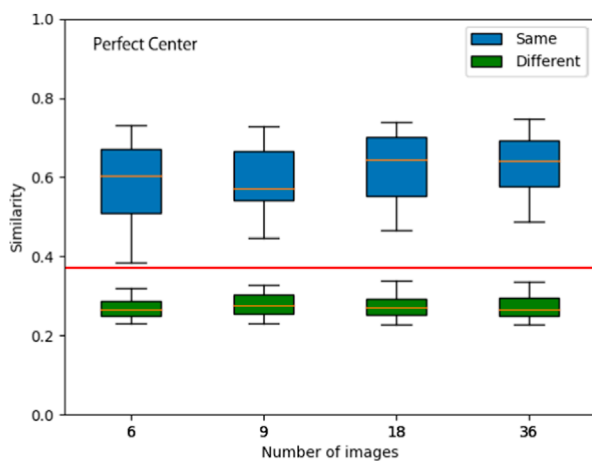
Whether a junction is known or unknown dictates whether the topological map is up to date. Given an unknown junction, the robot firstly adds the image set  $d_c$  to the database  $D$ , creates a new vertex  $V_n$ , and adds an edge  $E$  between it and the previous vertex the robot visited  $V_{last}$ . When moving between two known junctions, the edge may or may not have been traversed previously. In the absence of an edge in the map, a new edge will be added between the previous junction to the current position. Topological maps typically encode vertices and edges in an adjacency graph. For navigation purposes, it is also useful to disambiguate the ordering of the edges for each junction. Here, we use the order in which edges are traversed to assign them integer labels ( $W$ ). Whenever a new junction is added, if the edge used to reach the junction is new as well, it is assigned the next unused integer as its label. Thus, once the map is complete, each edge has a unique label (see Algorithm 4). Given the exploration algorithm (in our case, consistently taking the right-most branch), it is then possible to determine the relative orientation of the pipe sections.

## 3 Results

In this section, we present our experiment with a simulated robot in a pipe environment. Our overall goal is to test the performance of our system by building a topological map doing autonomous



**FIGURE 5**  
Robot simulated in the pipe network.



**FIGURE 6**  
Assessing junction similarity for different sized image sets. Junction similarity results for 24 image sets (4 per junction at 90-degree intervals) under Perfect Center conditions. The similarity is compared for different numbers of images per image set. Boxplot (showing the Upper whisker, Upper quartile, Median, Lower quartile, and Lower whisker from the top to the bottom for the known junction (blue) and unknown junctions (green).) Applying a threshold of 0.36 (red horizontal line) would allow for accurate recognition of known junctions and distinguishing unknown junctions.

control. However, before we present these topological map-building results in Section 3.6, we will present a set of simpler experiments that we have performed to test the localization performance of our robot. The following five experiments will be presented.

1. *Perfect Center*: Our first experiment tests if the robot can identify which junction it is in with respect to the junction in the database when the robot is initialized perfectly at the center of the junction. In this case, autonomous navigation is not

used. We also use a robot initialized perfectly at the center of junctions to decide how many images to collect in each junction. The experiment suggests that six images are sufficient (see Section 3.2).

2. *Noise*: To validate that six images suffice even for the imperfect positioning of the robot in the center of the junction, we repeated the above experiments such that robot will be located at 3 cm off center (see Section 3.3).
3. *Active*: To test the robot's ability to improve its localization through active vision, in this set of experiments, the robot is initialized at the off-center position and uses active vision (see Section 3.4).
4. *Entrance*: To more realistically simulate the process of the robot traveling autonomously in the pipeline, we place the robot on the edge of the junction in the pipeline, and perform a similar set of experiments. The robot moves forward a distance of the junction radius to the center before switching on its camera (see Section 3.5).
5. *Autonomous*: The robot is simulated under the fully autonomous exploration and control mode. A topological map is built (see Section 3.6).

### 3.1 Experimental setup

We model the mobile robot in a pipe network (Figure 1), and test our localization algorithm using ROS (Quigley et al., 2009) and the physics simulator Gazebo 9 (Koenig and Howard, 2004). The robot 3D model was designed in Solidworks (see Figure 5; Nguyen et al. (2022) for details of the design) and imported into Gazebo. The imported model in Gazebo is shown in Figure 5. The robot has six wheel-legs, with wheel diameter 28 mm. Three left wheel-legs are connected and are actuated by a DC motor. Similarly, three right wheel-legs are connected and controlled by the second DC motor. The two motors are independent and controlled by



**TABLE 1** Summary of similarity results across different image sets under different conditions. Note the low similarity obtained for unknown junctions, across all conditions.

