

Article

# An Intelligent Inspection Robot for Underground Cable Trenches Based on Adaptive 2D-SLAM

Zhiwei Jia \*, Haohui Liu, Haoliang Zheng, Shaosheng Fan and Zheng Liu

Hunan Province Key Laboratory of Electric Power Robot, School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410205, China

\* Correspondence: jiayege@csust.com

**Abstract:** With the rapid growth of underground cable trenches, the corresponding inspections become a heavy burden, and an intelligent inspection robot for automatic examinations in underground cable trenches would be a suitable solution. To achieve this, this paper establishes one new navigation methodology for intelligent inspection robots, especially when applied in complex scenarios and the corresponding hardware. Firstly, to map the underground trenches with higher precision, an improved graph optimization cartographer-SLAM algorithm is proposed, which is based on the combination of depth camera and LIDAR. The depth image is converted into pseudo laser data, and fused with LIDAR data for calibration. Secondly, to overcome the low precision of the Laser odometer due to the uneven ground, an adaptive keyframe selection method is designed. Thirdly, the forward A\* model is presented, which has been adjusted in three aspects, including the convergence of node searching, the cost function, and the path smoothness, to adapt to the narrow underground environment for global path planning. Fourthly, to realize dynamic obstacle avoidance, an improved fusion scheme is built to integrate the proposed global path planning algorithm and the dynamic window approach (DWA). In the case study, the simulation experiments showed the advantage of the forward A\* algorithm over the state-of-the-art algorithm in both time consumption and the number of inflection points generated, the field tests illustrated the effect of the fusion of depth camera images and LIDAR. Hence, the feasibility of this navigation methodology can be verified, and the average length of path and time consumption decreased by 6.5% and 17.8%, respectively, compared with the traditional methods.

**Keywords:** inspection robot; underground cable trench; inspection robot; Simultaneous Localization and Mapping (SLAM); navigation; LIDAR; forward A\* algorithm; path planning; dynamic window approach (DWA)



**Citation:** Jia, Z.; Liu, H.; Zheng, H.; Fan, S.; Liu, Z. An Intelligent Inspection Robot for Underground Cable Trenches Based on Adaptive 2D-SLAM. *Machines* **2022**, *10*, 1011. <https://doi.org/10.3390/machines10111011>

Academic Editors: Qing Gao, Michael V. Basin and Yu Pan

Received: 19 September 2022

Accepted: 30 October 2022

Published: 1 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Compared with overhead cable, underground cable is safer, easier to maintain, and better able to satisfy the demands of urban planning. As time goes on, underground cable monitoring will require more and more applications as it continues to become a very important part of urban distribution networks [1]. The growing length of underground cables requires more and more routine maintenance. In addition, the environment of most underground cable trenches is relatively treacherous [2]. The trenches are narrow and enclosed and are surrounded by high-voltage cables. They can be dangerous and difficult for workers to access. The risk of fire or explosion increases with time due to accumulating harmful gas and aging cables within the trenches. Narrow spaces make them hard to access, especially for workers equipped with multiple inspection tools. These aspects make manual inspection dangerous, time-consuming, and inefficient.

To precisely and timely assess the conditions of the underground cable trench, gas concentration, temperature, and humidity, the in-site and real-time pictures or videos should be collected [3]. A cable trench monitoring system is a pioneering trend. It can

obtain all kinds of information in the cable trench in real-time, and report when and where a fire is found or could happen. Unfortunately, owing to the long length of the cable trench, the use of fixed monitor systems has its downsides: blind spots, and high construction costs. Both of these make this system hard to implement in the great lengths of existing underground cable trenches [4]. Therefore, an intelligent inspection robot for underground cable trenches is of great significance for routine patrol inspection. The use of intelligent inspection robots will avoid the dangers of manual inspection and the work intensity will be greatly reduced [5,6].

The orbital robot [7,8] has been shown to be more suitable for underground pipe galleries with a relatively wide area. Although the speed of the ground walking robot is relatively low, it is suitable for underground cable trenches with narrow spaces. There are already some reports about such robots. A remote-controlled cable trench monitoring robot was designed by Shanghai Jiaotong University, but the remote-control mode is inconvenient for such a narrow cable trench environment and is unsuitable for routine inspections [7]. Even though inspection robots for substations are becoming more robust, they are relatively large and their mapping and positioning methods are not very reliable in the cable trench environment. An inspection robot for the underground cable trench could perform the inspection function according to the given route in a standard environment, but the current mapping methods contain large errors and the robot by itself cannot navigate in such complex situations [8].

Automatic navigation is one of the key technologies needed for this robot. As shown in Figure 1, the main challenge of navigation is the complicated environment within the trench. The typical size of an underground cable trench is about  $100\text{ m} \times 1.2\text{ m} \times 1.5\text{ m}$ , cable brackets are installed on both sides. Multiple cables are placed on the brackets. The surfaces of the trench are composed of sand and pipes and might even contain construction waste and bodies of animals). The trenches are composed of up-and-down slopes and pits, which makes odometer readings unreliable and inaccurate.



**Figure 1.** The real environment of the underground cable trench.

To inspect within such complex environments, an intelligent inspection robot is designed and a relevant navigation methodology are proposed. The main contributions are outlined as:

1. The hardware of an inspection robot is designed according to the demands of the underground cable trench inspection works.
2. An adaptive 2D-SLAM method is established for better localization and more accurate mapping within such structured environments of uneven ground.
3. A path planning method for the underground cable trench is proposed.

The remaining sections of this paper are outlined as follows. Section 2 presents the related work. Section 3 presents the methods used including system demands, system overview, simultaneous localization and mapping, and path planning. Section 4 describes the results and analysis. Section 5 presents the conclusions.

## 2. Related Work

With the continuous development of Simultaneous Localization and Mapping (SLAM) research, various sensors are widely used in different scenarios, owing to their specific characteristics. Of these sensors, the laser sensor is the most mature one. It obtains environmental information accurately and quickly and also meets real-time requirements with low computational complexity. Therefore, it is widely used to create traditional two-dimensional raster maps. LIDAR data and IMU (Inertial measurement unit) data could be used for self-localization and mapping in the indoor environment. However, single LIDAR cannot deal with the complex situations found within the cable trenches, especially hanging cables and cable brackets, because the detection area of single LIDAR is planar instead of 3D.

Due to the overreliance on odometer readings and poor large-scale mapping, the conventional SLAM algorithm based on filters [9,10] cannot correct the pose estimation and allows errors to accumulate. Cartographer SLAM algorithm [11] based on graph optimization is adapted because it can get a complete estimation of global trajectory and map instead of the local pose and avoid the mapping failures led by the accumulated errors of odometer readings. However, considering the cost and production efficiency, the Cartographer SLAM algorithm cannot play to its strengths within an underground cable trench.

Therefore, the use of multi-sensor data fusion in such scenarios is a prerequisite for mapping and localization. Vision sensors, such as binocular cameras and depth cameras, are another way to detect the environment. Compared with single-line LIDAR, it can obtain rich environmental information for SLAM. For example, ORB-SLAM2 of RaulMur Artaf can use monocular and binocular cameras and depth cameras to create sparse point cloud maps of the environment [12]. Felix Endres, et al. proposed that RGB-D-SLAM based on Kinect makes full use of the information of depth cameras to create environmental maps. Therefore, the fusion of laser sensor information and depth camera information is a feasible way to create a three-dimensional raster map of the environment.

SLAM technology based on LIDAR is a relatively mature positioning scheme. It is applied to inspection robots in long-channel environments. Most of them use the 2D-SLAM method for map construction and positioning. There are two main ways to realize this method: filtering method and graph optimization. The main representative of filtering-based 2D-SLAM is GMapping-SLAM [13]. This algorithm relies too much on particle confidence and no loopback detection, and it is easy to cause map dislocation when building larger maps. The main representative of 2D-SLAM based on graph optimization is Cartographer-SLAM. Graph optimization introduces the idea of loop detection, which optimizes the entire pose of the robot by optimizing the poses of all keyframes of the state quantity, and solved the problem of large scene mapping and positioning. However, in a heavily structured environment, if only laser registration is used to obtain state estimation, the 2D-SLAM based on graph optimization has problems, such as scene degradation and laser odometry [14]. Therefore, using a variety of sensors for fusion has become a necessary option for accurate mapping and positioning.

When fusing multi-sensor data, loose-coupling or tight-coupling methods can be used. At present, 2D-SLAM multi-sensor fusion mainly adopts a loose-coupling mode,

for example, Cartographer-SLAM [10] uses a lossless Kalman filter to fuse multi-source data to obtain the pose of the robot. The loose-coupling method of data is simple and has high real-time performance, but it is prone to problems, such as cumulative error and poor robustness. The tight-coupling method fuses the states of multiple sensors to construct the motion equation and observation equation. Even without loopback, the constructed map and positioning accuracy is high [14]. For example, the VINS-mono algorithm uses vision/IMU pre-integration for tight-coupling [15], and the LIO-mapping algorithm uses 3D LIDAR/IMU prediction score for tight-coupling. There are no reports on the application of tight-coupling to 2D-SLAM. Considering that it is difficult to use loop closure detection for long-structured scenes, a method is proposed to construct high-precision 2D maps by tightly coupling 2D radar and IMU pre-integration.

