

Mapping Based on Manhattan World Constraints

HuiZhuo Xiao

Abstract—Ultrasonic sensors are widely used in mobile robot navigation due to their low cost and strong anti-interference capability. However, their inherent low resolution, wide beam angle, and sparse point cloud characteristics limit mapping accuracy. Traditional point cloud clustering and fitting-based mapping methods are easily affected by noise, making it difficult to meet the high-precision line feature mapping requirements in structured environments (such as indoor corridors and rooms). The Manhattan World Constraint (MWC) effectively reduces the impact of sensor errors by aligning environmental structures to orthogonal directions. Therefore, this paper proposes a novel MWC-integrated line feature mapping framework that enhances the robustness and accuracy of ultrasonic mapping through dynamic clustering, direction correction, and global optimization.

Index Terms—Global Optimization; Line Feature Extraction; Manhattan World Constraint (MWC); Ultrasonic Mapping.

I. INTRODUCTION

Currently, environmental perception devices used for mobile robot mapping mainly include laser sensors, vision sensors, and ultrasonic sensors. However, the performance of these sensors varies significantly across different environments. For example, although laser sensors can acquire precise point cloud information, they exhibit weak reflection signals when encountering transparent surfaces or harsh environments, leading to large measurement errors. Additionally, in indoor contour detection, the high cost of LiDAR technology limits its advantages.

Compared to LiDAR and vision sensors, ultrasonic sensors have become a commonly used method for mobile robots to detect their surroundings and obtain contour features of unknown environments due to their low cost, high stability, and ease of control [1-3]. By utilizing the time delay of sound wave propagation for distance measurement, ultrasonic sensors can effectively perform short-range positioning and obstacle detection. Compared to other sensors, ultrasonic sensors offer relatively high accuracy in small-scale environments and are less sensitive to environmental changes. Therefore, they hold strong application potential in indoor environments [4-6].

Although ultrasonic sensors have certain advantages in terms of cost and applicability, they also face the following challenges:

(a)Low Angular Resolution: Ultrasonic sensors have a wide beam angle, resulting in significantly lower angular resolution compared to LiDAR and vision sensors. This leads to greater uncertainty in measurement results [7].

(b)Sparse Point Cloud Data: The point cloud data provided

by ultrasonic sensors is relatively sparse, making it difficult to be directly used for high-precision map construction.

(c)Multipath Reflection in Corners: Ultrasonic sensors are prone to multipath reflections in corners, which can lead to the Region of Constant Depth (RCD) effect [8].

Therefore, research on environment perception based on ultrasonic sensors is an important challenge.

In response to the demand for high-precision environmental perception, ultrasonic sensors, with their mature technology, low cost, and wide applicability, have shown enormous potential and commercial value in the field of mobile robotics. Therefore, optimizing environmental perception algorithms, reducing the crosstalk of ultrasonic signals, and improving detection efficiency to meet the required environmental perception accuracy are key factors in advancing ultrasonic mapping technology.

In recent years, significant progress has been made in ultrasonic technology, with many research teams continuously optimizing the ultrasonic transmission and reception structures to improve measurement accuracy [9]. Wei Song and others developed an ultrasonic distance sensor with a small beam angle to address the false alarm issues caused by the large beam angle of ultrasonic sensors in reverse driving applications, effectively reducing measurement errors. Chen and colleagues combined composite ceramic sol-gel thin film technology with sol-gel infiltration methods to successfully produce ultrasonic transducers with frequencies as high as 300 MHz, promoting the advancement of transducer frequencies in this field [10]. Meanwhile, new sonar positioning algorithms have been proposed to further optimize positioning accuracy. Inspired by the development of computer vision and machine learning technologies, researchers have begun to integrate ultrasonic waves with deep learning, leveraging signal processing and visual constraints to optimize sonar positioning systems and improve overall perception capabilities. For example, Zhao and others used deep learning methods to analyze sonar images, effectively enhancing the network's target recognition ability; Yonghong Zhang and colleagues employed wavelet transform-based empirical mode decomposition to process ultrasonic signals, significantly improving denoising effects.

Currently, research on ultrasonic SLAM is still in a relatively immature stage. Researchers mainly focus on improving the accuracy of ultrasonic data and extracting key features. However, issues such as the sparsity of point clouds and low distance measurement accuracy of ultrasonic sensors remain unresolved.

