IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2021, accepted March 1, 2021, date of publication March 8, 2021, date of current version March 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064198

A Position and Area Localization Algorithm for Obstacles in the Environment of Sparsely-Deployed Sensors

ZHIGANG GAO[1], XIAOWEI YANG[1], BO WU[2], (Member, IEEE), HUIJUAN LU[3], JIANHUI ZHANG[1], (Member, IEEE), WENJIE DIAO[1], AND QUN JIN[2], (Senior Member, IEEE)

[1]College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Faculty of Human Sciences, Waseda University, Tokorozawa 359-1192, Japan
[3]Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China

Corresponding author: Qun Jin (jin@waseda.jp)

This work was supported in part by the National Natural Science Foundation of China under Grant 61877015, Grant 61902185, Grant 61850410531, Grant 61803338, Grant 61572164, and Grant 61473109; in part by the Zhejiang Provincial Natural Science Foundation under Grant LY21F020028, Grant LY19F020016, Grant LY19F020042, and Grant LY19F030021; in part by the Jiangsu Provincial Natural Science Foundation under Grant BK20190448; and in part by the Project of Quality Engineering under Grant GK208802299013-103.

ABSTRACT Measuring the space area of obstacles is one of the important problems in obstacle localizing fields. Most of the existing research works on the localization of obstacles focus on where the obstacles are, and few of them measure both the positions and the areas of the obstacles. In this paper, we propose a Minimum convex bounding Polygon localizing algorithm based on Visible light Tracking (MPVT) in order to rapidly and accurately locate the position and area of a 2D obstacle in the environment of sparsely-deployed sensors. MPVT first determines the initial localization light by Visible Light Tracing method (VLT). Second, it searches for the first side of the Minimum Convex Bounding Polygon (MCBP) of the obstacle. Third, MPVT calculates the subsequent other sides and the vertexes of MCBP until the next side coincides with the first side. In order to evaluate the approximation degree between the actual values and the localization values in terms of areas, positions and shapes, we propose two performance evaluation indexes, i.e., the area ratio and the ratio of equivalent radius. We conducted experiments on the influence of obstacle orientation and sparseness of sensor deployment, the accuracy comparison with the existing methods, and the time complexity. Experiment results show that MPVT can accurately locate the position and area of the obstacle in the environment of sparsely-deployed sensors with low time overhead, and is suitable for low-cost obstacle localization applications.

INDEX TERMS Minimum convex bounding polygon, obstacle localization, ratio of equivalent radius, sparse environment, visible light tracking.

I. INTRODUCTION

With the development of society, the applications of wireless sensor networks are becoming pervasive, such as area monitoring, biological detection, home care, object tracking, etc. [1]. There may be various obstacles in the areas where sensor networks are deployed, which not only hinder data transmission and reception between sensor nodes, but also cause a series of security problems. Therefore, it is very important to locate the position and size of obstacles in order to take preventive measures in time [2], [3].

There are a number of positioning technologies in the existing research. According to the difference of the technical means used for the first time, this paper classifies the existing research related to obstacle localization into the range-based method, the image-based method, the wireless sensing-based method, and the light-based method.

In the range-based method, ultrasonic (or infrared) rangefinders and lasers are usually used to measure the distance between observation points and obstacles. This method only locates the positions of obstacles, but cannot locate

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

the shape and size of obstacles. When using ultrasonic (or infrared) rangefinders, the reflection effect is easy to be affected by the surface conditions of different objects, and the ultrasonic auxiliary equipment is easy to be interfered by the ultrasonic source [4]. When using lasers, the equipment cost is very high and the resolution is easy to be affected by environment [5]. In the wireless sensing-based method, considering the space constraints, the organization of paper, and the relationship with the obstacle positioning, we only include the literal based on fingerprints and anchors. The fingerprints of wireless signals or the positions of anchor points are used to detect the positions of obstacles, and it has the advantage of wide applicability (e.g., it can be used indoors or outdoors, and there can be walls or other objects in the measurement area.) [6]–[8]. Due to the characteristics of radio waves, this method only locates the positions or recognizes the behaviors of obstacles, and usually need complex equipment. In the image-based method, monocular or binocular cameras are usually used to capture the surrounding environment, and then machine learning algorithms (e.g., the deformable grid method [9], the convolutional neural network [10], the image segmentation [11], the support vector machine (SVM) [12], or the Markov chain [13], etc.) are used to process and extract the information of obstacles in the images, thereby locating the position and size of the obstacle. However, the localization effect of this method is susceptible to the motion tracking errors of cameras, and has the disadvantages of high calculation overhead and costs. In the light-based method, LED is usually used to locate the positions. During LED localization, LED lights are used as light sources, and the photoelectric sensors are used as receivers. LED lights are located above objects, and receivers can receive lights within the range covered by LED lights. This method can obtain the relative position between receivers and LED lights by analyzing the intensity and angle of the received optical signals [14], [15], or the received light beams marked with labels [16]. This method has the disadvantage that the intensity of the lights received by each receiver is different, resulting in a large deviation when locating the receiver. Therefore, this method is suitable for indoor localization.

Few of the existing research efforts on the localization of obstacles consider both the position and the area of the obstacle. Aiming at the problem of obstacle localization in two-dimensional (2D) areas, this paper presents a new method named MPVT (Minimum convex bounding Polygon localizing algorithm based on Visible light Tracking). MPVT locates the positions and areas of obstacles by using the visible areas of LED lights from different viewpoints. The application scenarios of MPVT are that light-emitting nodes are deployed along the sides of a rectangular, and a light-receiving node moves around the rectangular sides and detects the lights. The light-emitting nodes are sparsely distributed on the rectangular sides with a fixed gap, and the light-receiving node can receive the light signals. In the process of localizing, at each light-receiving position (i.e., viewpoint), MPVT will find two lights that bound and are closest to the outer surfaces of the obstacle, and then it derives MCBP (Minimum Convex Bounding Polygon) of the obstacle from all the viewpoints. The position and area of the minimum surrounded polygon are regarded as where the obstacle lies.

The contributions of this paper are twofold. First, it presents MPVT, a new localization algorithm and its implementation process in sensor-sparse environment. Second, it presents two indexes for localization performance evaluation, i.e., the area ratio S_r, and the ratio of equivalent radius L_r. Compared to the existing localization methods, MPVT can accurately locate not only the position, but also the area of the obstacle in the environment of sparsely-deployed sensors while has low time overhead.

The rest of this paper is organized as follows. Section II introduces the related work in the field of obstacle detection. Section III gives the research scenario of this paper and the definitions of related terms. Section IV details the implementation process of MPVT. Section V conducts the experiments and analyses. Finally, Section VI summarizes this paper.

II. RELATED WORK

In this section, we introduce the related work in the range-based method, the wireless sensing-based method, the image-based method and the light-based method.

In the range-based method, in order to solve the problem of obstacle avoidance during the flight of a drone, Gageik *et al.* [4] proposed to use ultrasonic sensors and infrared rangefinders to detect the distance and the position between the drone and the obstacles. In order to make the mobile robot avoid obstacles, Peng *et al.* [5] placed a 2D lidar on the robot for distance measurement, and obtained the position information of obstacles by filtering and clustering the laser point cloud data. Hussein *et al.* [17] adopted a combination of stereo vision and laser rangefinder for obstacle avoidance of ground vehicles. They first measured the approximate position of the obstacle by the laser rangefinder, and then took an image by using a binocular camera to identify the obstacle in the image. Zhang *et al.* [18] first used millimeter-wave radar to obtain rough position information of obstacles, and filtered data to extract effective targets. Second, they used coordinate system conversion and camera parameter calibration to project effective targets to the image plane and generate ROI (Region of Interest). Finally, they detected and tracked the vehicle targets in the ROI based on image processing and machine learning techniques.