Robot path planning is used to plan a collision-free path movement to a specific destination in the constructed static or dynamic obstacle maps [16]. The artificial potential field (APF) method [10] uses a potential field function to cover the whole map and plan the robot's path at once. However, for more complex environments, the destination is often unreachable due to the local minimum. Sampling-based algorithms, such as probability roadmap (PRM) [17] and rapidly exploring random trees (RRT) [18], are effective methods to solve high-dimensional and multi-constrained path-planning problems. However, they only provide weak completeness (also known as probabilistic completeness), and have more inflection points. Without considering the path cost, the resulting paths generated are not optimal. Ant colony algorithm [19] has the advantages of a large amount of calculation, slow convergence, and ease of falling into the local optimal solution.

The Dijkstra algorithm uses traversal search. With more points, the node network becomes very large, and the algorithm is inefficient [20]. Based on the Dijkstra algorithm, the A\* algorithm introduces the estimated cost from the current point to the target point and determines the path search direction according to the estimated cost. This improves the efficiency of the algorithm [21]. The A\* algorithm can quickly determine the collision-free shortest global path of the mobile robot in the known environment. However, the node search strategy of the improved A\* algorithm is an eight-neighborhood search. This leads to the planning of the navigation path of a larger map. Low efficiency and more inflection points are not conducive to the normal movement of the robot. This is especially true within cable trenches, where the road surface is covered with sand, has more pits, and up and down slopes.

The dynamic window approach (DWA) [22,23] has a good obstacle avoidance ability by combining sensor data for online real-time local path planning and can give the optimal forward speed as well as the angle of the actual robot. It is suitable for robot path planning in a dynamic environment. Unfortunately, the low speed (even in simple environments) leads to a lack of timeliness for global planning.

Taking the special underground cable trench environment and time consumption into consideration, a novel path-planning method, combining the local path-planning algorithm with the global path-planning algorithm should be researched. This research should not only consider the obstacles in the environment but also ensure the completeness and accuracy of real-time path planning and meet the motion constraints of the robot.

### 3. Methods

#### 3.1. System Demands

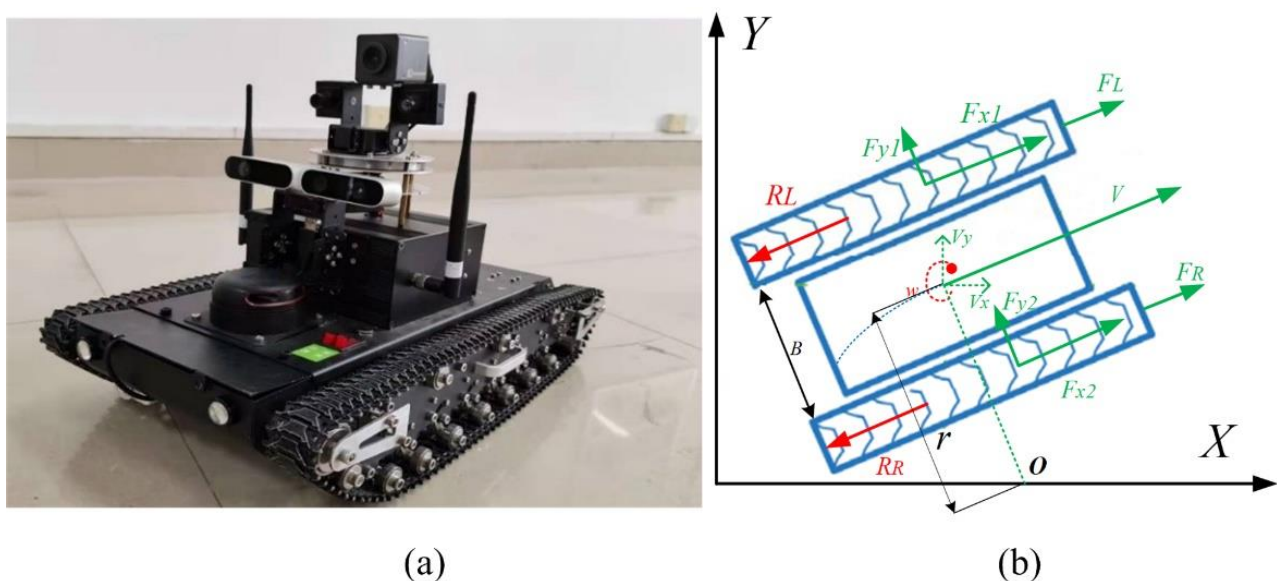
Considering the narrow and irregular passible areas in the underground cable trench (as shown in Figure 1), the moving mechanisms should fit in the size and the obstacle (the max ramp angle should be large than 30°). In addition, many tasks need to be done by the robot, in addition to evaluating the underground cable trench's overall internal environment. There are four modules needed to be added to the robot, due to the tasks and environmental features.

- (i) The visual module. To observe and detect abnormalities of the underground cable trench by infrared thermal imaging and a high-definition camera.

- (ii) The information detection module. To detect harmful gases, temperature, and humidity by multi-sensor.
- (iii) The automatic navigation module. Perform SLAM and plan inspection path.
- (iv) The assessment module. To assess the underground cable trench environment by the diagnosis system based on leading-edge intelligence.

### 3.2. System Overview

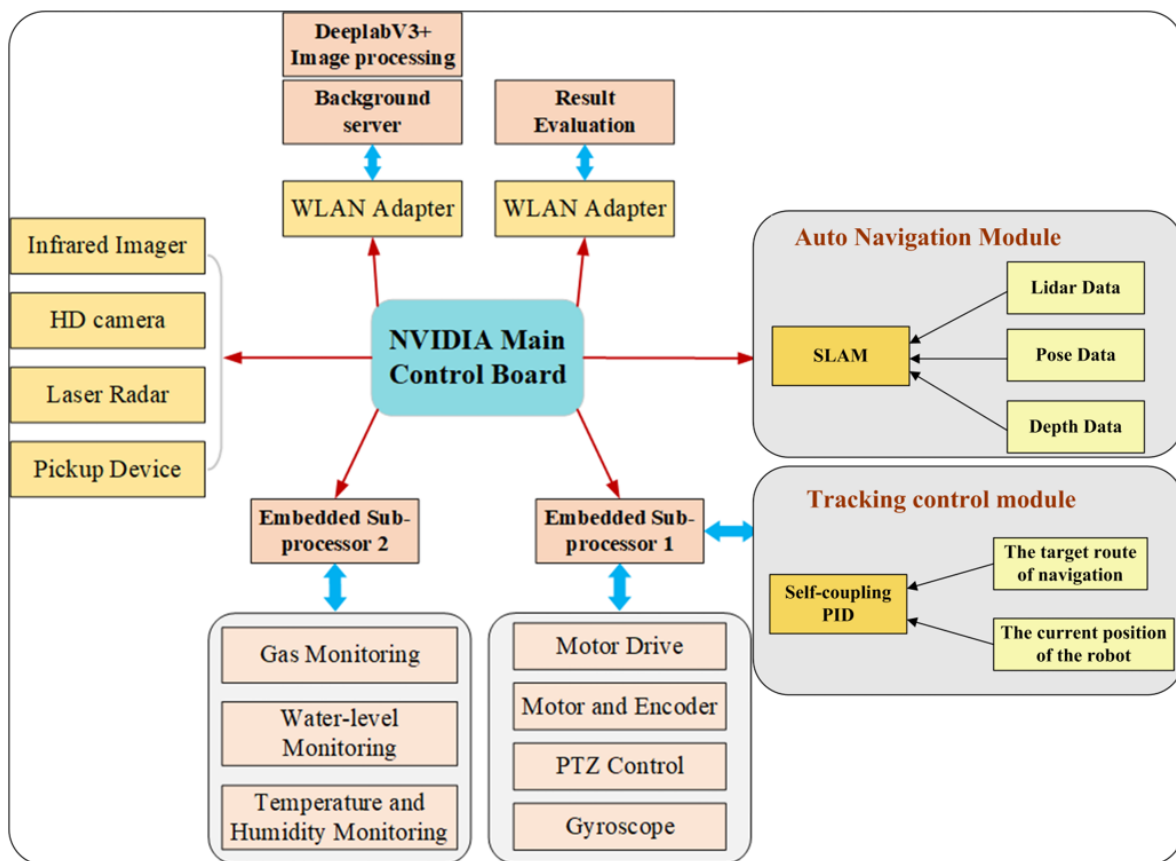
To meet the special demands of the underground cable trench environment, the robot is designed as a double crawl-type drive as shown in Figure 2a. The double drive mode adapts the crawler to connect the front and rear wheels for driving, which has some advantages, such as low consumption, small size, and convenient control. The actual parameters of the robot are as follows: length, width, and height: 500 mm  $\times$  300 mm  $\times$  120 mm. Chassis dead weight: 8.45 kg; maximum load: 30 kg; maximum climbing angle: 35°; maximum obstacle crossing: 90 mm; maximum span: 250 mm. The corresponding kinematics model of the robot is shown in Figure 2b [24]. The motion mechanism is a dual-motor caterpillar chassis, which is driven independently by the brushless DC motors on both sides to drive the driving wheels on both sides. There is no mechanical steering mechanism between the driving wheels on both sides. The dual-motor caterpillar chassis system is tuned by controlling the speed and torque of the motors on both sides through an electronic differential steering system, and an incremental PID control method is designed to track the track provided by the navigation system.