This paper primarily studies methods for map construction based on ultrasonic sensors in indoor environments. In indoor settings, the walls and buildings typically conform to the Manhattan World Constraint (MWC), meaning that the walls and major structures in the environment are mostly

Manuscript received April 10, 2025

HuiZhuo Xiao, School of Computer Science and Technology, TianGong University, TianJin, China

parallel or perpendicular to fixed global coordinate axes. This geometric constraint has been widely applied in visual SLAM [11-13] and LiDAR-based map optimization [14]. Abadi et al. [15] first applied it to map construction based on ultrasonic sensors. This method can effectively describe the structural features of indoor environments, but the resulting maps are incomplete, lacking global map optimization, and do not integrate odometry and IMU sensors for simultaneous localization. This study optimizes the mapping results by incorporating dynamic clustering, direction correction, and global optimization into the map construction method.

Section 2 of this paper explains the three steps of the mapping algorithm. Section 3 presents the experimental results, which demonstrate that applying the MWC constraint can provide a more accurate indoor map. The experimental results are compared with the ground truth map.

II. METHODOLOGY

In the method described in this section, it is assumed that the robot's trajectory is known. A line feature-based map construction method based on the Manhattan World Constraint (MWC) is used to build the environmental map of the robot, addressing the issues of low precision, low resolution, and sparse point clouds in ultrasonic sensors.

The method consists of three main steps. In the first step, an improved DBSCAN clustering algorithm is used, with adaptive parameter adjustments to enhance the robustness of the clustering process. Then, line segment fitting is performed using RANSAC and PCA. In the second step, the MWC is applied for line segment direction correction, ensuring strict alignment with horizontal or vertical directions to reduce ultrasonic measurement errors. Finally, in the third step, global map optimization is carried out through edge and corner point classification, line segment merging, virtual intersection generation, and corner completion to improve the accuracy and completeness of the map.

A. Step1(Point Cloud Clustering and Line Segment Fitting)

The raw point clouds generated by ultrasonic sensors suffer from high sparsity and significant noise. The traditional DBSCAN algorithm, due to the fixed neighborhood radius (ϵ) and minimum points for clustering ($MinPts$), is prone to over-segmentation or missed detection. To address this issue, this section uses an improved DBSCAN algorithm that enhances the clustering robustness of sparse point clouds through dynamic parameter adjustment and local region constraints.

The traditional DBSCAN algorithm uses a fixed ϵ , making it difficult to adapt to changes in point cloud density at different distances (high density at close range, low density at long range). The improved DBSCAN algorithm dynamically calculates ϵ based on the average neighbor distance of the current scanned point set. This adjustment allows the algorithm to better handle variations in point cloud density, improving its ability to cluster sparse data effectively.

In addition, to avoid overfitting or misclassification of the

data, this paper also introduces a strategy for adaptively adjusting the $MinPts$ based on the data distribution. The dynamic $MinPts$ is set as follows:

$$MinPts = \max(3, \lfloor \beta \cdot N_{total} \rfloor) \quad (1)$$

In the formula, N_{total} represents the total number of points in a single scan, and β is the proportional coefficient. This method enforces a minimum threshold when the point cloud is sparse to ensure that $MinPts$ does not become too low, which could lead to noise points being misclassified. Meanwhile, when the data is larger, the value of $MinPts$ can reasonably reflect the distribution characteristics of the data, avoiding errors during clustering.

By incorporating the pose of the mobile robot, the clustering search range is restricted to the sector area in the current scanning direction, preventing cross-region misclustering.

As shown in Figure 1, (a) depicts the real environment being detected, while (b) and (c) compare the clustering results of this typical data frame in the ultrasonic point cloud data using the traditional DBSCAN algorithm and the improved DBSCAN algorithm, respectively.

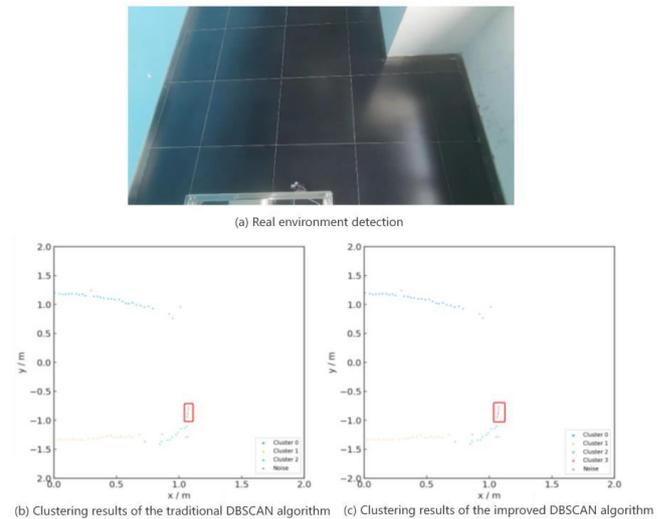


Figure 1 Comparison of clustering results between the traditional DBSCAN and the improved DBSCAN algorithms

In Figure 1, the areas highlighted by the red boxes in (b) and (c) were misclassified as noise in the traditional DBSCAN algorithm, whereas in the improved DBSCAN algorithm, they were correctly classified as point cloud data of the real environment.