In the wireless sensing-based method, Yang *et al.* [6] presented a multi-dimensional RSS feature fuzzy mapping and clustering method with high positioning accuracy and low positioning overhead. This method first obtained the Maximum Information Coefficient (MIC) between different APs (Access Points), the set of the correlated APs, and the RSS feature fuzzy weight in the online phase, and then selected the non-redundant APs with the high location resolution for the target positioning. Singh *et al.* [19] proposed a localization method based on analytical geometry. This method used three

positions of non-collinear beacon points to derive the position of a sensor node, and it could archive high localization accuracy because an adaptive mechanism was used to select random points with different sizes of the constraint area. Xiong *et al.* [20] proposed a positioning method of hybrid received signal strength (RSS) and time-of-arrival (TOA) in order to locate multiple targets in a 3-D cooperative wireless sensor network. This method first derived the coarse coordinates of targets' positions, and then optimized the targets' positions through the communication information between the targets and the specific sensors. Zhou *et al.* [8] modeled the relationship between sample capacity and localization errors, and constructed an energy-efficient location fingerprint database by sampling the minimal number of independent and non-redundant signals with the expected localization accuracy.

In the image-based method, researchers usually take a photo or video for the obstacle areas by cameras or camcorders, and then extract obstacles in the image by image processing techniques. Pham *et al.* [21] used the Kinect sensor to collect the color depth data in front of the user, and then used the 3D sensing OpenNI framework to process and extract accurate 3D information of the obstacles by the PCL library. Kang *et al.* [9] adopted a DG (Deformable Grid) method to detect obstacles in order to help persons with visual impairment. DG initially has a regular grid shape. This grid will gradually deform with the movement of objects in the scene captured by the camera, and returns to the original shape when the object leaves the scene. The DG method detects the position and size of objects which have collision risks by the deformation degree of DG. In order to detect the nearest obstacle from a driving vehicle, some researchers [10], [22]–[24] first used the measured dataset to train a convolutional neural network, and then input the image data captured by the monochrome camera into the trained convolutional neural network to extract the position of obstacles. Kristan *et al.* [11] used the Markov random field frame to segment the single video stream, and then extracted the texture features of the obstacle to identify the position of the obstacle in order to avoid collision between ships and obstacles during navigation. Abiyev *et al.* [12] proposed an obstacle detection and pathfinding algorithm using SVM (Support Vector Machine) and A* algorithms in mobile robots. Gupta *et al.* [13] integrated the appearance of the image and 3D cues (e.g. image gradient, curvature potential, and depth variance) into the MRF (Markov Random Field) formula to identify the area of obstacles.

In the light-based method, some researchers [14], [15] used the photoelectric sensor as a receiver to receive the signals emitted by the LED light, and estimated the position of the LED light relative to the receiver from the received signal strength and reception angle in order to locate indoor obstacles. Xie *et al.* [16] used the light sensor deployed around the object as a receiver, labeled each beam, and then counted the labeled beams received by the receiver to identify the positions of indoor objects. Guan *et al.* [25] presented a double-light positioning system based on image sensors. This system first identified the IDs (IDentities) of LED lights by utilizing the rolling shutter mechanism of the CMOS (Complementary Metal Oxide Semiconductor) image sensors and machine learning algorithms such as the Fisher classifier and the linear support vector machine, and then it located the position of an object by coordinate transformation. This system had high real-time performance and positioning accuracy, and low complexity. Moreover, Guan *et al.* [26] proposed an indoor robot VLP (Visible Light Positioning) localization system based on Robot Operating System (ROS). They designed a VLP localization package, and implemented a prototype system on a Turtlebot3 Robot which achieved the indoor positioning accuracy within 1 centimeter and the average computational time of only 0.08 second.

III. PROBLEM DESCRIPTION

In this section, we will give the definitions of localization scenarios and some related terms used in this paper.

A. LOCALIZATION SCENARIOS

The purpose of this paper is to identify obstacles by using the light-based method in a sparse environment. In the following parts of this paper, an obstacle refers to an object which lies in the measurement area. The obstacle has no communication function and indication function, and can affect the normal work within the measurement area (e.g. signal transmission, or robot cruise). The deployment positions of the LED emitters are called the light-emitting points, and the light acquisition positions of the light receivers are called the light-receiving points. To the best of our knowledge, there is no uniform definition of sparseness, and how to define sparseness is related to the research goals [27]–[30]. In this paper, if the permitted localization error is α meter, we must set the deployment distance between sensor nodes to no more than $\alpha/2$ meter. We can define $\alpha/2$ meter as one unit of the x axis and the y axis. We define that the sensors are sparely deployed when the distance between two light-emitting points is $m^*\alpha/2$ meter (m$\geq$2), i.e., the distance between two light-emitting points is m units along the sides of the measurement area. Besides, we assume that the measurement area is a square, the length and width of the obstacle are all not less than 1 and the obstacle is a convex polygon. Note that although an obstacle is a 3D object, this paper only aims at the 2D localization problem where the measurement area is a horizontal plane, and it does not consider the pose of the obstacle. The position and shape of the obstacle that MPVT derives is the crosscut between the obstacle and the horizontal plane where the light-emitting points and light-receiving points lie. In this paper, we believe that the obstacle is opaque and its surfaces only block lights but not reflect lights, and do not consider the influence of emitting-receiving devices on the performance or sparseness.

Fig. 1 shows a measurement area with the side length of 6. The measurement area is enclosed by four vertices with the coordinate points of (0, 0), (6, 0), (6, 6), and (0, 6), where one of the vertices is used as the coordinate origin, and the

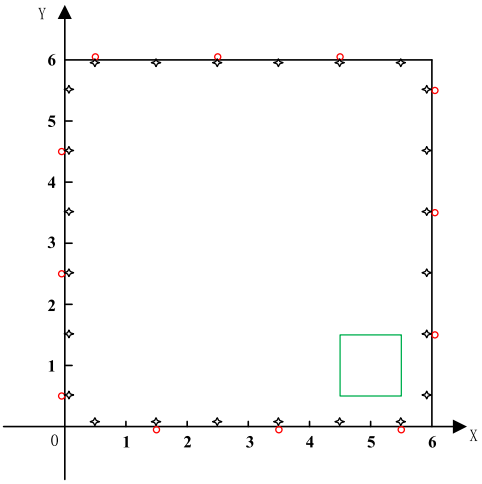


FIGURE 1. Measurement area.

measurement area is in the first quadrant. The square at the bottom right of the measurement area is an obstacle. At the side of the measurement area, the light-receiving points and the light-emitting points are set in the center between two scales. From the position of (0.5, 0), in the counter-clockwise direction, we set light-receiving points at every unit. In Fig. 1, the black diamond points denote the positions of the light-receiving points, and they are located in the center between two scales, e.g. (0.5, 0), (1.5, 0), ..., and so on. The light-emitting points are set according to the degree of sparsity. For example, if the degree of sparsity is m, the distance between two emitters will be m units. The red circles in Fig. 1 are the deployment positions of the light-emitting points when the sparsity is 2.

In real-world applications, the light-emitting points are LED emitters, and a small vehicle with a light-receiver moves around the sides of the measurement area to receive the signals from LED emitters in order to reduce costs. In order to avoid the interference between the light-emitting points and the light-receiver, the light-receiver moves in front of the light-emitting points (i.e., the inner side of the measurement area), or keeps a smaller vertical space from the light-emitting points (although this paper considers them to be on the same horizontal plane).

B. DEFINITION OF TERMS

We first introduce the concept of viewpoint in order to make it easy to understand the process of MPVT. Since a light-receiving point P_i is in fact the point where the lights are viewed, P_i is also called the viewpoint V_i. Without loss of generality, we use V_i (1.5, 0) in Fig. 2 as the viewpoint to illustrate the definition of terms used in the following parts of this paper.

Visible area at V_i: The area(s) where the passing lights will not be blocked by obstacles. As shown in Fig. 2, R_1 and R_3 are the visible areas at V_i.

Blocking area at V_i: The area(s) where the passing lights will be blocked by obstacles. As shown in Fig. 2, R_2 is the blocking area at V_i.

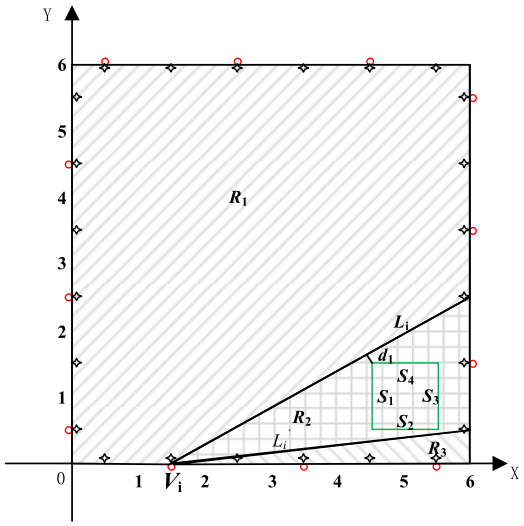


FIGURE 2. Three areas of viewpoint.