**Figure 2.** (a) The robot; (b) the kinematics model of the robot.

To design the system structure as shown in Figure 3 of the underground cable trench robot that can meet the requirements of multi-sensor information collection and cooperative multitasking, multi-CPU collaborative work architecture is used to improve the ability of data processing. The information on location and mapping is provided by LIDAR. The core information computing module uses NVIDIA core main control board and ROS, which can provide a system platform for achieving autonomous navigation algorithms. The NVIDIA main control board connects an infrared imager, HD camera, and other visual sensors to realize high-definition images and infrared thermal imaging information reading. The video data stream is uploaded to the operator terminal via WLAN. In addition, it also contains two embedded sub-processors. One is used for determining the attitude and platform control of the robot. Another is used to perceive the environment of the underground cable trench, which utilizes several sensors for monitoring temperature and humidity, and gas and water levels.





**Figure 3.** System structure of intelligent inspection robot for cable trench.

To show and evaluate the collected information, the relevant sensor information was collected, analyzed, and communicated over WLAN to the operating terminal and background server. Dual wireless network cards and high-power base stations are combined to transmit a large quantity of data, boost wireless transmission power, and guarantee the stability and reliability of communication to ensure the stability and reliability of data transmission.

### 3.3. Simultaneous Localization and Mapping

The overall framework of the mapping and positioning system proposed in this paper is mainly composed of data preprocessing, front-end odometer, map construction, and LIDAR/IMU tight coupling optimization.

The depth camera and the laser sensor are used to jointly calibrate the position.

In the data preprocessing part, the attitude is obtained by IMU integration to perform adaptive inter-frame registration selection and angle de-distortion of the laser point. The front-end odometer part registers the laser points at successive moments obtained after screening to obtain laser odometer information. In the map construction process, the current laser frame whose state estimation update exceeds 0.2 m and satisfies the pose similar to the starting pose of the map is used as a keyframe to register with the global map and added to the global map as the registration object of the next frame keyframe. The tightly coupled optimization part of LIDAR/IMU constructs a global residual block system. Each residual block inserts the relative pose between adjacent keyframes as a measurement item and the pre-integration between adjacent keyframes as an estimation item.

The depth camera and the laser sensor are used to jointly calibrate the position based on the data collected and to create a two-dimensional raster map of the environment. The fusion scheme is shown in Figure 4.

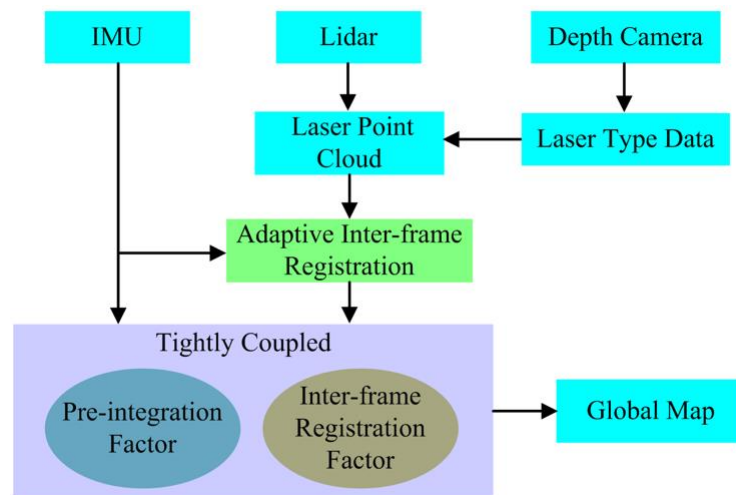


Figure 4. The scheme of simultaneous localization and mapping.

Defining the effective range of depth images is the first step to utilizing the depth information of depth images. The pixels generated, which are located higher than the top of the robot or lower than the maximum height of the ability to surmount the vertical obstacle of the robot, are discarded since they are not obstacles to the movement of the robot. As shown in Figure 5, avoiding too many inflection points in the route, the detection distance is the minimum turning radius of the robot ( $r$ ). Projecting the depth image onto the plane  $x = d + r$ , the whole height of this projection is  $h_4$ , which could be calculated as:

$$h_4 = h_1 + (d + r) \times \tan\left(\alpha + \theta - \frac{\pi}{2}\right) \tag{1}$$

$$\alpha = \text{atan}\left(\frac{d + r}{h_1}\right) \tag{2}$$

where:  $h_1$  is the height of the camera,  $d$  is the distance from the camera to the front of the robot,  $\theta$  is the sum of up and down shooting angles of the depth camera.  $h_2$  is the upper limit of the perpendicular obstacle capability of the robot,  $h_3$  is the projection that is out of the reach of the robot and could be calculated as:

$$h_3 = h_4 - h_1 - \tan(\beta) \times (d + r) \tag{3}$$

where:  $\beta$  is the real-time elevation of the robot provided by the gyroscope.

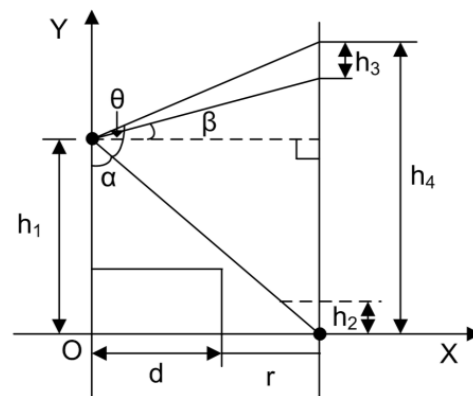
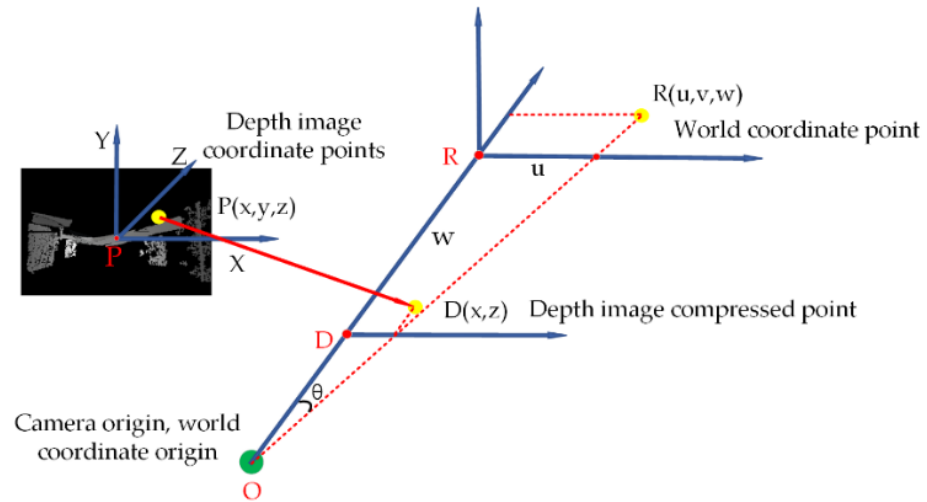


Figure 5. Available detection range of depth camera.

Since the resolution of the depth map obtained is  $240 \times 320$ , the section of the available lines are  $\left[\left[240 \times \frac{h_2}{h_4}\right], \left[240 \times \frac{h_4 - h_3}{h_4}\right]\right]$ .

As shown in Figure 6, any point  $P$  in-depth image could be expressed as  $P(x, y, z)$ , where  $y$  is the line number,  $x$  is the column number,  $y$  is the line number and  $z$  is the distance between the nearest obstacle in this direction and the camera. When converting a 3D depth image into a 2D grid map, all the points are projected to the ground, and the nearest points to the camera in every column turn out to be the nearest obstacle in the direction of projection.