To improve the quality of the clustering results, this study also applied post-processing to the clustered data. First, isolated points were removed by eliminating clusters with fewer than $MinPts$ points to reduce noise. Then, boundary points of the clusters were smoothed, and outliers deviating from the main distribution were removed through a secondary density check, thereby enhancing the accuracy of the clustering.

After completing point cloud clustering, precise line segment features need to be extracted from each local point cloud cluster. To achieve high-precision and high-robustness line segment detection, this method employs an iterative RANSAC-PCA fusion algorithm to generate candidate line segments, and estimates endpoints and merges redundant

segments through a statistical-geometric joint optimization strategy.

First, the RANSAC algorithm is used to iteratively generate candidate line segments. After obtaining a local point cloud cluster, two points are randomly sampled from the cluster to generate a line model $L : ax + by + c = 0$. The perpendicular distance d_i from all points in the cluster to the line L is calculated. If $d_i < d_{th}$, the point is considered an inlier of the line segment. d_{th} is a threshold set experimentally. The number of inliers for this line segment is then calculated and recorded.

Then, two different points are randomly sampled from the point cloud cluster—distinct from the previous pair—and the above steps are repeated. This process continues until the model with the highest number of inliers is selected. The corresponding line is retained in the candidate line segment set $L_{candidate} = \{L_j\}$, and its inliers are removed from the point cloud cluster. The process is iterated until the number of remaining points falls below $MinPts$.

Moreover, for each line segment detected and deemed suitable for inclusion in the candidate set, the PCA method is employed to verify its quality and improve its accuracy. This is done by calculating the covariance matrix of the point set belonging to each candidate line segment, from which eigenvalues and eigenvectors are obtained to assess the goodness of fit. Specifically, for each candidate line segment L_j , the covariance matrix C of the points belonging to the line segment is computed as follows:

$$C = \frac{1}{N} \sum_{i=1}^N (p_i - \mu)(p_i - \mu)^T \quad (2)$$

Here, μ represents the mean of all the points. By computing the eigenvalues λ_1 and λ_2 , if the eigenvalues of the line segment satisfy condition $\lambda_1 / (\lambda_1 + \lambda_2) \geq 0.85$, the segment is considered valid; otherwise, it is removed from the candidate line segment set. The results of the line segment fitting are shown in Figure 2.

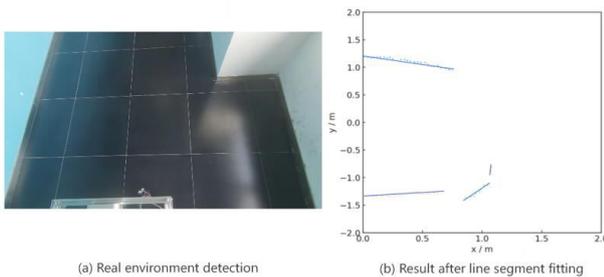


Figure 2 Line Fitting Results

The overall iterative process of the algorithm is shown in Figure 3.

