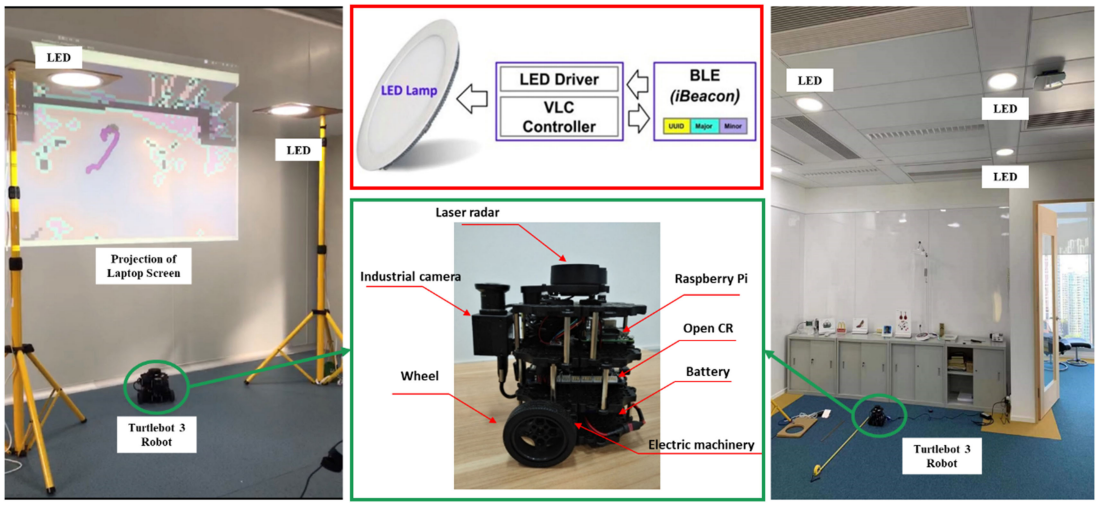
IEEE Photonics Journal
An IEEE Photonics Society Publication

Open Access

High-Accuracy Robot Indoor Localization Scheme Based on Robot Operating System Using Visible Light Positioning

Volume 12, Number 2, April 2020

Weipeng Guan
Shihuan Chen
Shangsheng Wen
Zequn Tan
Hongzhan Song
Wenyuan Hou



DOI: 10.1109/JPHOT.2020.2981485

High-Accuracy Robot Indoor Localization Scheme Based on Robot Operating System Using Visible Light Positioning

Weipeng Guan,[1,2] **Shihuan Chen**,[3] **Shangsheng Wen**,[1] **Zequn Tan**,[3] **Hongzhan Song**,[3] **and Wenyuan Hou**[1]

[1]School of Materials Science and Engineering, South China University of Technology, Guangzhou 510641, China
[2]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China
[3]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China

DOI:10.1109/JPHOT.2020.2981485
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Manuscript received February 14, 2020; revised March 9, 2020; accepted March 14, 2020. Date of publication March 18, 2020; date of current version April 13, 2020. This work was supported in part by the National Undergraduate Innovative and Entrepreneurial Training Program under Grants 201510561003, 201610561065, 201610561068, 201710561006, 201710561054, 201710561057, 201710561058, 201710561199, and 201710561202; in part by the Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation ("Climbing Program" Special Funds) under Grants pdjh2017b0040 and pdjha0028, and in part by the Guangdong Science and Technology Project under Grant 2017B010114001. (*Guan Weipeng and Chen Shihuan are co-first authors of this article.*) Corresponding authors: Weipeng Guan; Shangsheng Wen (augwpscut@mail.scut.edu.cn; shshwen@scut.edu.cn).

Abstract: Visible light positioning (VLP) is widely believed to be a cost-effective answer to the growing demand for service-based indoor positioning. Meanwhile, high accuracy localization is very important for mobile robots in various scenes including industrial, domestic and public transportation workspace. In this paper, an indoor robot VLP localization system based on Robot Operating System (ROS) is presented for the first time, aiming at promoting the application of VLP in mature robotic system. On the basis of our previous researches, we innovatively designed a VLP localization package which contains the basic operation control of the robots, the features extraction and recognition of the LED-ID, cm-level positioning, and robust dynamic tracking algorithms. This package exploited the proposed lightweight algorithm, distributed framework design, the loose coupling characteristics of the ROS, and the message communication methods among different nodes. What's more, an efficient LED-ID detection scheme is proposed to ensure the lightweight and accuracy of the positioning. A prototype system has been implemented on a Turtlebot3 Robot[1]. Experimental results show that the proposed system can provide robot indoor positioning accuracy within 1 cm and an average computational time of only 0.08 s.