As shown in Fig. 2, the visible area R_1 is in the counter-clockwise direction of the straight line segment L_i with endpoints coordinates of (1.5, 0) and (6, 2.5), and L_i is defined as the left side line of the blocking area R_2; The visible area R_3 is in the clockwise direction of the straight line segment L_i' with endpoints coordinates of (1.5, 0) and (6, 0.5), and L_i' is defined as the right side line of the blocking area R_2. When locating an obstacle, starting from any light-receiving point, the measurement area can be divided into one blocking area and one or two visible areas.

Visible surface of the obstacle from V_i: The largest visible outer surface of an obstacle from a certain viewpoint V_i is defined as the visible surface of the obstacle at V_i. For example, in Fig. 2, the visible surfaces of the obstacle at V_1 are S_1 and S_2.

Distance between a viewpoint V_i and an obstacle: The shortest vertical distance from the visible surface of the obstacle seen from V_i to L_i is defined as the distance between V_i and the obstacle. For example, in Fig. 2, the distance between V_1 and the obstacle O is the length of d_1.

Visible Light Tracing method (VLT): VLT is a method which is used to find the next visible light. The key idea of VLT is that it starts from one light and then finds the next visible light according to a specified mean.

In this paper, MPVT uses VLT in the process of searching for the initial localization light and the sides of MCBP.

It should be noted that, according to the reversibility principle of lights, when a position V_i is set to both a light-emitting point and a light-receiving point, V_i can see all other light-emitting points and light-receiving points. But if V_i only sets a light-receiving point, V_i can only see other light-emitting points.

For each viewpoint V_i, there is a left side line L_i of the blocking area and a right side line L_i' of the blocking area. After obtaining the left and right side lines of all viewpoints, we can get a MCBP of the obstacle. The purpose of MPVT is to find MCBP, i.e., the position of the obstacle.

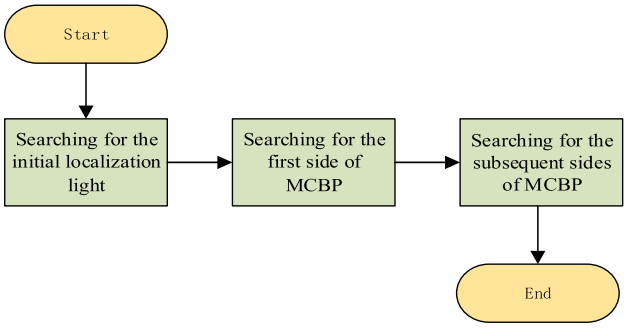


FIGURE 3. The implementation flow of MPVT.

IV. IMPLEMENTATION PROCESS OF MPVT

In this section, we will detail the implementation process of MPVT.

As shown in Fig. 3, the MPVT algorithm first finds the initial localization light. Second, from the initial localization light, it uses the law of counter-clockwise intersecting lights to find the first side of MCBP of the obstacle. Third, from the first side of the minimum polygon, the law of counter-clockwise intersecting light is used to find the subsequent sides of MCBP.

A. SEARCHING FOR THE INITIAL LOCALIZATION LIGHT

It includes the following two steps.

Step 1: Find the left side line and right side line of the blocking area of each viewpoint in the measurement area.

Suppose the side length of the measurement area is n, MPVT first creates a $4n \times 2$ matrix *lineList*. Second, starting from V_i (0.5, 0), MPVT traverses each viewpoint in a counterclockwise direction. Third, MPVT finds the left side line and right side line of each viewpoint, and stores the two endpoints coordinates of the left side line and right side line of each viewpoint into *lineList*.

We take Fig. 4 as an example to explain how to calculate the left line and right side line of a viewpoint V_i. MPVT first calculates the minimum angle θ_{min} and the maximum angle θ_{max} between the connection lines from V_i to each vertex of the obstacles and the sides of the measurement area where V_i lies as the x-axis direction. Second, MPVT calculates the intersection points. P_{min} is the intersection point between V_i and the side of the measurement area when the intersection angle is θ_{min}. P_{max} is the intersection point between the light from V_i and the side of the measurement area when the intersection angle is θ_{max}. Third, find the nearest light line in the counter-clockwise direction of P_{max}, i.e., the left side line L_i; similarly, find the nearest light in the clockwise direction of P_{min}, i.e., the right side line L_i'.

Step 2: Use VLT to find the initial localization light.

We use VLT to find the initial localization light according to the following steps.

First, the line segment stored in *lineList[0][0]*, i.e., the left side line of the blocking area at V_1 (0.5, 0) is defined as the base line L_b. The other vertex of L_b except V_1 is set to V_2. Second, starting from V_2, the other vertex of L_2' (L_2' is the right side line of V_2) except V_2 is set to V_3. Third, starting from V_3, the next vertex of L_3 (L_3 is the right side line of V_3) except V_3 is set to V_4. The above process is repeated until the next new light cannot be found. The final light is the initial localization light L_p, and the viewpoint is V_p.

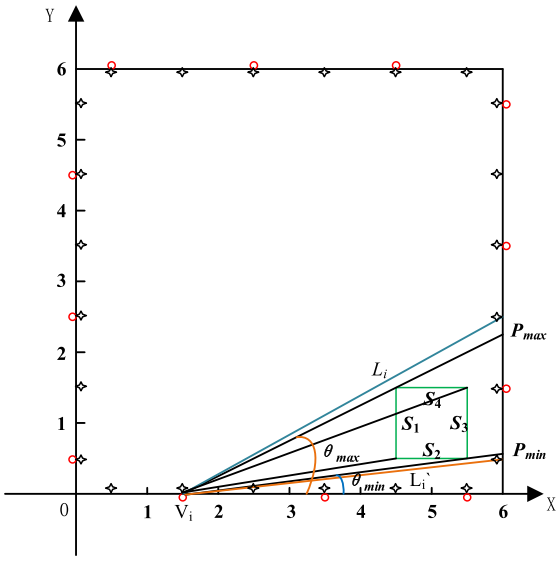


FIGURE 4. Localization of the left side line and right side line of obstacles.

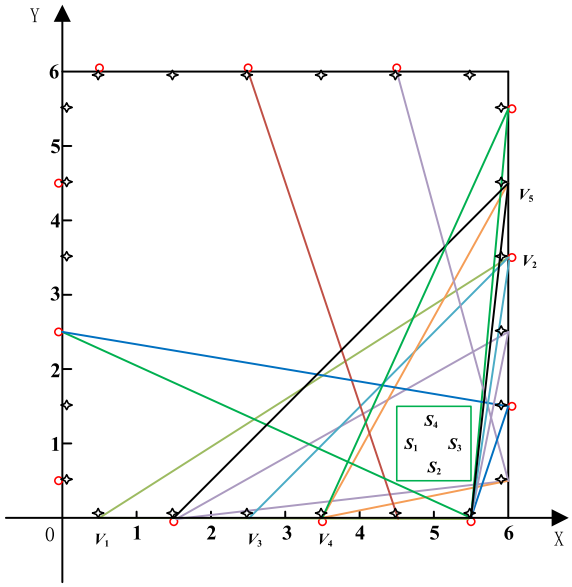


FIGURE 5. Example of searching for the initial localization light.

For example, in Fig. 5, starting from V_1, the L_1 is V_1V_2, and L_2' is V_2V_3. Starting from V_3, L_3 is V_3V_2, which coincides with the previous line, so the initial localization light L_p is V_3V_2, and the end position V_p is V_3.

B. SEARCHING FOR THE FIRST SIDE OF MCBP

In this subsection, we present the law of counter-clockwise intersecting lights and Theorem 1 to search for the first side of MCBP of the obstacle.