**Figure 6.** Conversion of depth map to a laser point.

The distance between the camera and the obstacle in the 2D grid is  $D_{(x,z)}$ , the direction of which is from the camera to the projection of column  $x$ .  $D_{(x,z)}$  could be expressed as follows:

$$D_{(x,z)} = \min_{st. y \in [240 \times \frac{h_2}{h_4}, 240 \times \frac{h_4 - h_3}{h_4}]} z, x \in [0, 320] \quad (4)$$

As shown in formula (5), after compressing, depth image points  $D_{(x,z)}$  are mapped to the points  $R_{(u,v,w)}$  in the depth camera coordinate system.

$$\begin{cases} u = h * x * \frac{du}{f} \\ v = 0 \\ w = z \end{cases} \quad (5)$$

where:  $(u, v, w)$  represents the three-dimensional coordinate point in the world coordinate system, the depth image has been compressed into two-dimensional data  $d_{(x,z)}$ .

Therefore,  $v$  is always 0, and  $f$  is the internal parameter of the camera.

Formula (6) illustrates the processing of this mapping, and  $\theta$  is calculated as follows:

$$\theta = \text{atan}(u/w) \quad (6)$$

when  $\theta$  is mapped into the corresponding laser data slot, the rotation range of the LIDAR is  $[0, 360^\circ]$ , and the laser resolution is 0.5, which means, the laser beam is subdivided into 720 laser beams.

The laser data converted from the depth image represented as array `laser_depth [N]`, and the index value  $N$ , projected from point  $R(u, v, w)$ , is calculated as follows:

$$N = 720 \times \theta / 360 = 2 \text{atan}(u/w) \quad (7)$$

The distance of the corresponding laser data slot is  $OR$ , where  $R$  is the point that  $R(u, v, w)$  projects to  $U$  Axis, and  $O$  is the camera origin.  $OR$  could be calculated as:

$$\text{laser\_depth}[N] = OR = \sqrt{u^2 + w^2} \quad (8)$$



Laser data and LIDAR data are both obtained based on the position of the sensors instead of the position of the robot. As a result, when the system is established with the center of the robot chassis as the origin, the coordinates relative to the LIDAR as well as the camera should be transformed to the robot one.

After coordinate transformation, the laser\_depth [N] is obtained from the depth image relative to the center of the robot, and the laser [N] of the LIDAR is obtained. If the laser data obtained by the LIDAR of the current angle minus the depth image is more than 5 cm, the laser [N] of the current index is replaced by laser\_depth [N]:

$$laser[N] = laser\_depth[N] \quad (laser[N] - laser\_depth[N] > 5(\text{cm})) \quad (9)$$

Owing to the lack of loopback constraint for mapping in a long channel environment, high accuracy of the laser odometry is demanded. Laser odometry is obtained by the registration form frame-to-model or frame-to-frame. When the robot is moving on rough ground, the lidar may vibrate violently. On the one hand, there may be a large number of laser spots hitting the ground, leading to a significant increment in the inter-frame error and the obvious effects on the accuracy of the odometer. On the other hand, even if a bad situation, such as a laser spot hitting the ground, does not happen, the violent jitter of the LIDAR will make the actual overlap of the obtained laser frame and the keyframe very negligible. Although the long channel environment can make the inter-frame registration converge to obtain the pose structure as well, the two-dimensional map constructed by the pose, in this case, has a large deviation from the actual environment, resulting in the reduction in the overall map length, more map ghosting, etc. As a result, choosing a suitable keyframe, which contains enough new information and no significant deviation in pitch angle from the former keyframe, is very critical for high-precision mapping in such an environment. To select a keyframe correctly, an adaptive keyframe selection method is designed.

The adoptive keyframes selection method could be illustrated in the flow chart of Figure 7. Assuming the original position and gesture of the robot is  $P_0$ , the prediction position and gesture of the robot obtained by IMU integration based on the last frame of LIDAR data is  $P_I$ , and the position and gesture of the robot observed by the registration of the inter-frame of LIDAR is  $P_L$ , the current position and gesture of the robot is  $P_m$ , then the absolute angle error (AAE) between  $P_0$  and  $P_I$  could be expressed as:

$$AAE = \|\text{Log}(P_0^{-1}P_I)\| \quad (10)$$

The absolute translational error (ATE) between  $P_0$  and  $P_L$  could be expressed as:

$$ATE_{trans} = \|\text{trans}(P_0^{-1}P_L)\| \quad (11)$$

If AAE is less than a certain threshold value  $\theta$ , then this frame meets the condition of pitch angle. If  $ATE_{trans}$  is larger than a certain threshold value  $\tau$ , then this frame meets the condition of absolute translational error. When both AAE and  $ATE_{trans}$  are less than their respective threshold values, the current frame of LIDAR could be treated as the new keyframe and perform registration. In all other cases, the registration failed. Once the number of registration failures exceeds a certain number K, the optimization fails.

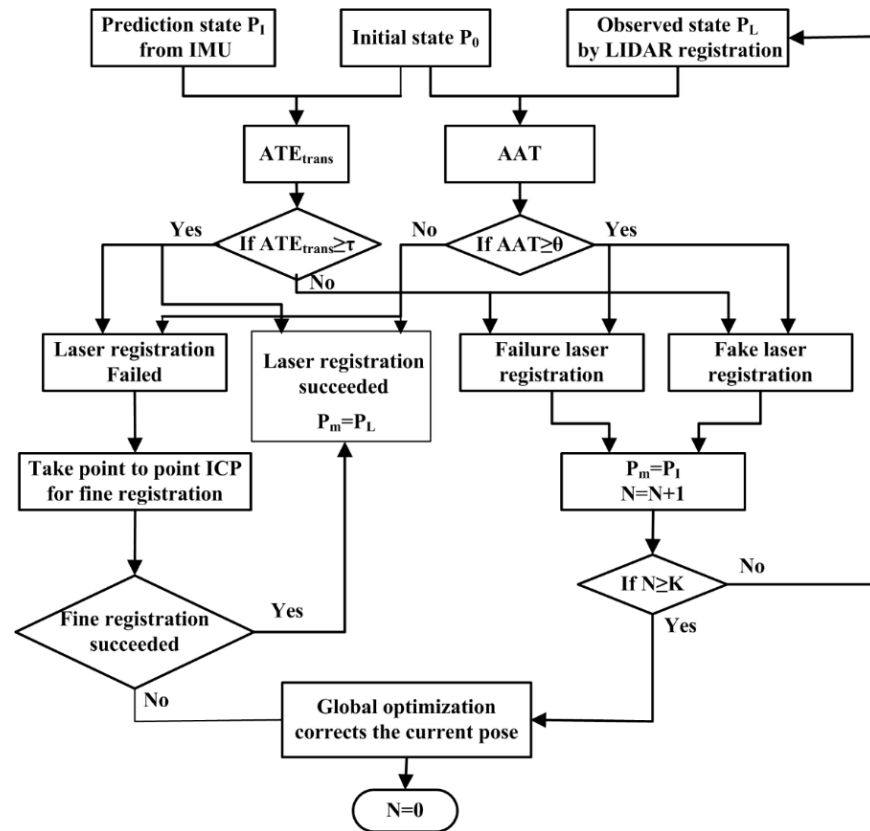


Figure 7. The flow chart of the proposed adaptive registration method.

### 3.4. Path Planning

Path planning includes two kinds of situations. One is global path planning, dealing with situations without dynamic obstacles. The other one is local path planning, dealing with situations with dynamic obstacles. The improved A\* algorithm [25] is adopted as the original model for global path planning, since it combines the breadth-first algorithm and heuristic search, and shows advantages in solving the optimal path in a static environment. The improved A\* algorithm estimates the cost function through an 8-neighborhood search. The cost function formula of the A\* algorithm is as follows:

$$F(n) = G(n) + H(n) \quad (12)$$

where:  $G(n)$  is the actual path cost of the node from the starting point to the current point,  $H(n)$  is the estimated cost of the current node to reach the endpoint, excluding whether there is an obstacle in front of,  $F(n)$  for the sum of the estimated path cost of the current node from the starting point to the endpoint. In the implementation of path planning, the A\* algorithm mainly uses two tables to expand nodes and select the best points, and records nodes through the OpenList table and CloseList table. Among them, the function of the Openlist table is to save the extended nodes encountered in the search process, sort these nodes according to the cost, take the node with the lowest value as the current node, and then add all the neighboring nodes of the current node to the Openlist table according to the neighbor node rule. However, there are too many turn points planned by the improved A\* algorithm, which makes the robot move a short distance at a time, and causes the stuttering phenomenon of the robot. Therefore, the forward A\* algorithm is proposed to be applied in the environments of the underground cable trench, and the main improvements compared with the improved A\* algorithm are as follows:

- I. The improvement in the node search scope

Taking the relative size of the cable trench and robot into consideration, backward motion is not allowed during the planning, since the general direction is definitely to move forward, eight search directions of the A\* algorithm could simplify into five, with a non-backward choice. This is the reason why we named this algorithm as the forward A\* algorithm.