Similarity $\pm$ std	Perfect center	Noise	Active	Entrance	Autonomous
Known junction	0.700 $\pm$ 0.157	0.614 $\pm$ 0.112	0.675 $\pm$ 0.166	0.643 $\pm$ 0.150	0.631 $\pm$ 0.109
Unknown junction	0.276 $\pm$ 0.028	0.276 $\pm$ 0.027	0.276 $\pm$ 0.024	0.280 $\pm$ 0.031	0.273 $\pm$ 0.025

**TABLE 2** Summary of rotation error across different image sets under different conditions.

Rotation error (radians)	Perfect center	Noise	Active	Entrance	Autonomous
Known junction	0.054	0.044	0.051	0.047	0.101
Unknown junction	0.377	0.473	0.610	0.717	0.856

two PID (proportional integral differential) controllers. Each motor controller is implemented in Gazebo-ROS using the `ros_control` package (allowing complex joint control algorithm to be applied to the DC motor instead of standard differential drive).

Figure 5 shows the simulated pipe structure in Gazebo and the simulated pipe network with the robot. The grayscale texture was imposed on the simulation environment to ensure that junctions look distinct. This is an underlying assumption of our work—i.e. that there is a visual difference between geometrically identical junctions. In a pipe network of any age, this is likely to be true but needs to be experimentally verified in future work. The robot is controlled by sending commands to its two `ros_control` motor controllers. The simulation provides an idealized scenario for the robot model, in which we can develop and test the proposed algorithms for localization and mapping.

### 3.2 Initializing the robot at the center of the junction

We collected images for each junction, sampled with a robot positioned precisely at the center of each junction as a reference database. Figure 3 shows sample data collected from one junction.

To determine a robust number of images that would suffice for localization, we performed a preliminary experiment, in which we placed the robot in the center of each of the six junctions in our network. As each image has a view of about 90°, more than four images would be required to ensure some overlap between adjacent images. We compared image sets with 6, 9, 18, and 36 images. For each case, four image sets were collected from each junction, facing four different directions (at 90-degree intervals) resulting in 24 image sets. We used the same robot to collect all the data sets required for the experiment at the same position in the pipeline. Defining the image sets from different robot orientations of the same junction as “Same” and image sets from different junctions as “Different” we calculated the similarity scores to the Same and Different junctions for image sets with 6, 9, 18, and 36 images. We present the results in Figure 6. While increasing the number of images yields higher accuracy, even with only six images, junctions can be recognized and the rotational error is small ( $\leq 3^\circ$ ). The result demonstrates the robustness of the algorithm to the number of

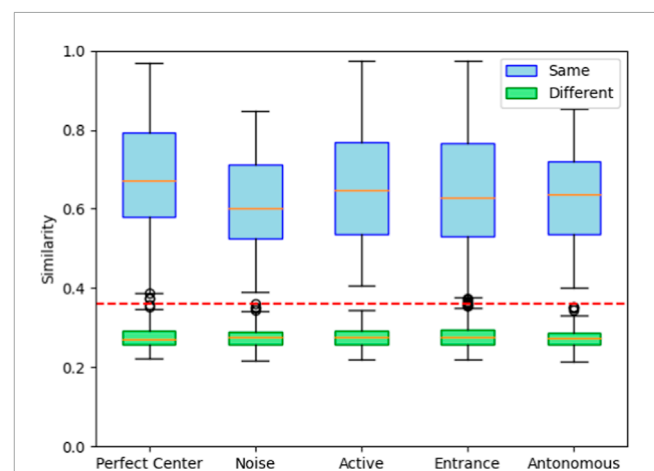
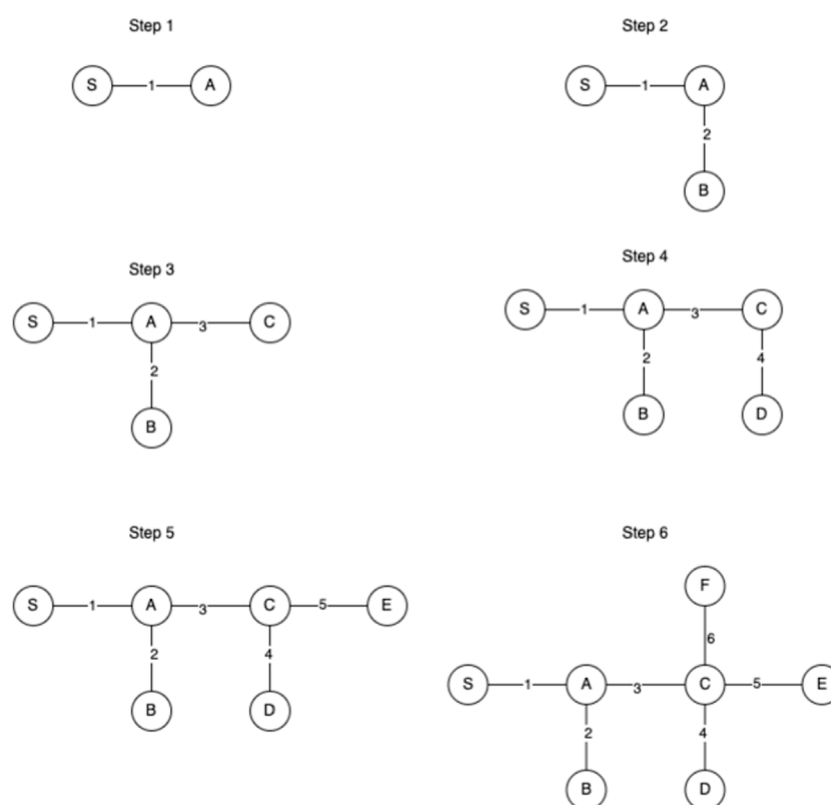
**FIGURE 7**