Law of counter-clockwise intersecting lights: It is the method of searching for the intersection points between the

left side line L_i of V_i and the left side lines of other related viewpoints. This method includes the following two steps. First, it starts from V_i, and moves counter-clockwise along the right side of V_i and the next side of the measurement area until the corresponding intersection point between L_i and the side of the measurement area arrives. Second, it calculates all the intersection points between L_i and the left side lines of all the viewpoints that it meets in the last step.

Theorem 1: Taking V_p as the viewpoint, the light L_d whose intersection point with L_p has the farthest distance from V_p according to the law of counter-clockwise intersecting lights must be a part of MCBP of the obstacle.

Proof: It can be concluded from the method of finding the initial localization light that L_p is the light closest to the obstacle among all the lights starting from point V_1. Assume that the surface of the obstacle closest to L_p is S_i. (If it lies at the intersection of two surfaces, then we select the surface in the counter-clockwise direction. For example, in Fig. 5, the closest point from the obstacle to V_2V_3 is the intersection of S_1 and S_4, so the obstacle surface where the closest point lies in this case is S_1.). According to the law of counter-clockwise intersecting lights, we can get the left side line L_d of V_d which has the farthest distance from V_p. Because the intersection of the inner blocking areas of the left side lights L_d and L_p has the minimum area, the union of the visible area at V_p and V_d has the largest visible area, and L_d must be the light closest to S_i. Otherwise, if L_d is not the light closest to S_i, it means that there is a visible area under the intersection point of L_d and L_p. There is still a light in the counter-clockwise direction of L_d that is closer to S_i, so L_d is not the initial localization light, which contradicts the assumption. Therefore, L_d must be the light closest to S_i, and it must be the side of MCBP of the obstacle.

According to the law of counter-clockwise intersecting lights and Theorem 1, the method to search for the first side of MCBP of the obstacle is as follows:

Step 1: Create a matrix M_1 to store the coordinates of MCBP when searching for MCBP's side;

Step 2: Traverse the elements in the first column of *lineList*, mark the line where V_p lies as r_1, and mark the line where the other endpoint of L_p lies as r_2;

Step 3: Traverse each element in *lineList* from r_1 to r_2 to find the left side line of the blocking area of each viewpoint;

Step 4: Traverse each left side line L_j, and use the principle of vector cross product to determine whether L_j has an intersection point with L_p.

Here we take two intersecting lines AB and CD in Fig. 6 as an example.

As shown in Fig. 6, when it is true that both C and D lie on different sides of the line AB and A and B lie on different sides of the line CD, it means that AB intersects with CD. The above principle can be denoted by the equations as follows:

$$\begin{cases} (\overrightarrow{AB} \times \overrightarrow{AC}) \cdot (\overrightarrow{AB} \times \overrightarrow{AD}) \le 0 \\ (\overrightarrow{CD} \times \overrightarrow{CA}) \cdot (\overrightarrow{CD} \times \overrightarrow{CB}) \le 0 \end{cases} \tag{1}$$

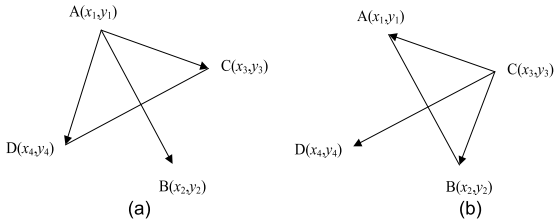


FIGURE 6. Determining whether two straight lines intersect by using cross product.

The equation of the vector cross product for $\overrightarrow{AB} \times \overrightarrow{AC}$ is defined as

$$\overrightarrow{AB} \times \overrightarrow{AC} = [(x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)] \overrightarrow{k} \tag{2}$$

The direction of $\overrightarrow{k}$ is vertical to this paper surface and points at the reader's direction.

Similarly:

$$\overrightarrow{AB} \times \overrightarrow{AD} = [(x_2 - x_1) * (y_4 - y_1) - (x_4 - x_1) * (y_2 - y_1)](-\overrightarrow{k}) \tag{3}$$

$$(\overrightarrow{AB} \times \overrightarrow{AC}) \cdot (\overrightarrow{AB} \times \overrightarrow{AD}) = \left|\overrightarrow{AB} \times \overrightarrow{AC}\right| * \left|\overrightarrow{AB} \times \overrightarrow{AD}\right| * \cos < \overrightarrow{k}, -\overrightarrow{k} > \tag{4}$$

The cross product of the other vectors in Equation (1) can be calculated in a similar manner.

If AB and CD do not intersect, jump to Step 5. If they intersect, suppose the equations of AB and CD are $a_1x + b_1y + c_1 = 0$ and $a_2x + b_2y + c_2 = 0$, respectively (where $a_1 = y_1 - y_2, b_1 = x_2 - x_1, c_1 = x_1 * y_2 - x_2 * y_1, a_2 = y_3 - y_4$, $b_2 = x_4 - x_3, c_2 = x_3 * y_4 - x_4 * y_3$). Then we use Equation (5) to calculate the coordinates P_i of the intersection point, and use Equation (6) to find the distance d between V_p and P_i.

$$\begin{cases} x = (b_1 * c_2 - b_2 * c_1)/(a_1 * b_2 - a_2 * b_1) \\ y = (a_2 * c_1 - a_1 * c_2)/(a_1 * b_2 - a_2 * b_1) \end{cases} \tag{5}$$

$$d = \sqrt{(X_i - X_p)^2 + (Y_i - Y_p)^2} \tag{6}$$

where $(X_\mathrm{p}, Y_\mathrm{p})$ is the coordinate of V_p, $(X_\mathrm{i}, Y_\mathrm{i})$ is the coordinate of P_i.

Step 5: The light which has an intersection with L_p and the intersection point is farthest from V_p is the first side of MCBP.

For example, in Fig. 5, the initial localization light is V_3V_2. When we traverse all viewpoints between V_3 and V_2 in the counter-clockwise direction of V_3, only the left side lines of the blocking areas of viewpoints (3.5, 0), (4.5, 0), and (5.5, 0) have intersection points with V_3V_2. Besides, the left side line of the blocking area from the viewpoint (3.5, 0), i.e., the line segment L_4 (i.e., L_d) composed of (3.5, 0) and (6, 4.5), has the largest distance from the intersection point with V_3V_2 to V_3. L_4 is the first side of MCBP of the obstacle.

C. SEARCHING FOR THE SUBSEQUENT SIDES OF MCBP

After obtaining the first side of MCBP of the obstacle, we can use the law of counterclockwise intersecting lights

and Theorem 2 to calculate the subsequent sides of MCBP until all the sides of MCBP are obtained.

Theorem 2: Taking the side of MCBP of the obstacle as the left side line L_i of the blocking area from V_i, the light L_j that is the farthest away from V_i according to the law of counter-clockwise intersecting lights must be the next side of MCBP of the obstacle O.

Proof: Since L_i is a side of MCBP of the obstacle, and according to the law of counter-clockwise intersecting lights, we can get the light L_j that is farthest from the intersection point of the left side line of the viewpoint V_i. Because the intersection area has the minimum blocking area in the direction of S_i or the next visible surface S_{i+1} in the counter-clockwise direction, L_j must be the next side of MCBP of the obstacle O.

According to the law of counter-clockwise intersecting lights and Theorem 2, the method for determining the next light of the subsequent minimum surrounded polygon is as follows:

Step 1: Mark the starting point of the first side L_d of MCBP as the viewpoint V_1, and traverse each element of *lineList*. Once the line of the coordinates of the starting point V_1 is found, we marked it as r;

Step 2: Determine the number of viewpoints N that need to be traversed when looking for viewpoints on the right of V_1 and the next side of the measurement area according to the law of counter-clockwise intersecting lights;

Step 3: Traverse from the first column of the $(r+1)$th row to the first column of the $(r+N)$th row in the *lineList* to find the left side line of the blocking area of each viewpoint;

Step 4: Traverse each left side line to determine whether it intersects with L_d. If it is *true*, we calculate the coordinates of the intersection points and the distance d between V_1 and each intersection point;

Step 5: The light that has an intersection point with L_d and the distance between the intersection point and V_1 is the largest one is the next side of MCBP. We store the intersection point coordinates in M_1;

Step 6: Mark the newest found side as L_d, and the viewpoint of L_d as V_1, and repeat the steps from 1 to 5. If L_d is the same as the first side of MCBP (e.g. the coordinates of the endpoints are the same), it means that all the vertices of MCBP have been found. We calculate the intersection point with the last side of MCBP and store it to M_1. By traversing all the coordinates of the vertices in M_1 and connecting them in sequence, then we get MCBP of the obstacle.