The direction angle between the current and the target position is calculated as:

$$\theta_k = \arctan\left(\frac{y_{goal} - y_{pre}}{x_{goal} - x_{pre}}\right) \quad (13)$$

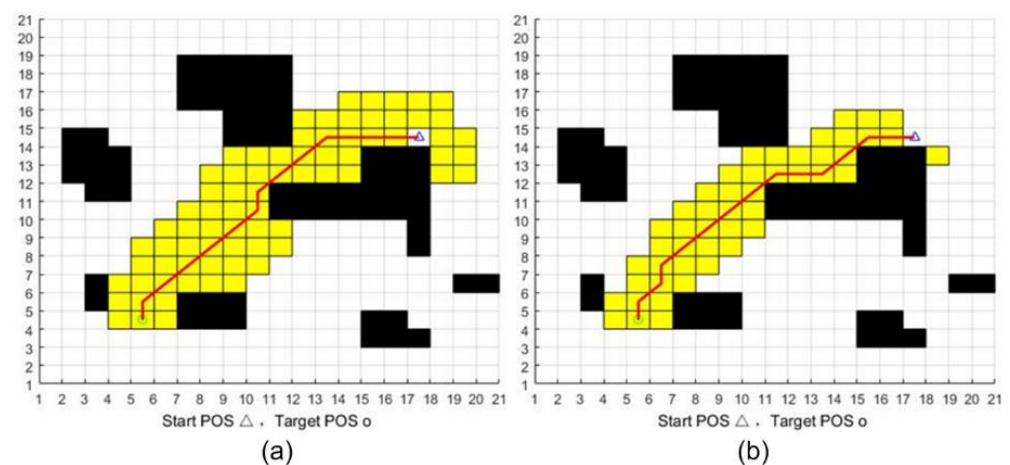
where:  $\theta_k$  is the angle between the vector from the  $k$ th node  $(x_{pre}, y_{pre})$  to the target point  $(x_{goal}, y_{goal})$  and the X-axis.

To satisfy the forward demand, these search directions, of which the absolute value of the angle between them with  $\theta_k$  bigger than  $5/4\pi$  being abandoned. Furthermore, converging faster in the direction of heuristic search, a coefficient constraint is added to the cost equation  $H(n)$ :

$$F(n) = G(n) + (1 + r_c/R)H(n) \quad (14)$$

where  $r_c$  is the distance from the current search node to the target point, and  $R$  is the distance from the starting point to the target point.

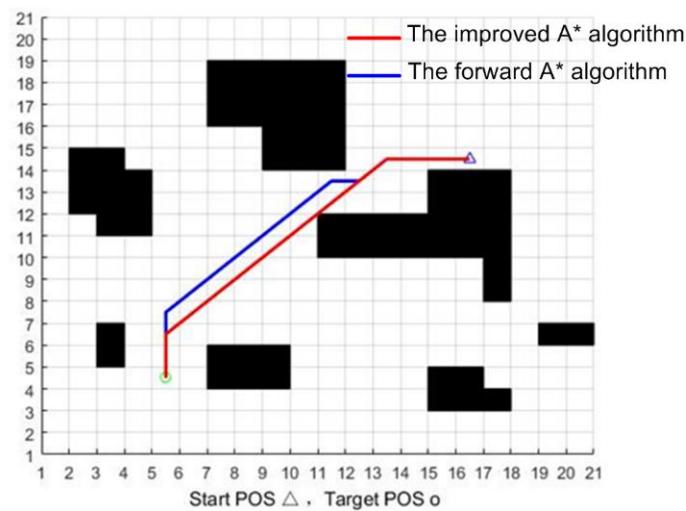
The search scope of the improved A\* algorithm and the forward A\* algorithm is illustrated in Figure 8. Compared with the improved A\* algorithm, the node search scope of the forward A\* algorithm has dropped by almost one-third, which surely raises the response speed of the robot.



**Figure 8.** (a) Search scope of the improved A\* algorithm; (b) search scope of forward A\* algorithm.

## II. The improvement in avoiding collision

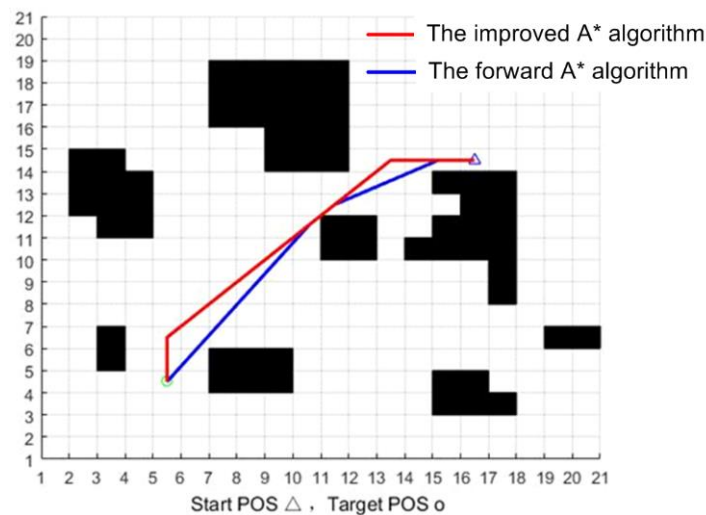
If there are three nodes with an obstacle at the edges of the Openlist, the improved A\* algorithm will treat the middle node as a redundant one and delete it, and there will be a diagonal line passing through the vertices of the obstacles, which is particularly dangerous in the real navigation process. In solving this problem, the one with the smallest cost function in the three nodes will be treated as a collision high-risk point and will be deleted in the forward A\* algorithm. In this way, there is a certain safe distance between the path planning and the obstacles. Figure 9 shows the pathway of the improved A\* algorithm (in red) and the forward A\* algorithm (in blue). That is to say, the pathway of the forward A\* algorithm avoids the risk of colliding with an obstacle in certain special situations.



**Figure 9.** The pathway when there are three nodes with an obstacle at the edges of the Openlist.

### III. Improvement in path smoothing

Too many inflection points in the path make the robot turn too much and easily slip, affecting the robot's positioning. The proposed improvement starts at the second node of the path planning. If the motion direction of the current node is the same as its parent node, delete the parent. Otherwise, change the motion destination of the parent node to the current node destination, and determine whether the straight line connected by the two points contains obstacles, if not, update the parent node to the current node, otherwise, no change should be made. Figure 10 shows the initial pathway obtained by the improved A\* algorithm (in red) and the smoothing result of the forward algorithm (in blue). By doing so, the number of turning nodes is reduced and the motion of the robot will more fluid.



**Figure 10.** The pathway before and after smoothing.

In determining the global path, the local path-planning method is used to deal with dynamic obstacles. The dynamic window approach, which is based on the forward kinematics solution of the robot, is adapted to local path planning [26]. The integration of global path planning and local path planning is closely related to the performance of path planning. As for this fusion scheme [27], after obtaining the global path and deleting those redundant points on the path, the turning point in the global path is regarded as the start point and endpoint of the local path planning. When the robot reaches a certain endpoint through local planning, this endpoint will turn into the next start point and search the way to the next endpoint until the robot reaches the target endpoint. However, the existing

fusion method, which merges the global path and the local path in the underground cable trench, may lead to the following problems:

- (i) Due to the narrow space in the trench, the local planning will lead to more inflection points during the straight section of the global path, which would destroy the smoothness of the global path and make the motion of the robot discontinuous.
- (ii) If dynamic obstacles appear in the global path, the dynamic window approach may be called several times, which makes the robot linger around the obstacles for a while and lower the inspection efficiency.

To overcome these problems, a novel fusion algorithm of global path planning and local path planning for the underground cable trench environment is proposed. The flow chart of the fusion algorithm is illustrated in Figure 11. If the current trajectory is feasible, move along the current trajectory at a constant speed and keep detecting the surrounding environment in real-time. Once a new obstacle is detected in the global path, a circle with this obstacle as the center and the length of the robot as the radius will be determined. All the global path points inside the circle will be deleted and the next nearest point on the global path will be the endpoint of local path planning. DWA local planning will be called. All the paths generated from the starting point to the endpoint will be evaluated by the evaluation function and the trajectory with the lowest cost is chosen as the best one.