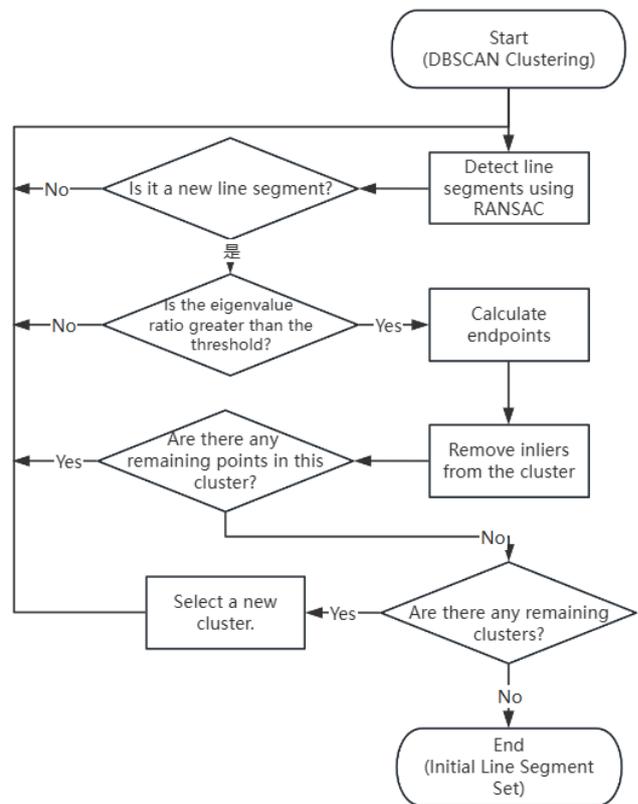


Figure 3 Iterative Flowchart of the Line Segment Fitting Algorithm

B. Step 2 (Application of Manhattan World Constraints)

The Manhattan World Constraints (MWC) establish a framework for spatial perception by aligning a structured coordinate system with the environmental geometric features, providing spatial constraints for the mobile robot. The implementation process first relies on the calibration of the initial pose: when the mobile robot starts, it must ensure that the initial heading direction is as aligned as possible with the dominant physical structures in the environment (such as wall edges or corridor orientations). The initial pose is used as the global coordinate system, with the initial heading direction defined as the X-axis of the global coordinate system. This direction is also defined as the primary segment direction, which is then used to constrain the correction direction of detected line segments. This allows alignment with the dominant structures in the environment (such as walls), thereby reducing the impact of the ultrasonic sensor's low precision and resolution on map building.

The line segment angle correction based on the primary segment direction is the core component of the Manhattan World Constraints (MWC) application. It involves rotating and adjusting the direction of the line segments to make them strictly parallel or perpendicular to the primary segment direction (i.e., the X/Y axes of the global coordinate system). This section describes in detail the generation of the correction angle, the selection of the optimal rotation strategy, and the implementation process of the midpoint-based rotation transformation.

First, the correction angle for all line segments is calculated. For each line segment L_i , its current direction angle θ_i is computed based on its endpoints. Then, two candidate correction angles are generated: the parallel

correction angle and the perpendicular correction angle. The parallel correction angle refers to rotating the line segment to make it parallel to the primary segment, with the target direction being 0° or 180° . The corresponding rotation angles are $\theta_i^{\parallel} = -\theta_i$ or $\theta_i^{\perp} = \pi - \theta_i$.

Then, to ensure the minimum geometric disturbance after the rotation, the smallest absolute rotation angle is selected from the parallel correction angle and the perpendicular correction angle as the final correction angle:

$$\hat{\theta}_i = \arg \min_{\theta \in \{\theta_i^{\parallel}, \theta_i^{\perp}\}} |\theta| \quad (3)$$

Next, the rotation operation is applied to the line segments that need correction.

To avoid cumulative errors introduced by global coordinate transformations, the rotation strategy uses the midpoint C^i of each line segment as the local reference point. First, the endpoints of the line segment are translated to a local coordinate system with C^i as the origin:

$E_{local}^i = E_{global}^i - C^i$. Then, a rotation matrix $R(\hat{\theta}_i)$ is applied to rotate the line segment $E_{rotated}^i = R(\hat{\theta}_i) \cdot E_{local}^i$. Finally, the rotated coordinates are translated back to the global coordinate system: $E_{corrected}^i = E_{rotated}^i + C^i$.

With this strategy, the direction of the line segments that need correction is adjusted to align with the Manhattan constraint direction, while keeping the midpoint of the line segment unchanged. This avoids global rotations that could cause overall map distortion.

In some complex environments, if the correction rotation angle of a line segment is large, it may be due to noise or misdetection. In this case, instead of correcting the line segment, it is temporarily moved to the corner point candidate set (see Step 3) to avoid erroneous corrections. The corrected results are shown in Figure 4.

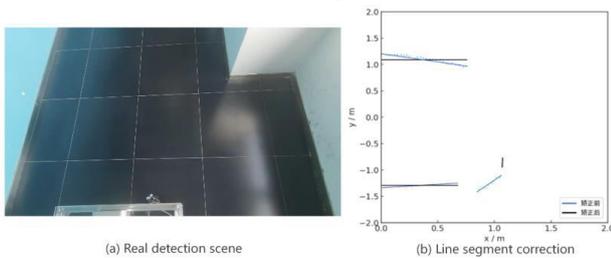


Figure 4 Results after applying Manhattan World Constraints

C. Step 3 (Global Map Optimization)

During the ultrasonic mapping process, directly using the extracted line segments to construct the map may be affected by sensor noise, the ultrasonic Region Constant Depth (RCD) effect, and data sparsity, leading to issues such as false corner points and broken line segments. Therefore, further optimization of the global map is required to improve its coherence and accuracy.

The distinction between edges and corner points is a key step in constructing a structured line-based map. Due to the ultrasonic sensor's Region Constant Depth (RCD) effect near corner points, it fails to reflect the true characteristics of corner points, instead presenting false linear features, as shown in Figure 5. The area within the red box in the figure illustrates the RCD effect. Therefore, line segments need to

be classified into environmental main structures (such as wall edges) or potential corner points (such as room corners) based on their features. Since this study is based on the Manhattan World Constraints, the line segments corresponding to the expected indoor environment structures generally have small correction angles. This section implements the classification of edges and corner points based on the correction angle threshold and geometric feature verification strategy.