V. EXPERIMENT AND ANALYSIS

In this section, we first present the performance evaluation indicators for obstacle area localization, and then we conducted performance evaluation in three different scenarios: (1) The obstacles are located at the different distance from the center of the measurement area and in the same orientation; (2) The obstacles are located at the same distance from the center of the measurement area and in symmetric orientations; (3) The light-emitting points have different sparseness.

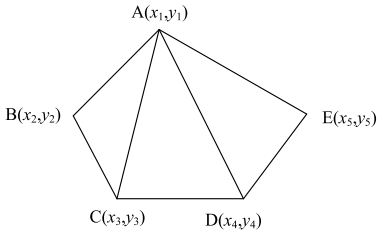


FIGURE 7. Convex polygon consisting of multiple triangles.

Moreover, we compared the time complexity between MPVT and the algorithm in [31].

A. EXPERIMENT ENVIRONMENT

The hardware platform of this experiment is a Dell OptiPlex 9020 desktop computer with the CPU of Intel Core i5-4900, the operating frequency of 3.30GHz, and the memory size of 12GB. We assume an obstacle lies somewhere in the measurement area in a static state, and we have obtained all the LED lights except the ones blocked at each light-receiving points. Therefore, we can locate the position and size of the obstacle by MPVT. We used C++ to develop the implementation of MPVT and conducted the localization experiments.

B. INDEXES FOR PERFORMANCE EVALUATION

After obtaining MCBP of the obstacle, in order to evaluate the approximation degree between the actual value and the localization value in terms of area sizes and positions, we propose two performance evaluation indexes as follows. (1) The area ratio S_r, which is the ratio of the actual area of the obstacle to the area of MCBP, reflecting the proximity degree of their area sizes; (2) The ratio of equivalent radius L_r, which is the ratio of the centroid distance between the obstacle and MCBP to the equivalent circle radius of the obstacle, reflecting the proximity degree of their positions and shapes. In this section, we take Fig. 7 as an example to illustrate the calculation method of S_r and L_r.

The calculation method of S_r is as follows.

Step 1: Calculate the actual area of the obstacle and the area of MCBP. As shown in Fig. 7, if we know the coordinates of all vertexes, we can calculate the area of the convex polygon ABCDE as follows. First, we decompose it into multiple triangles, and then calculate the areas of ABC, ACD, and ADE respectively. Second, we add their areas up to get the areas of convex polygon. For example, the area of ABC can be obtained by Equation (7) according to its vertex coordinates.

$$s = \frac{1}{2}(x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_2y_1 - x_3y_2) \quad (7)$$

The area of a convex polygon can be calculated by accumulating the area of all triangles it contains. As shown in Equation (8), where k is the number of triangles that the polygon consisting of. For the convex polygon ABCDE in Fig. 7, k is 3.

$$S = \sum_{i=1}^{k} s_i \quad (8)$$

Step 2: Calculate the area ratio of the obstacle to MCBP according to Equation (9):

$$S_r = \frac{S_a}{S} \tag{9}$$

where S_a is the actual area of the obstacle, S is the area of MCBP, and S_r is the ratio of their areas.

The calculation method of L_r is given as follows:

Step 1: Substitute the area S of the obstacle into Equation (10) to derive the radius R of its equivalent circle:

$$S = \pi R^2 \tag{10}$$

Step 2: Substitute the vertex coordinates (X_i, Y_i) of the obstacle into Equation (11) to derive the centroid coordinates of the obstacle:

$$\begin{cases} \overline{X} = \dfrac{\sum_{i=1}^{n} X_i}{n} \\ \overline{Y} = \dfrac{\sum_{i=1}^{n} Y_i}{n} \end{cases} \tag{11}$$

Similarly, the centroid coordinates of MCBP of the obstacle can also be calculated by Equation (11).

Step 3: Calculate L_r by using Equation (12), which is used to measure how close it is between MCBP of the obstacle and the actual position of the obstacle. The distance D between the centroid of the obstacle and its MCBP can be drived by Equation (6). The smaller L_r is, the closer they are.

$$L_r = \frac{D}{R} \tag{12}$$

C. EXPERIMENTS ON THE INFLUENCE OF OBSTACLE ORIENTATION

In the following experiments, unless additional explanations, the measurement area is a square with the side length of 20, the gap between light-emitting points is 2, and the obstacle is a square with a side length of 2. We conducted experiments in two different scenarios to evaluate the effectiveness of MPVT, and these scenarios include: (1) The obstacles are deployed at the same distance from the center of the measurement area and in a symmetric orientation; (2) The obstacles are located at the same distance from the center of the measurement area and in an asymmetric orientation.

1) THE OBSTACLE IS LOCATED AT THE SAME DISTANCE FROM THE CENTER OF THE MEASUREMENT AREA AND IN A SYMMETRIC ORIENTATION

We use the MPVT algorithm to obtain MCBP of the obstacle. When the obstacles are deployed at the center position C of the measurement area, the upper left corner UL, the upper right corner UR, the lower left corner LL, and the lower right corner LR respectively, S_r and related parameters are shown in Table 1, and L_r and related parameters are shown in Table 2.

As shown in Tables 1 and 2, when the obstacle is located at the same distance far from the center of the measurement area and in a symmetric orientation. S_r is all about 70%. Besides, the distance between centroids and L_r is about 10%. Considering the slight differences caused by the accuracy of the calculation, we can draw the conclusion that when the obstacles are located at the same distance from the center of the measurement area and in a symmetric direction, both S_r and L_r do not change.

2) THE OBSTACLE IS LOCATED AT THE SAME DISTANCE FROM THE CENTER OF THE MEASUREMENT AREA AND IN AN ASYMMETRIC ORIENTATION

In this experiment, we use the angle between the horizon orientation and the straight line orientation that links the center of the measurement area with the obstacle as the deployment position of the obstacle. The obstacle lies in the direction of 0°, 22.5° and 45° respectively. We gradually increase the distance between the centroid of the obstacle and the center of the measurement area from the distance of 1 to the position that the outer side of the obstacle aligns with the side of the measurement area.

In Figs. 8(a) and 8(b), in the direction of 0°, 22.5°, and 45°, the distance d_i between the centroid of the obstacle and the center of the measurement area increases from 1 to 8, from 1 to 9, and from 1 to 12, respectively. As shown in Fig. 8(a), with the increase of d_i, S_r generally has an increasing trend in the direction of 0° and 22.5°. While in the direction of 45°, S_r generally shows a trend of vibration. The reasons are as follows. With the increase of d_i, the obstacle is becoming closer to one side of the measurement area in the directions of 0° and 22.5°, and becoming farther away from the other sides, making the bounding polygon closer to the obstacle. Because the 45° direction lies at the middle position of the two sides, the shadow area caused by the obstacle has a greater influence, which counteracts the effect of approaching the side of the measurement area, and thereby S_r shows the vibration effect. In addition, due to the large shadow area, it is difficult to approximate the side of the obstacle for MCBP.

In the direction of 0°, 22.5°, and 45°, L_r generally shows a trend of vibration. It is because the shape of sides of MCBP and the distance from the obstacle is changing continuously when the obstacle was moved from the center of the measurement area to the side of the obstacle, which leads to the vibration of L_r. Except the maximum value of L_r in the direction of 45° and near the side of the measurement area, L_r is no more than 0.35, which shows the centroid deviation is small.

D. THE INFLUENCE OF SPARSENESS OF SENSOR DEPLOYMENT

In this experiment, the measurement area is a square with the side length of 30. In order to test the impact of sparseness on obstacle detection, we set the gaps between the light-emitting points to 1, 2, 3, and 4, and the gaps between the light-receiving points to 1, respectively. The four obstacles are numbered as O_1, O_2, O_3, O_4. O_1 and O_2 are squares with sides of 2 and 3 respectively. O_3 is an isosceles ladder shape

TABLE 1. S_r and related parameters.