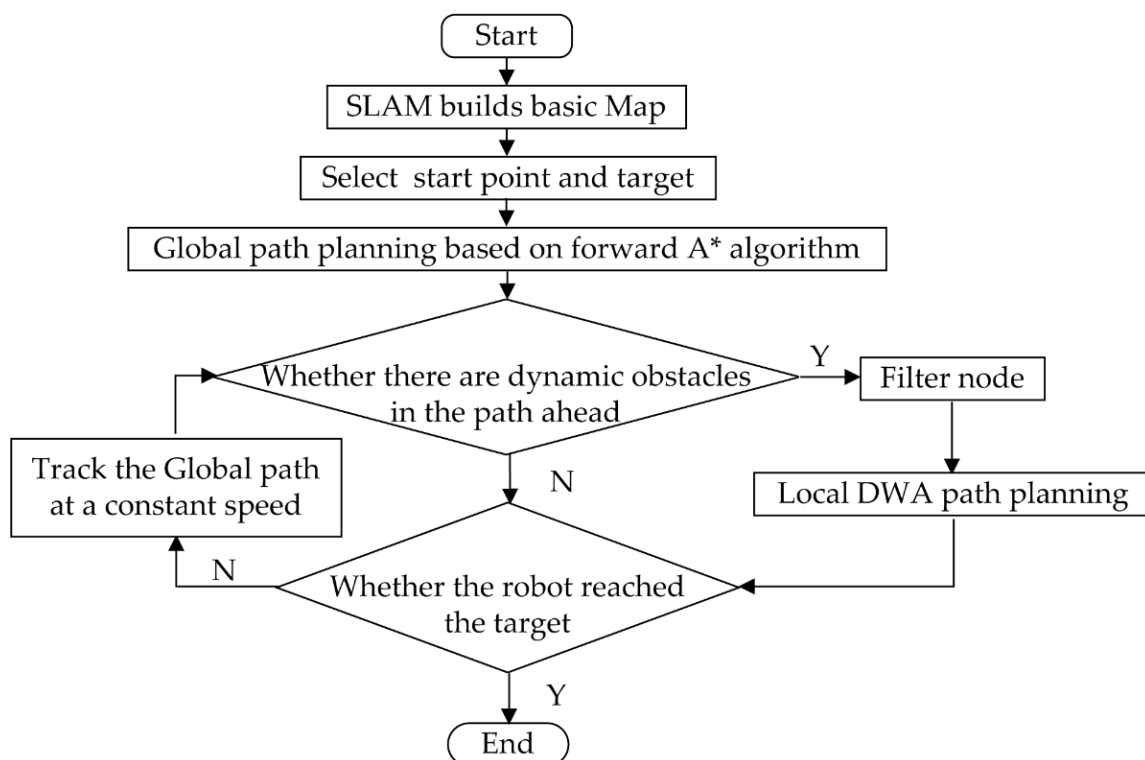


Figure 11. Multiple trajectory generation graphs of the robot.

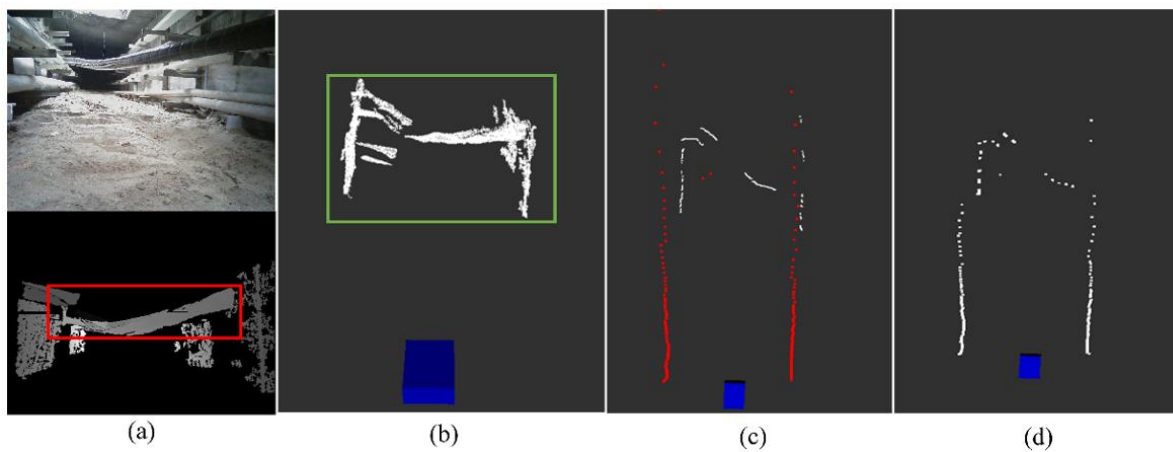
#### 4. Results and Analysis

The experiment uses Jetson Nano as the core processor, loads the Ubuntu 18.04 environment, and builds the ROS (Robot Operating System) system. The sensors are Silan-A2 single-line LIDAR, MPU6050 (gyroscope with a frequency of 200 Hz), and depth camera Astra Pro. All experiments were performed by the same robot with an angular velocity of 0.1 rad/s and a linear velocity of 0.2 m/s, and other experimental-related equipment was also the same.



### A. Experiments of the fusion of LIDAR and depth camera

The actual experimental verification is carried out in the cable trench environment in Figure 1. Figure 12 shows the process of laser data fusion in a real environment, where the top of Figure 12a is the RGB image of the underground cable trench obtained by the camera, and the bottom of Figure 12a is the corresponding depth map from the depth camera. Figure 12b is point cloud information that is converted from the depth map, Figure 12c is the comparison of the laser points (red ones) to the pseudo-laser points (white ones), and Figure 12d is the fusion results of the two-laser data. There is a wire hanging in front of the robot in Figure 12a, and the corresponding position is circled in the red box in Figure 9a. This depth information was converted into a white 3D point cloud and circled in the green box in Figure 12b. After being processed through the formula (3)–(8) projection and coordinate transformation, this depth information resulted in the white points cloud in Figure 12c. Finally, this depth information was fused with the laser data and resulted in the 2D grid map shown in Figure 12d.



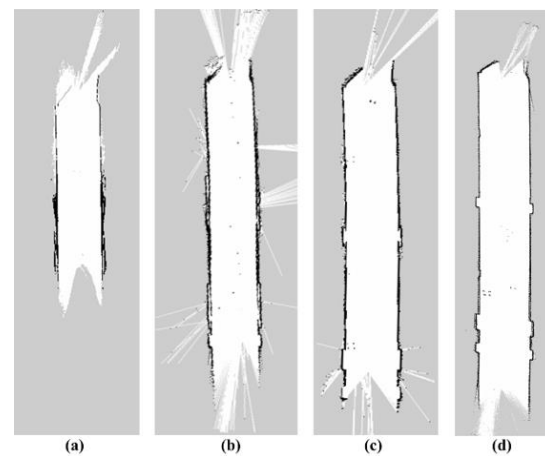
**Figure 12.** Laser Fusion of Underground (a) RGB Diagram and depth Image; (b) Conversion of point cloud information from depth map; (c) Laser points and pseudo-lasers; (d) Fused laser spot.

The robot can “see” more clearly by fusing the two-laser data. The combination of depth camera and single-line LIDAR makes up for the lack of 3D information when the robot navigates in the underground cable trench.

### B. Experiments of the adaptive registration method

To verify the effects of the proposed adaptive registration method, mapping tests are performed in the environment of Figure 12a at the same distance. There are three parameters contained in the method, the absolute translational error  $\tau$ , the pitch angle  $\theta$ , and the registration failure up limit  $K$ . Figure 13 shows the built map by the adaptive registration method with different parameters. Figure 13a is the map without the application of the proposed adaptive registration method. Since the keyframes are selected without constraints, a large number of laser spots hit the ground and make the registration fail frequently, leading to the significant shortening of the map and the disorder of the wall of the cable trench. Figure 13b is mapped when  $\theta = 15^\circ$ ,  $\tau = 0.5$  m,  $K = 20$ . Owing to the application of the adaptive registration method, the phenomenon of laser spots hitting the ground is reduced markedly, and the shortened degree of the map is decreased as well. However, the wall of the map is relatively crooked leading to false registration. Figure 13c is the built map when  $\theta = 10^\circ$ ,  $\tau = 0.3$  m,  $K = 20$ . Since the pitch angle is more limited, the disorder degree of the wall in the built map, as well as the length of the map, is improved. Figure 13d is the built map when  $\theta = 5^\circ$ ,  $\tau = 0.2$  m,  $K = 10$ . Compared with Figure 13c, attributing the success to the lower threshold value, the disorder degree of the wall in Figure 13d is further improved. However, when these threshold values are reduced