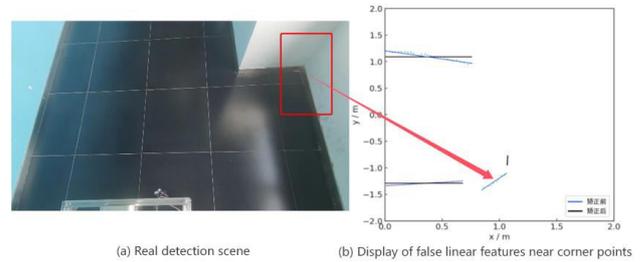


Figure 5 Constant Depth Effect Near Corner Points

For each line segment, the absolute value of its correction angle $\hat{\theta}$ reflects the magnitude of the direction adjustment. If the correction angle is small, it indicates that the original direction of the line segment is close to the Manhattan constraint axes (X/Y axes), typically corresponding to wall edges or large furniture edges. If the correction angle is large, it may be caused by the following reasons:

(a)Region Constant Depth (RCD) Effect Near Real Corner Points: The wide beam angle and low resolution of the ultrasonic sensor cause the point cloud near corners to form false linear features, leading to the RCD effect.

(b)Dynamic Occlusion or Sensor Noise.

The classification of edges and corner points is based on the following rules:

$$L_i \in \begin{cases} Edge\ Set\ E, & \text{if } |\hat{\theta}_i| \leq \theta_{th} \\ Corner\ Point\ Candidate\ Set\ C, & \text{otherwise} \end{cases} \quad (4)$$

Here, θ_{th} represents the angle threshold set for the experiment.

For the line segments that belong to the corner point candidate set, they are false linear features caused by the Region Constant Depth (RCD) effect near real corner points or noise. Therefore, these line segments need to be removed from the map. In the subsequent steps, the corner point candidate set will be used to further optimize the map.

The line segments classified as edges need to be merged into continuous segments. The goal during the line segment merging process is to combine two line segments that are spatially close and have consistent directions, thereby enhancing the continuity and compactness of the map. To achieve this goal, a set of merging conditions must be met. The specific conditions include direction consistency, projection overlap, and spatial proximity.

First, for two line segments L_1 and L_2 , it is necessary to verify whether they satisfy the following conditions:

(a)Direction Consistency: The angle θ between the two line segments must be less than 5° , meaning their directions are close.

(b)Projection Overlap: Calculate the projection overlap of the two line segments in the same direction. First, project the two line segments onto the same coordinate axis, then calculate the intersection of their projection intervals. The

projection overlap is the ratio of the length of the intersection to the maximum projection length of the two line segments.

(c) Spatial Proximity: Calculate the directional distance d_{center} between the centers of the two line segments. Specifically, when the angles of the two line segments are aligned with the x-axis, calculate the distance between the centers of the two line segments along the y-axis. When the angles of the two line segments are aligned with the y-axis, calculate the distance between the centers along the x-axis. This distance is referred to as the directional distance d_{center} between the two line segments. When $d_{center} < 0.15m$, the two line segments are considered to satisfy the spatial proximity condition.

When the two line segments meet the above conditions, the merging operation is performed. First, the Hausdorff distance d_H between the endpoints of the two line segments needs to be calculated. The Hausdorff distance is used to measure the farthest point distance between the two line segments in space. The formula for calculating the Hausdorff distance is:

$$d_H = \max \left(\begin{matrix} \max_{(x_1, y_1) \in L_1} \min_{(x_3, y_3) \in L_2} \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}, \\ \max_{(x_3, y_3) \in L_2} \min_{(x_1, y_1) \in L_1} \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \end{matrix} \right) \quad (5)$$

Use these two points as the new endpoints of the merged line segment.

To satisfy the Manhattan World Constraint, the merged line segment needs to be rotated. Based on the length ratio of the two line segments, the reference point for rotation is determined, and then the Manhattan World Constraint is applied to the merged line segment according to the method in Step 2.

Through these steps, the edge line segments that meet the conditions are merged, improving the accuracy and continuity of the map construction.

The generation and verification of virtual intersection points is the basis for further map optimization. The goal is to use geometric reasoning and point cloud feature analysis to determine whether the intersection point truly exists in the real environment, laying the foundation for constructing a coherent line-based map. The intersection points formed by the dashed lines in Figure 6(b) are virtual intersection points that exist in the real structure. To achieve this goal, intersection points need to be calculated based on non-parallel line segments, and the validity of these intersection points must be verified.