Index Terms: Visible light positioning (VLP), robot operating system (ROS), robot localization, high accuracy positioning, thresholding scheme, distributed framework

1. Introduction

Nowadays, visible light communication (VLC) has attracted attention from academia and industry since it has many advantages over the traditional radio frequency, including wide unregulated

[1]Experiment Demonstration is available at: https://kwanwaipang.github.io/Image/ROS.mp4

spectrum, high security, and low cost. With the rapid development of wireless networks and the popularity of mobile terminals, location service-based indoor positioning technology has broad application prospects. The indoor positioning technology based on VLC (also named as visible light positioning, VLP) employs the LED lamps as the transmitter to transmit the ID information, and the positioning terminal realizes indoor positioning through the photodetector (PD) or the camera. In contrast to the traditional indoor positioning technologies, such as Wi-Fi, Bluetooth, RFID, inertia navigation, and so on [1]–[4], the VLP has the advantages of high positioning accuracy, no electromagnetic interference, and environmental friendly [5], [6], [28]. At present, VLP methods can be divided into two categories according to the type of the receiver: PD-based [7] and image sensor-based [8], [29]. PD is not an ideal positioning device for mobile terminals, because the positioning based on PD will cause large errors due to angle measurement, received signal strength measurement, light intensity variation or the other reasons, and the accuracy of PD is largely dependent on the direction of beam. As for the image-based method, although it requires corresponding image processing technology and is limited by the field of view of the image sensor, it can obtain more stable signal and has stronger anti-interference ability as the spatial separation characteristics of lens in the image sensor [9]. Furthermore, with the popularity of image sensors installed on commercial intelligent terminals, such as mobile phones and robots. VLP based on image sensors has broader prospect and more commercial potential than the PD-based method, thus having aroused the attention of many experts and scholars in the world [8], [10]–[14].

Robots are required to do more difficult and intelligent works, the application scenario of robots becomes more complex and pluralistic. The robots might also operate in cluttered environments and complex structures. Scenarios inside domestic or industrial workspaces can conceal different and difficult operating situations. Hence, indoor navigation accuracy of such robots could be significantly affected by these kinds of operation and environmental conditions. In order to enhance the performance of robots during their works, the localization system of robots is required to be accurate and simultaneous. Over the years, the localization of mobile robots has been well studied. Dynamic estimation method where the position information depends on the outputs of the controller is one of the most popular estimation methods [15]. However, only relying on noumenal sensing is easy to have drift, having a bad impact on the autonomy-level of industrial task, which requires localization in the environment for a long time without interruption. Beyond that, WiFi-based positioning is also a common solution for robot positioning [16]. But owing to its inherent defect, WiFi-based indoor positioning has been limited to precision problems. Even if the data fusion of multiple access points is carried out through particle filtering, the positioning error is still above 1m. Another framework for robot platforms based on image and laser sensors is presented in [17], in which Monte Carlo Localization (MCL) algorithm and CNN algorithm are used to realize average accuracy of 40 cm. However, unlike human beings that have perfect visual system, it is hard for robots to accept more than decimeter level errors. And the complexity of the positioning algorithm also determines the location delay and the hardware cost. Based on this view, as a kind of absolute positioning, VLP should be the most potential technology for the robot indoor positioning with its high performance [8], [14], [18]. For example, in our previous works, we realized an accuracy of 4.38 cm by using 3 LEDs in [8], and average positioning error of 0.82 cm through 2 LEDs and the proposed calibration algorithm in [14]. In [10], it achieves an accuracy of 7.5 cm with mobile phone. In [13], average positioning error of 17.52 cm can be achieved by using single LED.