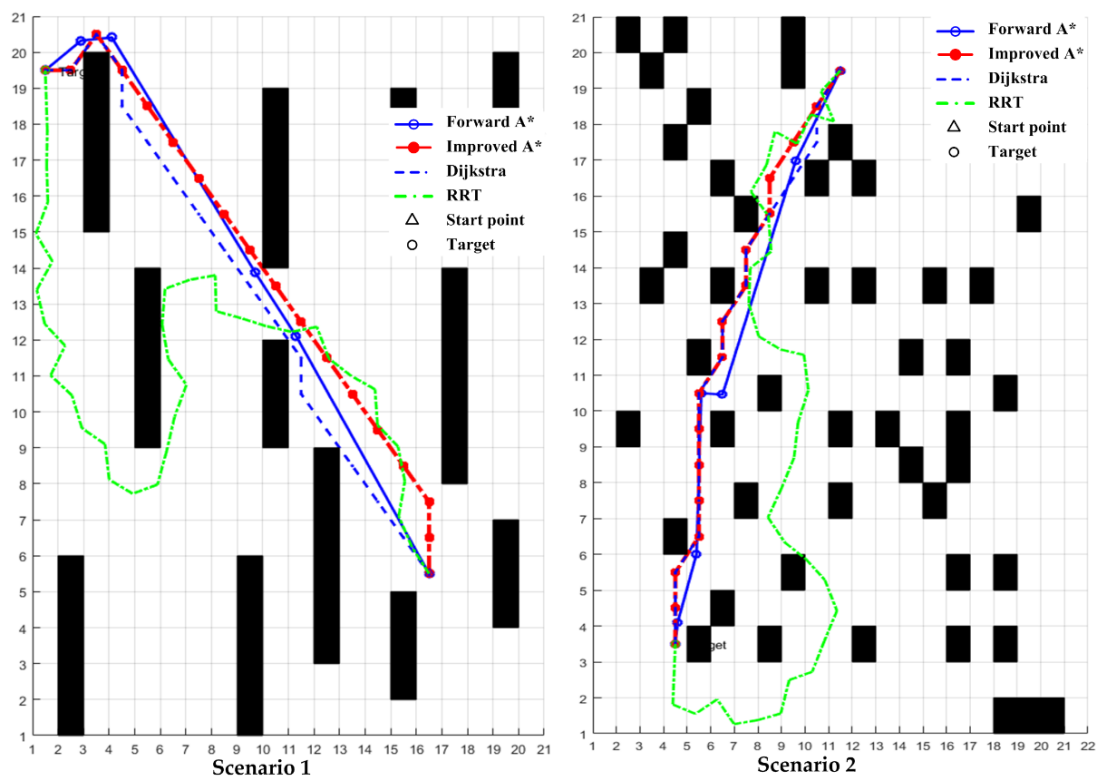
furtherly, the accuracy of the map is reduced instead, because the keyframes that satisfied all these conditions are too limited.



**Figure 13.** The map built by the adaptive registration method with different parameters. (a) map built without the adaptive registration method, (b) map built when  $\theta = 15^\circ$ ,  $\tau = 0.5$  m,  $K = 20$ , (c) map built when  $\theta = 10^\circ$ ,  $\tau = 0.3$  m,  $K = 20$ , (d) map built when  $\theta = 5^\circ$ ,  $\tau = 0.2$  m,  $K = 10$ .

### C. Experiments of our global path planning algorithm

To verify the effectiveness of our global path planning algorithm, three grid map scenes of the same size are constructed. with the same start point and endpoint. The forward A\* algorithm is performed as well as the improved A\* algorithm [27], Dijkstra [24], and RRT [26] algorithm. The simulation results are shown in Figure 14.



**Figure 14.** Simulation results of 2 scenarios.

As shown in Table 1, the computing time of the RRT algorithm is the shortest, but the RRT algorithm is composed of random tree nodes, with too many inflection points

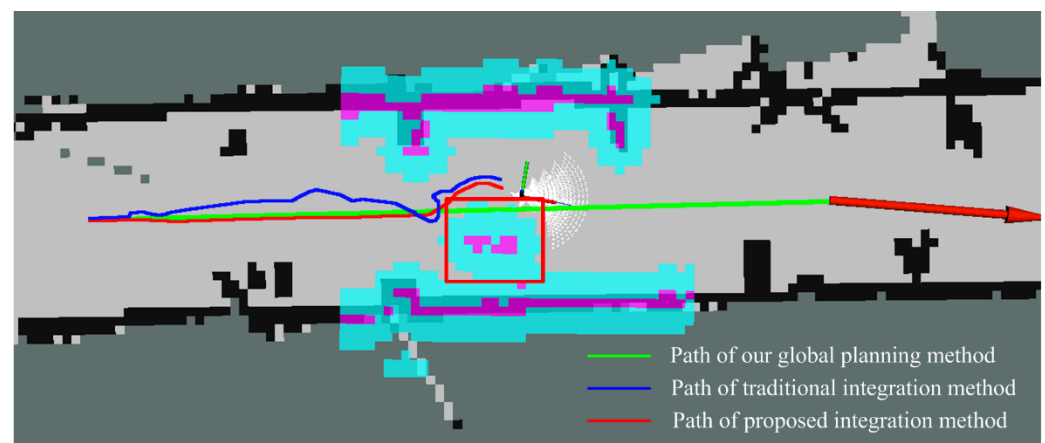
within the route. This will increase the probability of wheel slip of the robot, leading to low accuracy of positioning and even navigation failure. Compared with the Dijkstra algorithm and the improved A\* algorithm, the proposed Forward A\* method takes less time and has fewer inflection points after curve smoothing. Meanwhile, the risk of collision of the path planned by our method is lower since there is no diagonal crossing of the obstacle. Simulation results indicate that our method is safer, has better real-time performance, and has better smoothness and it is more suitable for high-risk environments such as cable trenches.

**Table 1.** Comparison of different path planning algorithms.

Algorithm	Forward A*			Improved A*			RRT			Dijkstra		
	1	2	3	1	2	3	1	2	3	1	2	3
Map scenario	3	1	2	3	1	2	3	1	2	3	1	2
Average time (ms)	11.78	12.12	12.45	13.57	13.56	15.01	6.28	6.96	8.45	14.28	14.75	15.46
Inflection point	3	3	4	5	6	5	11	9	16	7	8	6
Risk of collision	Low	low	low	high	high	high	high	high	high	high	high	high

#### D. Experiments of the fusion of global path and local path

The trajectories in the site of the underground cable trench given by the proposed global planning algorithm, traditional fusion algorithm, and proposed fusion method are illustrated in Figure 15, which are screenshots from Rviz on Ubuntu. Where the green curve, the blue curve, and the red curve are the path obtained by the proposed global planning algorithm, traditional fusion algorithm, and proposed fusion method, respectively, and the object in the red box is a dynamic obstacle. As shown, the global path provided by our global planning algorithm cannot avoid the dynamic obstacle, while the fusion path could avoid the dynamic obstacle. Furthermore, compared with the traditional fusion method (blue path), our fusion method (red path) has fewer turning points and avoids lagging around the dynamic obstacle.



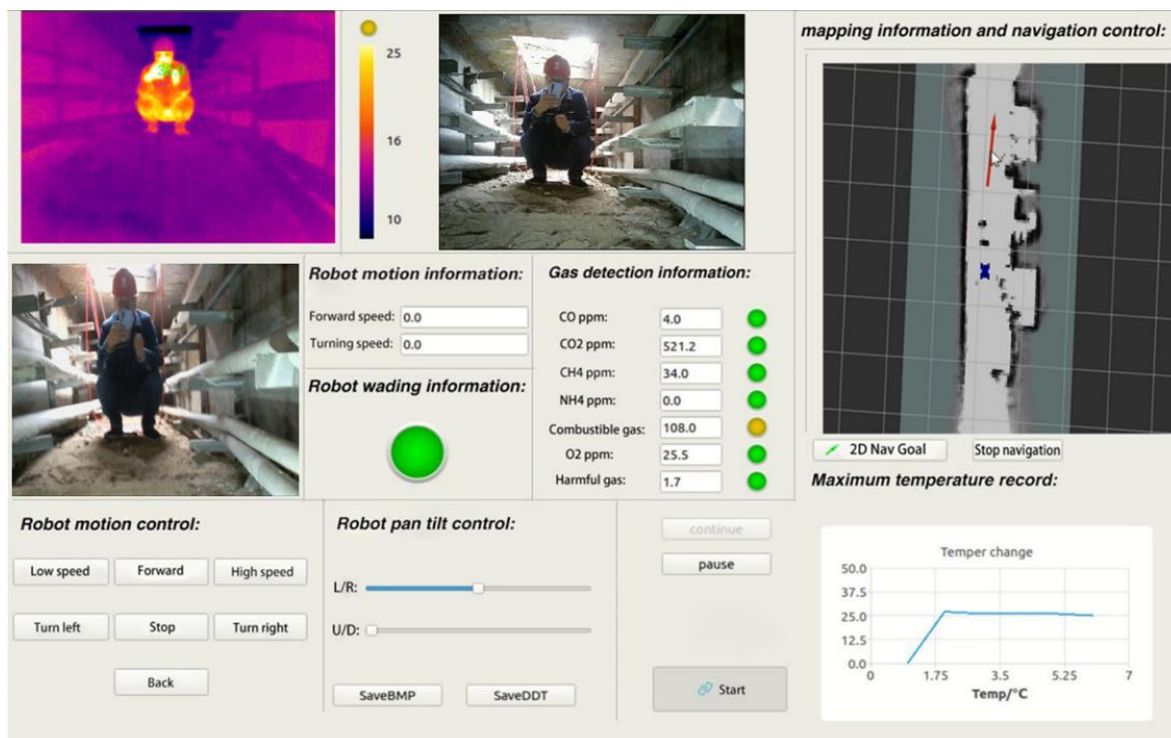
**Figure 15.** Trajectories in the site are given by different methods and shown in rviz on ubuntu.