Image matching between junctions in the simulated pipe network (orange line: mean values, lower quartile: median of the lower half of the dataset, upper quartile: median of the upper half of the dataset, red line: threshold 0.36). Similarity scores for “Same” are computed as  $S(d_i, d_j)$ , where  $d_i$  and  $d_j$  are different imagesets captured from the same junction  $i$ . Similarity scores for “Different” are computed as  $S(d_i, d_j)$ , where  $d_i$  and  $d_j$  are different imagesets captured from the different junctions,  $i$  and  $j$ . Same and different junctions can be robustly separate by the threshold.

images used, with six images being sufficient to reliably identify junctions under perfect center conditions.

To verify whether we can distinguish between known and unknown junctions, we collected sets of images for six junctions within the pipeline under different experimental conditions. For each experimental condition, we collected 5 imagesets at each junction giving us a total of 30 imagesets. We then compared the collected imagesets pairwise. Based on whether these sets of images originated from the same junctions or different junctions, we computed similarity results for junctions labeled as “known” and “unknown”. Notably, each known category comprises 60 data points (imageset pairs coming from same junction) in its box plot, while the unknown category encompasses 375 data points (imageset pairs coming from different junctions). These results are presented in Tables 1, 2; Figure 7.



**FIGURE 8**

Dynamic topological map, generated during autonomous exploration of a simulated pipe network (shown in Figure 1). The six steps correspond to the six updates, each time a new (unknown) junction is visited. The edge labels represent the order in which the junctions were added to the map, and their relative orientations were determined with the knowledge of the rule-based exploration algorithm, yielding an ordering consistent with the pipe network. For visualisation, 90° angles were always assumed between adjacent edges since this is usually the case in real life situations.

### 3.3 Initializing the robot in the off-center junction position

To validate that six images suffice even for the imperfect positioning of the robot in the center of the junction, we repeated the above experiment for eleven image sets collected from robots that were located 3 cm off the center from a random direction ('Noise') (Tables 1, 2; Figure 7). Henceforth, all image sets consist of six images per junction. Next, we tested the robustness of our junction identification algorithm. To verify that the identification of 'known' junctions was reliable, we also checked the relative angular offset (obtained from the image and pixel shifts) and confirmed that these agreed with the different robot orientations.

### 3.4 Using active vision

Our next experiment tests the robot's ability to improve its localization through active vision. While our noisy experiment still showed good results, we expect that more realistic pipe conditions may result in greater slippage or possible errors in estimating the radius of a junction. To address such conditions, here, we move the robot's position to bring it closer to the position at the time of

image set acquisition in the database (corresponding to previous visits to the known junction). Active vision dramatically improves the robot's position recognition accuracy over the 'noise' experiment above.