Obstacle position	Distance between the center of the obstacle and the center of the measurement area	Obstacle area	MCBP area	S_r
C	0	4	6.3506	0.6299
UL	7.0711	4	5.6703	0.7054
UR	7.0711	4	5.6703	0.7054
LL	7.0711	4	5.6703	0.7054
LR	7.0711	4	5.6703	0.7054

TABLE 2. L_r and related parameters.

Obstacle position	Centroid of the obstacle	Centroid of MCBP	Centroids distance D	Equivalent circle radius R of the obstacle	L_r
C	(10,10)	(9.8,9.9)	0.2236	1.1287	0.1981
UL	(5,15)	(5.12,15.02)	0.1218	1.1287	0.1079
UR	(15,15)	(14.98,14.88)	0.1218	1.1287	0.1079
LL	(5,5)	(4.98,5.12)	0.1218	1.1287	0.1079
LR	(15,5)	(14.88,4.98)	0.1218	1.1287	0.1079

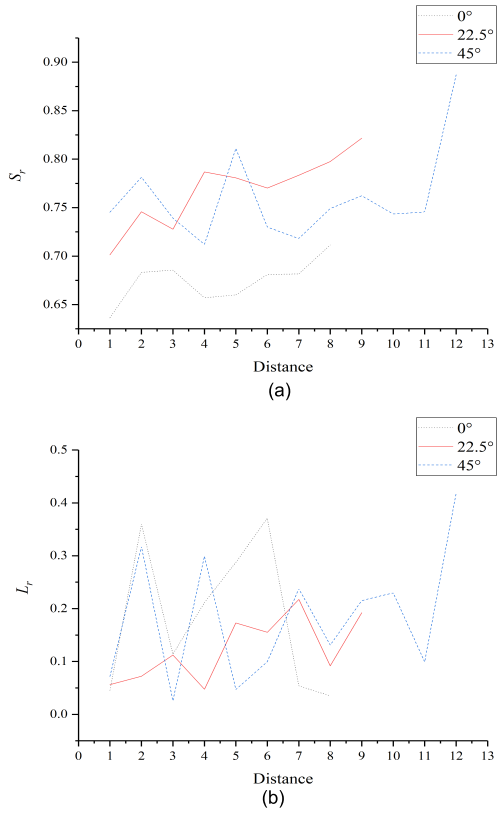


FIGURE 8. S_r and L_r at different distances.

with the upper line length of 2, the lower line length of 4, and the height of 4. O_4 is an isosceles ladder shape with the upper line length of 4, the lower line length of 6, and the height of 4. The centroids of O_1-O_4 are all located at the center of the measurement area.

The parameters of S_r and L_r with different gaps of light-emitting points are listed from Table 3 to Table 10.

With regard to the area of obstacles, we can see the following facts from Tables 3, 5, 7 and 9. (1) Although the area of the four kinds of obstacles and the gaps between the light-emitting points are different, the area of MCBP is always larger than that of the obstacle; (2) The values of S_r are all about 0.7, indicating that the area of MCBP obtained by the MPVT algorithm when the sensor is sparsely deployed is close to that of the obstacle; (3) At the same sparsity degree, the larger the area of the obstacle, the larger S_r is. It is because the larger the object area is, the less S_r is affected by the sparsity degree; (4) As the gap between the light-emitting points increases, the area of MCBP becomes larger due to the decrease in the localization resolution, and S_r decreases slightly. The errors of S_r in the sparse environment (when the gap between light emitting points is 1) vary between 0.01% and 3% compared to the dense environment.

With regard to the difference of centroids, as shown in Tables 4, 6, 8 and 10, L_r is within the range that the maximum value is about 20%, and the minimum value is less than 10%. It means the centroid of the obstacle is close to that of MCBP, and the distance between the two is much smaller than the equivalent circle radius of the obstacle. Therefore, their relative positions are close.

The above experiments show the MPVT algorithm proposed in this paper has a good effect in the sparse sensor deployment environment.

E. ACCURACY COMPARISON WITH THE EXISTING METHODS

The existing research works on the localization of obstacles are basically about where an obstacle is, without considering both the position and the size of the obstacle. We compared MPVT with the two baseline methods presented in [31], [32] and [33] which can measure the position and size of obstacles. The localization method of [31] can refer to the steps 3 in the following Subsection F. EXPERIMENT ON

TABLE 3. S_r when the gap between light-emitting points is 1.

Obstacle number	Obstacle area	MCBP area	S_r
1	4	6.1286	0.6527
2	9	12.4926	0.7204
3	12	16.0864	0.746
4	20	25.3467	0.7891

TABLE 4. L_r when the gap between light-emitting points is 1.

Obstacle number	Centroid of the obstacle	Centroid of MCBP	Centroid distance D	Equivalent circle radius R of the obstacle	L_r
1	(15,15)	(14.93,14.93)	0.1061	1.1287	0.094
2	(15,15)	(14.84,15.33)	0.3624	1.693	0.2141
3	(15,15)	(14.83,14.96)	0.1715	1.9549	0.0877
4	(15,15)	(15.14,15.35)	0.3818	2.5238	0.1513

TABLE 5. S_r when the gap between light-emitting points is 2.

Obstacle number	Obstacle area	MCBP area	S_r
1	4	6.129	0.6526
2	9	12.5532	0.7169
3	12	16.276	0.7373
4	20	25.5505	0.7828

TABLE 6. L_r when the distance between light-emitting points is 2.

Obstacle number	Centroid of the obstacle	Centroid of MCBP	Centroid distance D	Equivalent circle radius R of the obstacle	L_r
1	(15,15)	(15.08,14.92)	0.1179	1.1287	0.1044
2	(15,15)	(15.02,15.22)	0.2211	1.693	0.1306
3	(15,15)	(14.59,14.77)	0.4687	1.9549	0.2398
4	(15,15)	(15.19,15.07)	0.2029	2.5238	0.0804

TABLE 7. S_r when the gap between light-emitting points is 3.

Obstacle number	Obstacle area	MCBP area	S_r
1	4	6.3412	0.6308
2	9	12.9413	0.6954
3	12	16.5877	0.7234
4	20	26.8914	0.7437

TABLE 8. L_r when the distance between light-emitting points is 3.

Obstacle number	Centroid of the obstacle	Centroid of MCBP	Centroid distance D	Equivalent circle radius R of the obstacle	L_r
1	(15,15)	(15,14.91)	0.09	1.1287	0.0782
2	(15,15)	(15.19,15.09)	0.2054	1.693	0.1213
3	(15,15)	(14.87,14.62)	0.4018	1.9549	0.2055
4	(15,15)	(14.8,15.49)	0.5224	2.5238	0.207

TABLE 9. S_r when the distance between light-emitting points is 4.

Obstacle number	Obstacle area	MCBP area	S_r
1	4	6.3579	0.6291
2	9	12.9566	0.6946
3	12	16.9355	0.7086
4	20	27.6948	0.7222

TIME COMPLEXITY. In [32], Yu *et al.* proposed an obstacle detection and recognition method based on ROI and multiple relevance vector machines in order to classify obstacles into four classes such as ditches, rocks, slopes and stones. In [33], Shahdib *et al.* use ultrasonic sensors and cameras to detect the distances between obstacles and robots, and then derive the widths and heights of the obstacles.

We conducted 4 sets of comparative experiments, and the areas of obstacles in groups 1-4 are 21540, 8670, 4853, and 40260, respectively. In the following experiments,

TABLE 10. L_r when the distance between light-emitting points is 4.