In this paper, we novelty introduce VLP into the robot positioning system for the first time. More specifically, we designed a VLC localization and navigation package based on the distributed framework of the Robot Operating System (ROS), which contains the basic operation control of the robots, LED-ID detection, dynamic cm-level VLP algorithms, and so on. We have proposed LED-ID detection method based on feature recognition using Machine Learning (ML) [19]–[21], cm-level VLP algorithm just using 2 LEDs [18], anti-occlusion and anti-interference dynamic LED-ROI tracking algorithms[22], [23], in our prior demonstrations, but all of those works are limited and independently verified by simple experimental setup. For example, as for the dynamic tracking algorithm, we use the tracking error of the LED-ROI to simulate the theoretical positioning error in our previous works [22], [23]. Although the tracking error can reflect the performance of the

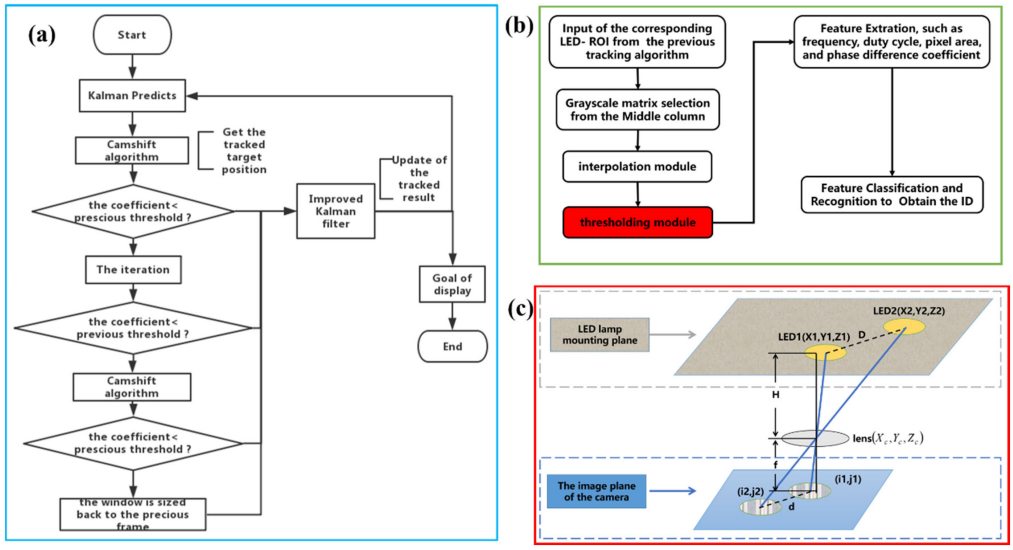


Fig. 1. The architecture and workflow of the proposed VLP algorithm. (a) The dynamic tracking algorithm for LED-ROI [22]; (b) The proposed efficient LED-ID detection scheme; and (c) The cm-level positioning algorithm using 2 LEDs [18].

algorithm, it is still no experimental verification with the positioning algorithm. Besides, as for the LED-ID detection algorithm, the modulation and recognition method has been firstly proposed by our previous work [19] using the ML for the feature recognition. With the generalization ability of the ML, the modulation and recognition method based on ML has good robustness. However, ML and image processing methods in [19]–[21] are too complexity, which create positioning latency for mobile robot. Therefore, lightweight and robustness LED-ID detection scheme is important. Therefore, in this paper, we integrate these effective algorithms organically in the form of ROS nodes by the design of distributed framework and the effective message communication methods among the nodes. What's more, a novel lightweight and efficient LED-ID detection scheme is proposed for the LED-ID features detection, aiming at reducing positioning latency. These features ensure both high speed and high accuracy for indoor navigation and more comprehensive verification of the effectiveness of the algorithms when compared to our prior works. The rest of this paper is organized as follows. The design of the proposed VLP package based on ROS is introduced in Section 2. And the experimental results are presented in Section 3.

2. The Workflow and Lightweight Image Processing

As shown in Fig. 1, the architecture of the proposed VLP algorithm is mainly comprised of three parts: dynamic tracking algorithm for LED-ROI, the features of LED-ID extraction and detection, and high precision positioning algorithm just using double LEDs. Our main contributions are the design of the distributed VLP node (in Section 3) and the lightweight image processing algorithm to ensure real-time operation for the mobile robot. In this section, we firstly introduce the pipeline of the proposed VLP and the proposed efficient LED-ID detection scheme.