### 3.5 Initializing the robot at the entrance of a junction

After setting up a simulation environment, we applied the same control methods described in Nguyen et al. (2022) to the simulated mobile robot. We updated the algorithm so the robot could move to the center of the junction and execute image collection for localization and mapping. The robot was able to explore the simulated pipe network exhaustively in the simulation environment, but also experienced failure in some maneuvers due to sensitivity to positioning, when resuming navigation of the network after visiting a junction. Here, we focus on results from successful runs.

To more realistically simulate the process of the robot traveling autonomously in the pipeline, we place the robot on the edge of the junction in the pipeline ('Entrance'). The robot is oriented toward the center of the junction where its distance sensors detect a junction ahead while advancing in the pipeline. From this 'entrance' point,

the robot moves forward at a fixed distance, set to be the radius of the junction. As in previous experiments, we collect image data at the position where the robot finally stops, which can better simulate the heading error in the real world. We collected eleven image datasets of robots after having moved autonomously from the edge to the center of the junction. A comparison of these eleven image sets' data in the database is shown in [Tables 1, 2](#); [Figure 7](#).

From [Table 1](#) results and using a threshold of 0.36 in the similarity, the robot can recognize the same junction, even off-center. In the third step of the active vision experiment, the robot achieves slightly improved mean similarities but is subject to larger variability. The robots from the junction entrance were all identified quite well and were considerably more accurate than the second and third sets of experiments.

### 3.6 Autonomously generating an ordered topological map

We now test our method under fully autonomous conditions. For this experiment, the junction database is initially empty and is updated every time the robot traverses a new and different junction, from those previously visited. We conducted 7 experiments each starting at different position in the pipe network shown in [Figure 1](#). The starting positions were: S,A,B,C,D,E,F. The robot was able to generate the correct topological map successfully from all of these starting positions. In [Figure 8](#), we shown one example. The robot starts at entrance location S and exhaustively explores the network until it returns to the entrance. At every junction, based on the image database of previously visited junctions and the robot's ability to recognize the current junction, the robot's topological map is updated in real-time. A successful map is shown based on the above perfect junction sampling.

We define the topological map as an undirected graph in which each node represents a junction that was visited and imaged by the robot. Each time the robot identifies its location with a junction in the database, it successfully proceeds without updating the map. Conversely, when the robot arrives at a new, unfamiliar node, the current junction and its set of images are successfully added to the database, a new node is added to the graph, and an appropriately labeled edge connects the previously visited node to the current node ([Figure 8](#)).

## 4 Conclusion

Our work employs a multimodal sensing strategy to combine a miniature autonomous robot's exploration, localization, and mapping in a simulated pipe network. While the results presented here are based on simulation experiments, we note that the robot is based on the SolidWorks model used to build a physical miniature model that successfully explored a similar pipe network. A key challenge in this work arises from the miniature robot's limited mobility and limited power. Hence we sought an efficient (power-saving) strategy that requires relatively little computation and

storage. We rely on our finding in ([Nguyen et al., 2022](#)) that the range sensor data in the physical robot is accurate and can make good estimates of the surrounding environment when the robot executes navigation and obstacle avoidance. We proposed that vision sensors are rich in data and we demonstrate that using fast normalized cross-correlation methods suffice in our experiments to identify environmental landmarks. We adapted traditional image-matching techniques to pipeline geometries and rotational movement of the camera. We further propose that active vision may be useful to compensate for the limitations of poor positioning of robots during realistic autonomous movement in pipeline environments. It is important to highlight that although a specific robot model was used in our simulated experiments, the vision-based localization technique we propose is designed to be compatible with various robotic systems, rather than being limited to a particular robot. As long as a robot is capable of exploring the pipe network and equipped with a camera, our method can be employed for topological mapping of the network.