Obstacle number	Centroid of the obstacle	Centroid of MCBP	Centroid distance D	Equivalent circle radius R of the obstacle	L_r
1	(15,15)	(15,14.94)	0.06	1.1287	0.0532
2	(15,15)	(15.13,15)	0.1333	1.693	0.0788
3	(15,15)	(14.72,14.91)	0.2967	1.9549	0.1518
4	(15,15)	(15.33,15.23)	0.4082	2.5238	0.1617

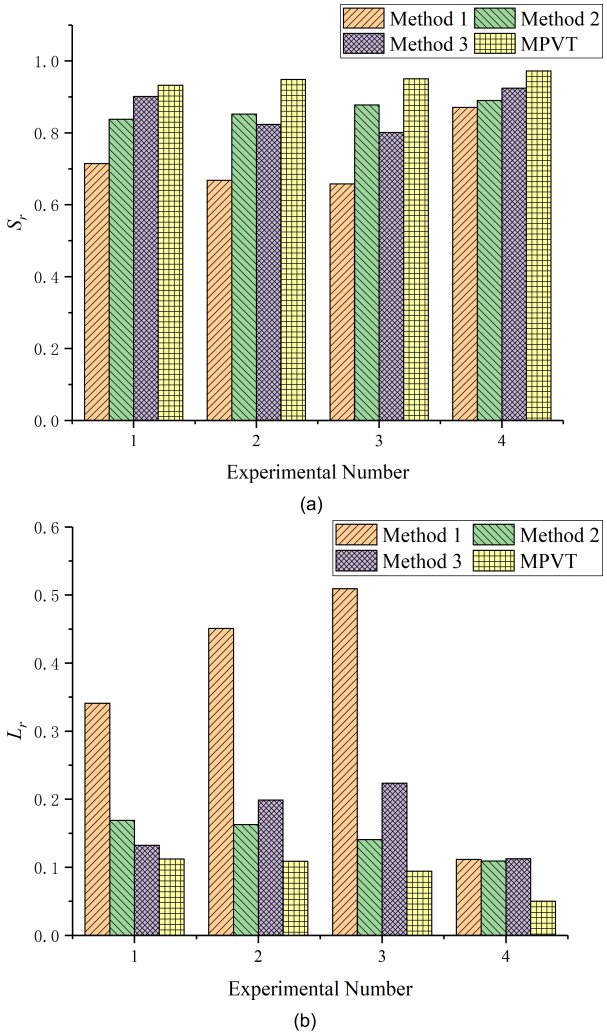


FIGURE 9. Comparison results of accuracy.

in order to make the areas of obstacles and the measurement region matchable, we set the experimental measurement areas of groups 1-4 to be squares with side lengths of 380*380, 270*270, 230*230, and 400*400, respectively. The gap between the light-emitting points is 2, and the obstacle is located at the center of the measurement region. The experimental setup for MPVT is also used in the experiments for the method in [31]. The experimental results are shown in Fig. 9.

In Fig. 9 and the following descriptions given below, Method 1 denotes the method presented in [32], Method 2 denotes the method presented in [33], and Method 3 denotes the method presented in [31].

As shown in Fig. 9(a), among these three methods, the S_r obtained by the MPVT algorithm is the largest, about 93%, which means that the area of MCBP is closest to that of the obstacle. As shown in Fig. 9(b), the maximum value of L_r obtained by the MPVT algorithm is about 10%, which means that the centroid interval between the obstacle and MCBP is the closest.

There are smaller $S_r s$ and larger $L_r s$ for Method 1 and Method 2 due to the following reasons. There are errors in the ROI regions extraction and the measurement for distances and lengths through intensity images, it leads to large errors when Method 1 is used for the positioning and measurement of obstacles. In addition, the method is based on the use of Time-of-Flight (ToF) cameras, which incurs a high cost. For Method 2, because of the limitation of the accuracy of ultrasonic measurement and the images' resolution, there are large errors in the localization of obstacles. Moreover, this method requires accurate reference objects in all measuring distances for the calculation of obstacle dimensions, and the ultrasonic rangefinding and the camera need to work in a stringently synchronized manner.

For Method 3, whether a grid where a light passes belongs to an obstacle or not is determined by the distance between the grid and the centroid of the obstacle. If the distance is less than the equivalent circle radius of the obstacle, the grid is a part of the obstacle, and vice versa. Even if a small part of the grid belongs to the obstacle, the whole grid area is counted, which results in a large error in S_r and L_r. Note that we set the obstacle to be located in the center of the measurement area in this experiment. Therefore, when the obstacle is located at other positions in the measurement area, S_r of MPVT will be better.

F. EXPERIMENT ON TIME COMPLEXITY

Because the method in [31] (i.e., Method 3) is similar to MPVT in terms of deployment environment, which is also based on sensor positioning and can measure the target shape, we compared the time complexity of MPVT with that of Method 3. The experiment steps are as follows:

Step 1: The measurement area is divided into 100∗100 grids, and the obstacle is set to a square with the area of 100;

Step 2: Traverse each viewpoint to obtain all the lights between the left side line and right side line of the blocking area from each viewpoint;

Step 3: For Method 3, traverse all the grids where lights pass, and use Equation (13) to determine whether there are

obstacles in the grid:

$$o_j = \begin{cases} 1, & if\ ||P_j - c_o|| < R_o \\ 0, & otherwise \end{cases} \quad (13)$$

where O_j denotes whether there is an obstacle in the j-th grid. If O_j is 1, there is an obstacle; otherwise, there is no obstacle. P_j represents the center coordinate of the j-th grid, C_o represents the centroid coordinate of the obstacle, and R_o represents the equivalent circle radius of the obstacle.

When the distance between the center of a grid and the centroid of the obstacle is less than the equivalent circle radius of the obstacle, it means there is an obstacle in the grid; otherwise, it means there is no obstacle in the grid.

For MPVT, it uses the steps presented in Section IV. IMPLEMENTATION PROCESS OF MPVT to locate MCBP of the obstacle.

The experimental results are as follows. The running time of Method 3 was 8.673 seconds, while the running time of MPVT is 1.249 seconds.

In this paper, the MPVT algorithm can get one side of the polygon in each step when calculating MCBP. After getting the last side of the polygon, MPVT finds the entire MCBP. However, the method in [31] needs to traverse each grid with lights before calculating and judging whether there are obstacles in the grid. The time complexity of the method in [31] is $O(n^3)$, which is greater than the $O(n^2)$ of MPVT.

Moreover, the polygon area that bounding the obstacles obtained from [31] is 130.2693, S_r is 0.7676, and L_r is 1.1145; the polygon area obtained by the MPVT algorithm in this paper is 106.7692, S_r is 0. 9366, and L_r is 0. 2683.

Therefore, the MPVT algorithm not only has high accuracy but also has low time complexity.

VI. CONCLUSION

In this paper, we propose and implement the MPVT algorithm for the problem of locating obstacles in the environments of sparsely-deployed sensors. The MPVT algorithm can recognize both the position and the area of the obstacle by constructing MCBP. MPVT includes the following four steps. Firstly, it gets the left side line and right side line of the blocking area of each light-receiving point; Secondly, it takes the left side line of the blocking area of the viewpoint (0.5, 0) as the reference line, and obtain the initial localization light according to VLT; Third, it uses the initial localization light as the reference line, and gets the first side. Fourth, it searches for all the other sides and vertexes of MCBP based on the law of counter-clockwise intersecting lights and related theorems proposed in this paper. We conducted the experiments to test the influence of obstacle orientation and sparseness of sensors, and compared the accuracy and time complexity with those of the existing methods.

In order to evaluate the localization deviation between an obstacle and its MCBP in terms of area sizes and positions, we propose two performance evaluation indexes, i.e., the area ratio S_r and the ratio of equivalent radius L_r, and give the calculation methods of S_r and L_r. We conducted the experiments to test the influence of obstacle orientation and sparseness of sensors, Experiment results show that MPVT has higher localization accuracy compared to the three baseline methods in [31], [32] and [33], and lower computing overhead.

As for the future work, we plan to extend MPVT to the scenario of multi-obstacles in order to make it suitable for the applications in complex environments. Moreover, how to extend MPVT to measure the size and positions of a moving obstacle when partial light-emitting nodes fail is also a research problem in real-world applications.

REFERENCES

[1] W.-T. Wang and K.-F. Ssu, "Obstacle detection and estimation in wireless sensor networks," *Comput. Netw.*, vol. 57, no. 4, pp. 858–868, Mar. 2013.

[2] L. Blazevic, J.-Y. Le Boudec, and S. Giordano, "A location-based routing method for mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 2, pp. 97–110, Mar. 2005.

[3] K. Yamazaki and K. Sezaki, "An ad-hoc routing protocol with obstacle evasion," in *Proc. 1st Int. Workshop Netw. Sens. Syst. Program (INSS)*, Tokyo, Japan, 2004.