To verify the effectiveness and operability of the proposed fusion algorithm, experiments on-site with different starting points and endpoints were done using the traditional fusion algorithm and the proposed method, respectively. Experiments (Video S1) are divided into three groups with different distances: 10 m, 20 m, and 30 m. Each group contains five experiments with different starting points and endpoints, and each experiment repeats five times. The average path length and average time consumption of each method are measured as shown in Table 2. Compared with the traditional method, the average length of path and time consumption of the proposed integration method decreases by 6.5% and 17.8%, respectively. The robot successfully arrived at the endpoint according to the planned path, and the actual tracking error is less than 10 cm, which satisfied the basic

demands of underground cable trenches inspection. The video of the robot's motion in the underground cable trench is attached in the complement. Moreover, Figure 16 illustrates the monitoring interface of the upper computer. The real-time gas concentrations and the wading information are given in the middle of the interface, the maximum temperature is recorded in the lower right, the built map is shown in the upper right, and infrared video, depth video, and high-definition video are illustrated as well. The gas concentrations and temperature of the upper computer correspond with the information on-site, which verified the feasibility of the robot.

**Table 2.** The results of experiments on site.

Algorithm	Traditional Integration Method			Proposed Integration Method		
Inspection channel length (m)	10	20	30	10	20	30
Average time (s)	79.4	164.3	240.7	61.2	131.6	197.8
Average total path length (m)	12.2	24.6	35.7	11.4	21.8	32.7



**Figure 16.** Monitoring interface of upper computer in cable trench.

## 5. Conclusions

The navigation technology of intelligent inspection robots for the underground cable trench is a bottleneck of auto patrol inspection tasks in such a complicated environment. Addressing this issue, the forward A\* algorithm is proposed for global path planning and DWA is adopted as local path planning, and an improved fusion algorithm is presented to integrate these two algorithms for underground cable trenches. Moreover, to deal with the uneven ground of the underground cable trench, an adaptive registration method for a Laser odometer is designed based on the map built by the combination of depth camera and LIDAR. Simulation results verified the superiority of our method over existing algorithms in the environment of the underground cable trench. A field test in an underground cable trench proved the effect of the mapping method combined with LIDAR and depth camera and shows the feasibility of the proposed robot.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/machines10111011/s1>, Video S1: The video of on-site experiment.

**Author Contributions:** Conceptualization, Z.J.; methodology, Z.J.; software, H.L.; validation, H.L., H.Z. and Z.L.; formal analysis, H.L.; investigation, Z.L.; resources, S.F.; data curation, H.Z.; writing—original draft preparation, Z.J.; writing—review and editing, Z.J.; visualization, H.L.; supervision, S.F.; project administration, Z.J.; funding acquisition, Z.J.. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Research Foundation of Education Bureau of Hunan Province, China (grant number 19B016, 20A028) and the Open Research Foundation of Hunan Province Key Laboratory of Electric Power Robot, China (grant number PROF1905, PROF1901).

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their very valuable comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ge, R. On the utilization of urban underground space from the construction of cable tunnel. *Shanghai Electr. Power* **2006**, *3*, 243–245. (In Chinese)
2. Khan, A.A.; Malik, N.; Al-Arainy, A.; Alghuwainem, S. A review of condition monitoring of underground power cables. In Proceedings of the International Conference on Condition Monitoring & Diagnosis, Bali, Indonesia, 23–27 September 2013.
3. Jia, Z.; Tian, Y.; Liu, Z.; Fan, S. Condition Assessment of the Cable Trench Based on an Intelligent Inspection Robot. *Front. Energy Res.* **2022**, *10*, 860461. [[CrossRef](#)]
4. Commission of the European Communities. *Background Paper under Grounding of Electricity Line in Europe*; Commission of the European Communities: Brussels, Belgium, 2003.
5. Hepburn, D.M.; Zhou, C.; Song, X.; Zhang, G.; Michel, M. Analysis of on-line power cable signals. In Proceedings of the International Conference on Condition Monitoring & Diagnosis, Beijing, China, 21–24 April 2008.
6. Deng, F. Research on Key Technology of Tunnel Inspection Robot. Master's Thesis, North China Electric Power University, Beijing, China, 2013. (In Chinese).
7. Silvaa, B.; Ferreira, R.; Gomes, S.C.; Calado, F.A.; Andrade, R.M.; Porto, M.P. On-rail solution for autonomous inspections in electrical substations. *Infrared Phys. Technol.* **2018**, *90*, 53–58. [[CrossRef](#)]
8. Xie, Z. Research on Robot Control System for Comprehensive Detection of Cable Tunnel. Ph.D. Thesis, Shanghai Jiaotong University, Shanghai, China, 2008. (In Chinese).
9. Lu, S.; Zhang, Y.; Su, J. Mobile robot for power substation inspection: A survey. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 830–847. [[CrossRef](#)]
10. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [[CrossRef](#)]
11. Sim, R.; Elinas, P.; Griffin, M.; Shyr, A.; Little, J.J. Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters. In Proceedings of the Conference on Computer & Robot Vision, Quebec, QC, Canada, 7–9 June 2006.
12. Wang, C.; Yin, L.; Zhao, Q.; Wang, W.; Li, C.; Luo, B. An Intelligent Robot for Indoor Substation Inspection. *Ind. Robot.* **2020**. ahead-of-print. [[CrossRef](#)]
13. Wu, Y.; Ding, Z. Research on laser navigation mapping and path planning of tracked mobile robot based on hector SLAM. In Proceedings of the International Conference on Intelligent Informatics & Biomedical Sciences, Bangkok, Thailand, 21–24 October 2018; pp. 210–215.
14. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB—D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
15. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3D mapping with an RGB—D camera. *IEEE Trans. Robot.* **2017**, *30*, 177–187. [[CrossRef](#)]
16. Li, S.-X. Research on 3D SLAM Based on Lidar/Camera Coupled System. Master's Thesis, PLA Strategic Support Force Information Engineering University, Zhengzhou, China, 2018.
17. Rui Huang, Y.Z. An adaptive scheme for degradation suppression in Lidar based SLAM. *Sens. Rev.* **2021**, *41*, 361–367. [[CrossRef](#)]
18. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real time loop closure in 2d lidar slam. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
19. Chou, C.-C.; Chou, C.-F. Efficient and Accurate Tightly-Coupled Visual-Lidar SLAM. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: Manhattan, NY, USA, 2022. [[CrossRef](#)]
20. LaValle, S.M.; Kuffner, J.J. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2016**, *20*, 378–400. [[CrossRef](#)]



21. Choubey, N.; Gupta, M. Analysis of working of Dijkstra and A\* to obtain optimal path. *Int. J. Comput. Sci. Manag. Res.* **2013**, *2*, 1898–1904.
22. Adel, A.; Bennaceur, H.; Châari, I.; Koubâa, A.; Alajlan, M. Relaxed Dijkstra and A\* with linear complexity for robot path planning problems in large-scale grid environments. *Methodol. Appl.* **2016**, *20*, 4149–4171.
23. Blaich, M.; Rosenfelder, M.; Schuster, M.; Bittel, O.; Reuter, J. Extended grid based collision avoidance considering COLREGs for vessels. *IFAC Proc. Vol.* **2012**, *45*, 416–421. [[CrossRef](#)]
24. Zhao, X.; Wang, Z.; Huang, C.K.; Zhao, Y.W. Mobility based on improved A\* algorithm. Robot path planning. *Robot* **2018**, *40*, 903–910. (In Chinese)
25. Chaari, I.; Koubaa, A.; Bennaceur, H.; Ammar, A.; Alajlan, M.; Youssef, H. Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 172988141666366. [[CrossRef](#)]
26. Cheng, C.; Hao, X.; Li, J.; Zhang, Z.; Sun, G. Global dynamic path planning based on fusion of improved A\*algorithm and dynamic window approach. *J. Xi'an Jiaotong Univ.* **2017**, *11*, 137–143. (In Chinese)
27. Li, Z.; Chen, L.; Zheng, Q.; Dou, X.; Yang, L. Control of a path following caterpillar robot based on a sliding mode variable structure algorithm. *Biosyst. Eng.* **2019**, *186*, 293–306. [[CrossRef](#)]