Our work primarily focuses on developing image-matching algorithms for active vision. However, it is important to address the main limitation encountered with the physical robot, which relates to the location and motorization of the mounted camera. In future work, we plan to conduct tests and evaluations in a physical setting using either the same robotic platform or an alternative one that allows for panoramic imaging, enabling us to assess the performance of both the vision-based algorithm and the control mechanism. It should be noted that the sensors used in our work were studied solely in the simulation environment, and therefore, it is challenging to reliably estimate the accuracy of the robot in an actual physical setting. Furthermore, conducting experiments with physical robots in realistic pipe networks introduces additional complexities due to sensor limitations, robot stability, and control issues. It is crucial to acknowledge the significant challenges associated with system integration and electronic isolation in real-world pipe networks, which are often wet, dirty, and cluttered. To thoroughly evaluate the performance, reliability, and durability of the in-pipe robot, extensive testing and validation in real-world pipeline environments are essential. Another noteworthy challenge in real-world deployments is failure recovery. For instance, if the robot becomes stuck inside the pipe network, our current method does not provide a solution for rescuing it. Additionally, real-pipe networks often pose the challenge of dark lighting conditions. Consequently, in a practical deployment, the robot may require onboard lighting to support the camera's visibility. Under such settings, the usefulness and feasibility of active vision can be more extensively tested. The experiments conducted in our work have primarily focused on a specific pipe texture. Conducting experiments in a physical pipe environment will enable us to test our proposed method with real pipe textures and verify our hypothesis that the distinct textures at junctions are sufficiently different to enable the differentiation of various junctions. In summary, while our work emphasizes image-matching algorithms for active vision, it is crucial to recognize and address the challenges posed by the physical robot's limitations, system integration in real-world pipe networks, failure recovery, lighting conditions, and the need to validate the method with different pipe textures in physical pipe environments.

## Data availability statement

Original datasets are available in a publicly accessible repository: The original contributions presented in the study are publicly available. This data can be found here: ([https://github.com/SheldonLeeLXY/Simulated\\_Data](https://github.com/SheldonLeeLXY/Simulated_Data)).

## Author contributions

XL designed the study, collected and analyzed the data, and wrote the manuscript. TN assisted with data collection. AC, MD, and NC provided significant guidance throughout the study and critical feedback on the manuscript. All authors contributed to the article and approved the submitted version.

## Funding

This research was funded by the EPSRC Pipebots project [<http://pipebots.ac.uk/>] (project EP/S016813/1). XL is funded by China Scholarship Council (CSC) from the Ministry of Education of China.

## References

- Alejo, D., Caballero, F., and Merino, L. (2019). A robust localization system for inspection robots in sewer networks. *Sensors* 19, 4946. doi:10.3390/s19224946
- Aqel, M. O., Marhaban, M. H., Saripan, M. I., and Ismail, N. B. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* 5, 2–3. doi:10.1186/s40064-016-3573-7
- Blaer, P., and Allen, P. (2002). "Topological mobile robot localization using fast vision techniques," in *Proceedings 2002 IEEE international conference on robotics and automation (IEEE)*, 1031–1036.
- Chrastil, E. R., and Warren, W. H. (2013). Active and passive spatial learning in human navigation: acquisition of survey knowledge. *J. Exp. Psychol. Learn. Mem. cognition* 39, 1520–1537. doi:10.1037/a0032382
- El-Rabbany, A. (2002). *Introduction to GPS: the global positioning system*. Artech house.
- Garcia-Fidalgo, E., and Ortiz, A. (2015). Vision-based topological mapping and localization methods: a survey. *Robotics Aut. Syst.* 64, 1–20. doi:10.1016/j.robot.2014.11.009
- Hertzberg, J., and Kirchner, F. (1996). "Landmark-based autonomous navigation in sewerage pipes," in *Proceedings of the first euromicro workshop on advanced mobile robots (EUROBOT'96)* (IEEE), 68–73.
- Jensfelt, P. (2001). *Approaches to mobile robot localization in indoor environments*. KTH, Superseded Departments (pre-2005), Signals, Sensors and Systems. (Jensfelt, 2001). Ph.D. thesis.
- Kazeminasab, S., Aghashahi, M., Wu, R., and Banks, M. K. (2020). "Localization techniques for in-pipe robots in water distribution systems," in *2020 8th international conference on control, mechatronics and automation (ICCMA)* (IEEE), 6–11.
- Kazeminasab, S., and Banks, M. K. (2021). A localization and navigation method for an in-pipe robot in water distribution system through wireless control towards long-distance inspection. *IEEE Access* 9, 117496–117511. doi:10.1109/access.2021.3106880
- Kazeminasab, S., and Banks, M. K. (2022a). Smartcrawler: a size-adaptable in-pipe wireless robotic system with two-phase motion control algorithm in water distribution systems. *Sensors* 22, 9666. doi:10.3390/s22249666
- Kazeminasab, S., and Banks, M. K. (2022b). Towards long-distance inspection for in-pipe robots in water distribution systems with smart motion facilitated by particle filter and multi-phase motion controller. *Intell. Serv. Robot.* 15, 259–273. doi:10.1007/s11370-022-00410-0
- Koenig, N., and Howard, A. (2004). "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (IEEE), 3, 2149–2154.
- Kortenkamp, D., and Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. *AAAI* 94, 979–984.
- Krys, D., and Najjaran, H. (2007). "Development of visual simultaneous localization and mapping (vslam) for a pipe inspection robot," in *2007 international symposium on computational intelligence in robotics and automation (IEEE)*, 344–349.
- Lee, J.-S., Roh, S.-g., Kim, D. W., Moon, H., and Choi, H. R. (2009). "In-pipe robot navigation based on the landmark recognition system using shadow images," in *2009 IEEE international conference on robotics and automation (IEEE)*, 1857–1862.
- Liu, Z., and Krys, D. (2012). The use of laser range finder on a robotic platform for pipe inspection. *Mech. Syst. signal Process.* 31, 246–257. doi:10.1016/j.ymssp.2012.03.006
- Lopes, L., Bodo, B., Rossi, C., Henley, S., Žibret, G., Kot-Niewiadomska, A., et al. (2020). Robominers—developing a bio-inspired modular robot-miner for difficult to access mineral deposits. *Adv. Geosciences* 54, 99–108. doi:10.5194/adgeo-54-99-2020
- Ma, K., Schirru, M., Zahraee, A. H., Dwyer-Joyce, R., Boxall, J., Dodd, T. J., et al. (2017). "Pipeslam: simultaneous localisation and mapping in feature sparse water pipes using the rao-blackwellised particle filter," in *2017 IEEE international conference on advanced intelligent mechatronics (AIM)* (IEEE), 1459–1464.
- Maneewarn, T., and Thung-od, K. (2015). "Icp-ekf localization with adaptive covariance for a boiler inspection robot," in *2015 IEEE 7th international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics (RAM)* (IEEE), 216–221.
- Mounce, S. R., Shepherd, W. J., Boxall, J. B., Kiril IV, H., and Boyle, J. H. (2021). *Autonomous robotics for water and sewer networks*.
- Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Trans. Robotics* 31, 1147–1163. doi:10.1109/ro.2015.2463671
- Nguyen, T. L., Blight, A., Pickering, A., Jackson-Mills, G., Barber, A., Boyle, J. H., et al. (2022). Autonomous control for miniaturized mobile robots in unknown pipe networks. *Front. Robotics AI* 9, 997415. doi:10.3389/frobt.2022.997415
- Nistér, D., Naroditsky, O., and Bergen, J. (2004). "Visual odometry," in *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition (IEEE)*.