First of all, by controlling the on-off state of the LED, ID data can be modulated onto the emitting visible light. The corresponding LED-ID detection can be realized by exploiting the rolling shutter effect of the CMOS camera. The raw movie file captured by the CMOS image sensor is converted into grayscale format, in which 255 and 0 represent full brightness and darkness respectively. Then, the image (800 $\times$ 600) showed in Fig. 2(a) can be observed. After that, the LED-ROI tracking method (as shown in Fig. 1(a)) is used to obtain the ROI of each LED, and a 1×28 matrix of grayscale levels can be obtained as shown in Fig. 2(b) and (c) (blue curve) after directly

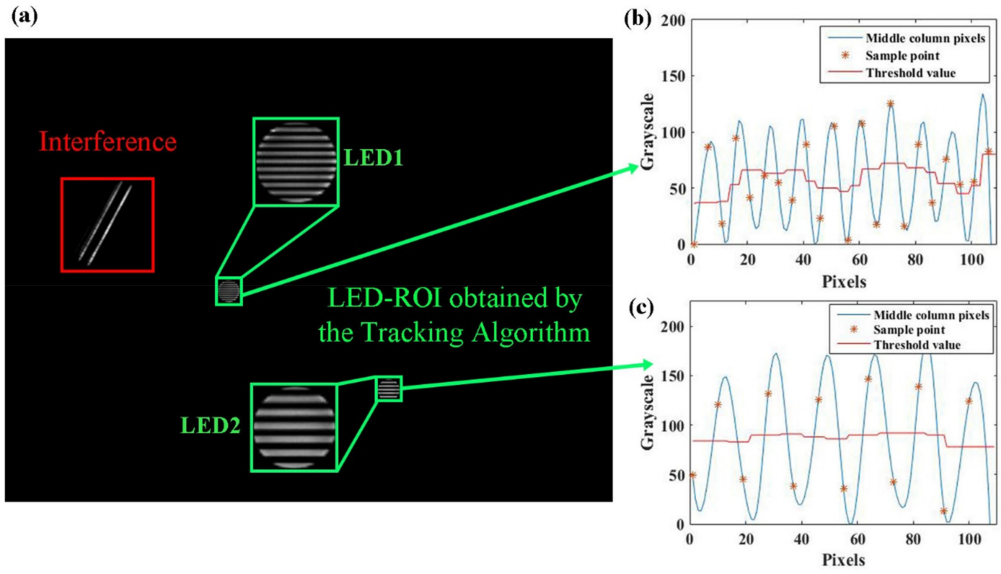


Fig. 2. (a) The original frame obtained by the camera, and the LED-ROI is tracking by [22]; (b) and (c) Experimental grayscale value pattern and the applied LEXW scheme (red curve) for two LED-ROI.

selecting the middle column pixels of ROI area. For the LED-ID detection, unlike the traditional optical camera communication (OCC) method [24], [25], [30], we use the OOK-PWM modulation method to introduce different features to the light and dark stripes captured in the image for different LED-ID, rather than directly demodulation (as shown in Fig. 1(b)). It can avoid the synchronize process, increase the accuracy of ID detection, and improve the robustness through the features detection and recognition method (have been verified in [9], [19]–[21]).

In general, more than 70% of the time-cost in the whole VLP scheme will be spent on ID-detection [9]. In order to reduce the positioning latency, we cannot use the complexity ML methods proposed in [19]–[21] for the LED-ID feature detection and classification. Without the strong generalization ability of the ML, it is very important to detect LED-ID feature or recover the logic light and dark stripes accurately, as the LED-ID features recognition performance is usually limited by the number of pixel row per bit. In our previous work [26], we verified that just changing the LED luminaire with more uniformity light output, the blooming effect could be removed. However, there is still a high data fluctuation owing to different illumination falling into different parts of the CMOS image sensor, which would affect the logic decision for the LED-ID features recognition. Therefore, two novel processes are employed as indicated in the "interpolation module" and "thresholding module" boxes in Fig. 1(b). Firstly, in order to increase the number of effective features for feature extraction, Cubic Spline Interpolation (CSI) is proposed and applied. Hence, the 1×28 matrix is expanded to 1×109 matrix after the proposed interpolation module.