[4] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[5] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-D lidar," in *Proc. IEEE Int. Conf. Inf. Autom.*, Lijiang, China, Aug. 2015, pp. 1648–1653.

[6] X. Yang, Z. Liu, W. Nie, W. He, and Q. Pu, "AP optimization for Wi-Fi indoor positioning-based on RSS feature fuzzy mapping and clustering," *IEEE Access*, vol. 8, pp. 153599–153609, 2020.

[7] X. Yang, R. Cao, M. Zhou, and L. Xie, "Temporal-frequency attention-based human activity recognition using commercial WiFi devices," *IEEE Access*, vol. 8, pp. 137758–137769, 2020.

[8] M. Zhou, Y. Wang, Z. Tian, Y. Lian, Y. Wang, and B. Wang, "Calibrated data simplification for energy-efficient location sensing in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6125–6133, Aug. 2019.

[9] M.-C. Kang, S.-H. Chae, J.-Y. Sun, J.-W. Yoo, and S.-J. Ko, "A novel obstacle detection method based on deformable grid for the visually impaired," *IEEE Trans. Consum. Electron.*, vol. 61, no. 3, pp. 376–383, Aug. 2015.

[10] D. Levi, N. Garnett, and E. Fetaya, "StixelNet: A deep convolutional network for obstacle detection and road segmentation," in *Proc. Brit. Mach. Vis. Conf.*, Swansea, U.K., 2015, pp. 1–12.

[11] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš, "Fast image-based obstacle detection from unmanned surface vehicles," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 641–654, Mar. 2016.

[12] R. H. Abiyev, M. Arslan, I. Gunsel, and A. Cagman, "Robot pathfinding using vision based obstacle detection," in *Proc. 3rd IEEE Int. Conf. Cybern. (CYBCONF)*, Exeter, U.K., Jun. 2017, pp. 1–6.

[13] K. Gupta, S. Upadhyay, V. Gandhi, and K. M. Krishna, "Small obstacle detection using stereo vision for autonomous ground vehicle," in *Proc. Adv. Robot. (AIR)*, New Delhi, India, 2017, pp. 1–6.

[14] H. Steendam, T. Q. Wang, and J. Armstrong, "Theoretical lower bound for indoor visible light positioning using received signal strength measurements and an aperture-based receiver," *J. Lightw. Technol.*, vol. 35, no. 2, pp. 309–319, Jan. 15, 2017, doi: 10.1109/JLT.2016.2645603.

[15] U. Nadeem, N. U. Hassan, M. A. Pasha, and C. Yuen, "Indoor positioning system designs using visible LED lights: Performance comparison of TDM and FDM protocols," *Electron. Lett.*, vol. 51, no. 1, pp. 72–74, 2015.

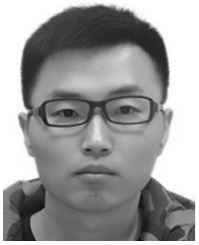
[16] B. Xie, G. Tan, and T. He, "Spinlight: A high accuracy and robust light positioning system for indoor applications," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, Seoul, South Korea, 2015, pp. 211–223.

[17] A. Hussein, P. Marin-Plaza, D. Martin, A. de la Escalera, and J. M. Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Gotenburg, Sweden, Jun. 2016, pp. 104–109.

[18] X. Zhang, M. Zhou, P. Qiu, Y. Huang, and J. Li, "Radar and vision fusion for the real-time obstacle detection and identification," *Ind. Robot: Int. J. Robot. Res. Appl.*, vol. 46, no. 3, pp. 391–395, May 2019.

[19] M. Singh, S. K. Bhoi, and P. M. Khilar, "Geometric constraint-based range-free localization scheme for wireless sensor networks," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5350–5366, Aug. 2017.
[20] H. Xiong, M. Peng, S. Gong, and Z. Du, "A novel hybrid RSS and TOA positioning algorithm for multi-objective cooperative wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9343–9351, Nov. 2018.
[21] H.-H. Pham, T.-L. Le, and N. Vuillerme, "Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor," *J. Sensors*, vol. 2016, pp. 1–13, Mar. 2016.
[22] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for obstacle detection with fully convolutional networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4296–4303.
[23] V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, and J. W. Jeon, "Learning framework for robust obstacle detection, recognition, and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1633–1646, Jun. 2017.
[24] G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Cochin, India, Jul. 2017, pp. 1–6.
[25] W. Guan, X. Zhang, Y. Wu, Z. Xie, J. Li, and J. Zheng, "High precision indoor visible light positioning algorithm based on double LEDs using CMOS image sensor," *Appl. Sci.*, vol. 9, no. 6, p. 1238, Mar. 2019.
[26] W. Guan, S. Chen, S. Wen, Z. Tan, H. Song, and W. Hou, "High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning," *IEEE Photon. J.*, vol. 12, no. 2, pp. 1–16, Apr. 2020.
[27] H. Luo, Z. Guo, K. Wu, F. Hong, and Y. Feng, "Energy balanced strategies for maximizing the lifetime of sparsely deployed underwater acoustic sensor networks," *Sensors*, vol. 9, no. 9, pp. 6626–6651, Aug. 2009.
[28] T. Ojha, S. Misra, and M. S. Obaidat, "SEAL: Self-adaptive AUV-based localization for sparsely deployed underwater sensor networks," *Comput. Commun.*, vol. 154, pp. 204–215, Mar. 2020.
[29] X. Shan and J. Tan, "Mobile sensor deployment for a dynamic cluster-based target tracking sensor network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 1452–1457.
[30] H. M. Ammari, "Joint K-coverage and data gathering in sparsely deployed sensor networks–Impact of purposeful mobility and heterogeneity," *ACM Trans. Sensor Netw.*, vol. 10, no. 1, pp. 1–33, 2013.
[31] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 621–632, May 2010.
[32] H. Yu, J. Zhu, Y. Wang, W. Jia, M. Sun, and Y. Tang, "Obstacle classification and 3D measurement in unstructured environments based on ToF cameras," *Sensors*, vol. 14, no. 6, pp. 10753–10782, Jun. 2014.
[33] F. Shahdib, M. W. Bhuiyan, M. K. Hasan, and H. Mahmud, "Obstacle detection and object size measurement for autonomous mobile robot using sensor," *Int. J. Comput. Appl.*, vol. 66, no. 9, pp. 28–33, 2013.

ZHIGANG GAO received the Ph.D. degree from Zhejiang University, China, in 2008. He is currently an Associate Professor with Hangzhou Dianzi University. His current research interests include mobile computing and cyber-physical systems.



XIAOWEI YANG received the B.S. degree in process equipment and control engineering from Anhui Polytechnic University. He is currently pursuing the M.S. degree in computer science and technology with Hangzhou Dianzi University. His research interests include data processing, neural networks, and route planning.

BO WU (Member, IEEE) received the Ph.D. degree in human sciences from Waseda University, Tokyo, Japan, in 2015. He is currently an Assistant Professor with the Department of Human Informatics and Cognitive Sciences, Waseda University. His current research interests include human motion capture, eye-movement analysis, machine learning and Big data analysis, and human–centered application system development.

HUIJUAN LU received the Ph.D. degree from the China University of Mining and Technology. She is currently a Professor with China Jiliang University. Her research interests include mobile computing and artificial intelligence.

JIANHUI ZHANG (Member, IEEE) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2008. He is currently a Full Professor with Hangzhou Dianzi University. He is also the Leader of the IoT Group. His research interests include IoT, energy-harvesting microsystems and networking, algorithm designing and applications, and crowdsourcing.

WENJIE DIAO received the B.S. degree in network engineering from the Fujian University of Technology. He is currently pursuing the M.S. degree in computer technology with Hangzhou Dianzi University. His interests include mobile computing, mobile phone security, and neural networks. His current research interests include mobile user behavior analysis and mobile user authentication.

QUN JIN (Senior Member, IEEE) is currently a Professor with the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Japan. He has been extensively engaged in research works in computer science, information systems, and social and human informatics. His recent research interests include human–centric ubiquitous computing, behavior and cognitive informatics, big data, personal analytics, individual modeling, intelligence computing, block-chain, cyber security, cyber-enabled applications in healthcare, and computing for well-being. He is a Foreign Member of the Engineering Academy of Japan (EAJ).