## Acknowledgments

The authors thank Rob Worley for useful discussions.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). "ROS: an open-source robot operating system," in *ICRA workshop on open source software* Kobe, Japan, 5.
- Rugg-Gunn, D., and Aitken, J. M. (2022). "Investigating scene visibility estimation within orb-slam3," in *Towards autonomous robotic systems: 23rd annual conference, TAROS 2022, culham, UK, september 7–9, 2022, proceedings* (Springer), 155–165.
- Siqueira, E., Azzolin, R., Botelho, S., and Oliveira, V. (2016). "Sensors data fusion to navigate inside pipe using kalman filter," in *2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA)* (IEEE), 1–5.
- Wang, L., Zhang, Y., and Wang, J. (2017). Map-based localization method for autonomous vehicles using 3D-LIDAR \* \*this work is supported in part by the national natural science foundation of China under grant No. 61473209. *IFAC-PapersOnLine* 50, 276–281. doi:10.1016/j.ifacol.2017.08.046
- Wang, Z., Tan, J., and Sun, Z. (2015). Error factor and mathematical model of positioning with odometer wheel. *Adv. Mech. Eng.* 7, 305981. doi:10.1155/2014/305981
- Worley, R., Yu, Y., and Anderson, S. (2020). "Acoustic echo-localization for pipe inspection robots," in *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)* (IEEE), 160–165.
- Wu, D., Chatzigeorgiou, D., Youcef-Toumi, K., Ben-Mansour, R. (2015). Node localization in robotic sensor networks for pipeline inspection. *IEEE Trans. Industrial Inf.* 12, 809–819. doi:10.1109/tii.2015.2469636
- Wu, Y., Mittmann, E., Winston, C., and Youcef-Toumi, K. (2019). "A practical minimalism approach to in-pipe robot localization," in *2019 American control conference (ACC)* (IEEE), 3180–3187.