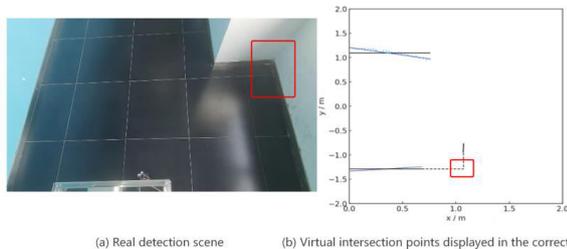


Figure 6 Virtual intersection points in the Manhattan-constrained map

For two non-parallel line segments L_i and L_j , the first step is to calculate their theoretical intersection point as a candidate virtual intersection point. First, parameterize line segment L_i , given its endpoint coordinates $E_1^i = (x_1^i, y_1^i)$ and

$E_2^i = (x_2^i, y_2^i)$. The parametric equation of the line segment is:

$$L_i : \begin{cases} x = x_1^i + t \cdot \Delta x_i \\ y = y_1^i + t \cdot \Delta y_i \end{cases} \quad (6)$$

Where $\Delta x_i = x_2^i - x_1^i$ and $\Delta y_i = y_2^i - y_1^i$ are the coordinates of the endpoints, and $t \in [0, 1]$ is the parameter.

Similarly, the equation of line segment L_j is:

$$L_j : \begin{cases} x = x_1^j + s \cdot \Delta x_j \\ y = y_1^j + s \cdot \Delta y_j \end{cases} \quad (7)$$

Where $\Delta x_j = x_2^j - x_1^j$ and $\Delta y_j = y_2^j - y_1^j$ are the coordinates of the endpoints, and $s \in [0, 1]$ is the parameter.

By solving the system of equations of these two line segments, the following linear system is obtained to solve for the intersection parameters t and s :

$$\begin{bmatrix} \Delta x_i - \Delta x_j \\ \Delta y_i - \Delta y_j \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} x_1^j - x_1^i \\ y_1^j - y_1^i \end{bmatrix} \quad (8)$$

If the system of equations has a solution and both $t \in [0, 1]$ and $s \in [0, 1]$ are valid, then the intersection point is considered the actual intersection of the two line segments; otherwise, the intersection point is considered a virtual intersection point on the extension of the line segments.

Ignoring the parameter range constraints, the virtual intersection point $P_{ij} = (x_p, y_p)$ can be directly solved.

$$P_{ij} = (x_p, y_p) = (x_1^i + t \cdot \Delta x_i, y_1^i + t \cdot \Delta y_i) \quad (9)$$

After generating the virtual intersection point, validity verification is required to ensure it is a real intersection point rather than a false one caused by mismatches. If the two line segments are orthogonal after correction, and the minimum distance $d_m < 0.35m$ between their endpoints or the intersection point P_{ij} lies within the neighborhood of a line segment L_k in the corner candidate set C , then $\exists L_k \in C, \min_{E \in \{E_1^k, E_2^k\}} \|P_{ij} - E\| \leq r_{th}$, it is considered that the intersection point is valid.

For actual intersection points, any part exceeding the intersection boundary should be removed. For valid virtual intersection points, the two line segments associated with the virtual intersection point should be extended to the intersection, in order to more accurately reflect the environmental features.

After completing the previous steps, there are still corner features in the map that are not effectively connected. In the map, there are parallel edge segments with nearby endpoints, which have not been effectively connected because they did not meet the merging conditions and did not form virtual intersection points. As shown in Figure 7(b), the section within the red box represents a corner that exists in the real environment but has not been completed in the map.

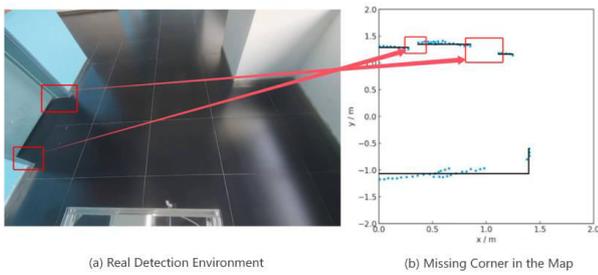


Figure 7 Partial map construction results

So, further optimization and completion of the map are required, and the process is as follows:

For the two segments L_1 and L_2 that meet the above characteristics, their adjacent endpoints $P_1 \in L_1$ and $P_2 \in L_2$ are used to calculate the midpoint coordinates M . Then, based on the direction vector of segment L_1 , an orthogonal line equation passing through point M is generated. Segments L_1 and L_2 are projected onto the orthogonal line, and the intersection points Q_1 and Q_2 are solved. Finally, extend the two segments to the corresponding intersection points, and retain the segment with endpoints Q_1 and Q_2 .

The result after optimization and completion is shown in Figure 8.

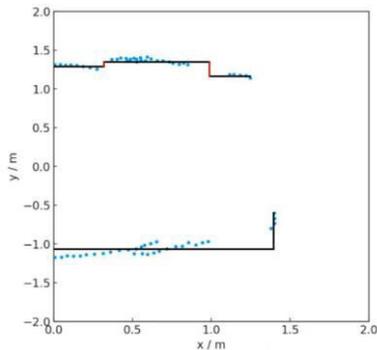


Figure 8 Optimized Completion Results

Thus, the line-based map construction based on the Manhattan World Constraint is completed. In the next section, a mapping experiment is conducted in a corridor scenario, and the results are analyzed.

III. RESULTS

This section conducts a mapping experiment in a structured corridor environment, which includes key structural features such as walls and doorframes, to validate the effectiveness of the line-based map construction method based on the Manhattan World Constraint (MWC). To assess the accuracy of the mapping results, manually measured results are used as the reference for the real map of the experimental environment, which is compared with the generated map. The experimental area and the corresponding real map schematic are shown in Figure 9.

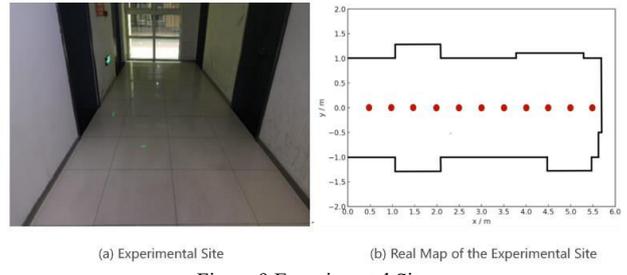


Figure 9 Experimental Site

The experimental robot is equipped with an ultrasonic sensor, which rotates and scans from 90° to -90° with a step size of 2° , collecting ultrasonic point cloud data frames, as shown in Figure 10. The robot moves along a straight line in the center of the corridor, with a measurement point set every 0.5 meters, as indicated by the red circles in Figure 9(b). At each measurement point, the robot pauses and scans. The collected data undergoes point cloud clustering, line segment fitting, and Manhattan World Constraint (MWC) optimization, ultimately resulting in a complete line-based map.



Figure 10 Experimental Robot

Figure 11 shows the initial point cloud data (Figure 11(a)), the map after applying the Manhattan World Constraint and completing line segment merging (Figure 11(b)), and the globally optimized line-based map (Figure 11(c)). The black solid lines in the figure represent the actual contours of the environment, the red line segments in Figure 11(b) represent the map after applying the Manhattan World Constraint and completing the line segment merging, and the red solid lines in Figure 11(c) represent the final optimized line-based map. Comparing these, it can be observed that the map after applying the Manhattan World Constraint better captures the corridor boundaries and aligns strictly with the main direction. However, the map lacks continuity, and some structural information is not effectively integrated. After further global optimization, the overall structure of the line-based map is more coherent and more closely matches the contours of the real environment.

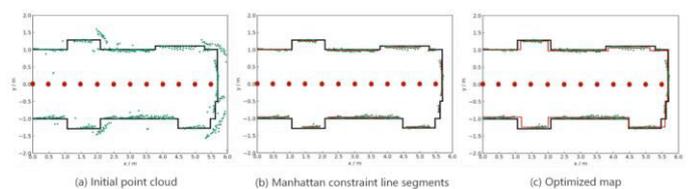


Figure 11 Line-based Map Construction

To objectively evaluate the mapping accuracy of the proposed method, Corner Detection Rate (CDR) is used to assess whether the map accurately reflects the true structural features of the environment, while Distance Error (DE) is employed to measure the overall geometric accuracy of the map.

(a)Corner Detection Rate (CDR):This metric is defined as the ratio of the number of correctly detected real corners to the total number of actual corners in the environment. It evaluates the accuracy of corner detection.

$$CDR = \frac{N_{\text{detected}}}{N_{\text{total}}} \times 100\% \quad (10)$$

Where N_{detected} is the number of detected corners and N_{total} is the total number of actual corners in the environment.

The calculated corner detection rate for this experiment is **90.91%**, indicating that the optimized map can effectively reflect the structural features of the real environment.

(b)Distance Error (DE): Defined as the average minimum distance between the constructed line-based map L_{map} and the set of ground-truth line segments L_{real} , this metric evaluates the overall geometric accuracy of the map. Given L_{map} and L_{real} as the respective sets of line segments, the definition is as follows:

$$DE = \frac{1}{M} \sum_{i=1}^M \min_{L_{\text{real}} \in L_{\text{real}}} d(L_{\text{map}}^{(i)}, L_{\text{real}}) \quad (11)$$

The calculation results show that the average distance error of the line-based map obtained from the experiment is 4.931 cm. Table 1 presents five sets of distance error data.

Table 1 Map Line Segment Endpoints and Errors

Serial Number	Endpoint Coordinates (m)	Error (m)
1	(0.006, 0.991) (1.184, 0.991)	0.009
2	(0.003, -0.947) (1.216, -0.947)	0.037
3	(1.216, -1.242) (1.216, -0.947)	0.134
4	(4.482, -1.242) (5.624, -1.242)	0.021
5	(5.663, -0.494) (5.663, 0.976)	0.012

IV. CONCLUSION

This paper focuses on the construction of line-based maps based on the Manhattan World Constraint (MWC) to address issues such as low accuracy, low resolution, and sparse point cloud data from ultrasonic sensors. To validate the effectiveness of the proposed method, mapping experiments were conducted in a structured corridor environment. The experimental results show that after applying the MWC, the line-based map can more accurately depict environmental boundaries and align with the main direction. After further global optimization, the map's continuity is significantly improved, closely matching the real environment's contours.

Quantitative evaluation using Corner Detection Rate (CDR) and Distance Error (DE) shows that the map's corner detection rate is 90.91%, and the average distance error is 4.931 cm, indicating that the method effectively enhances the accuracy and completeness of map construction.

Although the use of the MWC has improved the map accuracy in structured environments, its applicability remains limited in unstructured scenarios (such as outdoor and complex indoor environments). Therefore, future work could explore integrating geometric feature extraction methods to enhance the adaptability of ultrasonic SLAM in complex environments

REFERENCES

- [1] Dobrev Y, Gulden P, Vossiek M. An indoor positioning system based on wireless range and angle measurements assisted by multi-modal sensor fusion for service robot applications[J]. IEEE Access, 2018, 6: 69036-69052.
- [2] Vanderelst D, Steckel J, Boen A, et al. Place recognition using batlike sonar[J]. Elife, 2016, 5: e14188.
- [3] Eliakim I, Cohen Z, Kosa G, et al. A fully autonomous terrestrial bat-like acoustic robot[J]. PLoS computational biology, 2018, 14(9): e1006406.
- [4] Kim S J, Kim B K. Dynamic ultrasonic hybrid localization system for indoor mobile robots[J]. IEEE Transactions on Industrial Electronics, 2012, 60(10): 4562-4573.
- [5] Dos Santos M, Ribeiro P O, Núñez P, et al. Object classification in semi structured environment using forward-looking sonar[J]. Sensors, 2017, 17(10): 2235.
- [6] Plonski P A, Vander Hook J, Peng C, et al. Environment exploration in sensing automation for habitat monitoring[J]. IEEE Transactions on Automation Science and Engineering, 2016, 14(1): 25-38.
- [7] Abdessalem A B, Jenson F, Calmon P. Quantifying uncertainty in parameter estimates of ultrasonic inspection system using Bayesian computational framework[J]. Mechanical Systems and Signal Processing, 2018, 109: 89-110.
- [8] Leonard J J, Durrant-Whyte H F, Cox I J. Dynamic map building for an autonomous mobile robot[J]. The International Journal of Robotics Research, 1992, 11(4): 286-298.
- [9] Chen J, Fei C, Lin D, et al. A review of ultrahigh frequency ultrasonic transducers[J]. Frontiers in Materials, 2022, 8: 733358.
- [10] Chen X, Fei C, Chen Z, et al. Simulation and fabrication of 0–3 composite PZT films for ultrahigh frequency (100–300 MHz) ultrasonic transducers[J]. Journal of Applied Physics, 2016, 119(9).
- [11] Elloumi W, Treuillet S, Leconge R. Real-time camera orientation estimation based on vanishing point tracking under manhattan world assumption[J]. Journal of Real-Time Image Processing, 2017, 13(4): 669-684.
- [12] Huang Z, Gu N, Lin C, et al. Real time vanishing points detection on smartphones under Manhattan world assumption[J]. Pattern Recognition Letters, 2018, 115: 117-127.
- [13] Zou D, Wu Y, Pei L, et al. StructVIO: Visual-inertial odometry with structural regularity of man-made environments[J]. IEEE Transactions on Robotics, 2019, 35(4): 999-1013.
- [14] Peasley B, Birchfield S, Cunningham A, et al. Accurate on-line 3D occupancy grids using Manhattan world constraints[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 5283-5290.
- [15] Abadi I, El-Sheimy N. Manhattan world constraint for indoor line-based mapping using ultrasonic scans[C]//2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN). IEEE, 2022: 1-8.