Apart from the CSI, in order to properly defining the logic 1 and 0 to obtain accurate feature information in this highly fluctuated grayscale value pattern, we propose the efficient Local Extremum based on Sliding Window (LEXW), as shown in Fig. 3. The LEXW is to find the local maximum points and local minimum points in the expanded matrix in turn. The average value between the local maximum and local minimum is the threshold. We firstly define a sliding window, which is a matrix of $1 \times n$ (n is the sampling rate), slides from the current extreme point. Then, finding the maximum point of the 1×109 matrix (the first local maximum); where $P_{\max}$ and $G_{\max}$ represent the position and the corresponding grayscale value of the maximum point. Next step is to find the local minimum, which is done by increasing or decreasing the pixel position of the last element of the sliding window P_{slide} (initialized to $P_{slide} = P_{\max} + n$ for increasing, and $P_{slide} = P_{max}$ for decreasing) and its corresponding grayscale value is G_{slide}. A reference threshold T_r is set to avoid the fluctuation

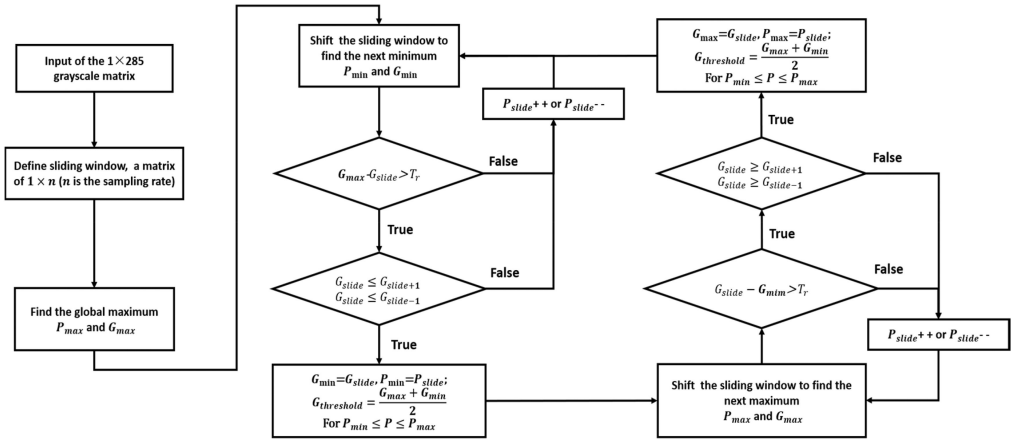


Fig. 3. Flow diagram showing the working principle of the proposed LEXW scheme.

caused by noises. When the local minimum is determined, the threshold value is the average of local maximum and local minimum; its pixel row location and grayscale value are $G_{threshold}$ and P respectively. The next procedure is almost the same but finding local maximum instead of local minimum. This loop will stop until the sliding window is shifted to the edge of the 1×109 matrix. Fig. 2(b) and (c) shows the LED-ROI grayscale value pattern and the applied LEXW scheme (red curve). After the LEXW, grayscale levels turn into data logics. Then, the features of the LED-ID, such as the frequency, duty cycle, pixel area, and phase difference coefficient proposed in [3]–[5], can be obtained for the features recognition to detect the LED-ID. Finally, the global coordinates of the LEDs can be obtained to calculate the positioning of the terminal through the positioning algorithm in [18].

3. The Design of VLP Package Based on ROS

In this section, we introduce the design of the proposed VLP package based on the ROS with the distributed framework concept. ROS is an open source Meta-Operating System for robots. It includes hardware abstraction layer similar to operating systems. However, unlike conventional operating systems, it can be used for numerous combinations of hardware implementation. Furthermore, it is a robot software platform that provides various development environments specialized for developing robot application programs. Last but not the least, it is also a well-equipped framework which integrates a lot of basic function for the control of mobile robot.

ROS is developed in unit of nodes, which is the minimum unit of executable program that has broken down for the maximum reusability. The node exchanges data with other nodes through messages forming a large program as a whole. The proposed VLP package is composed of four nodes, and we name them as: MVCAM Node, CamKF Node, ID-Detection Node, and VLC2Locator Node, respectively. There are two different methods of exchanging messages: a topic which provides a unidirectional message transmission/reception, and a service which provides a bidirectional message request/response, in the proposed VLP package. The structure of the proposed VLP package and the message communication methods among those nodes are shown in Fig. 4.

3.1 The Design of MVCAM Node

The proposed cm-level VLC positioning method can use 2 LEDs as the base-station and one CMOS camera as receiver to achieve 3D positioning, whose principle can be referred to our

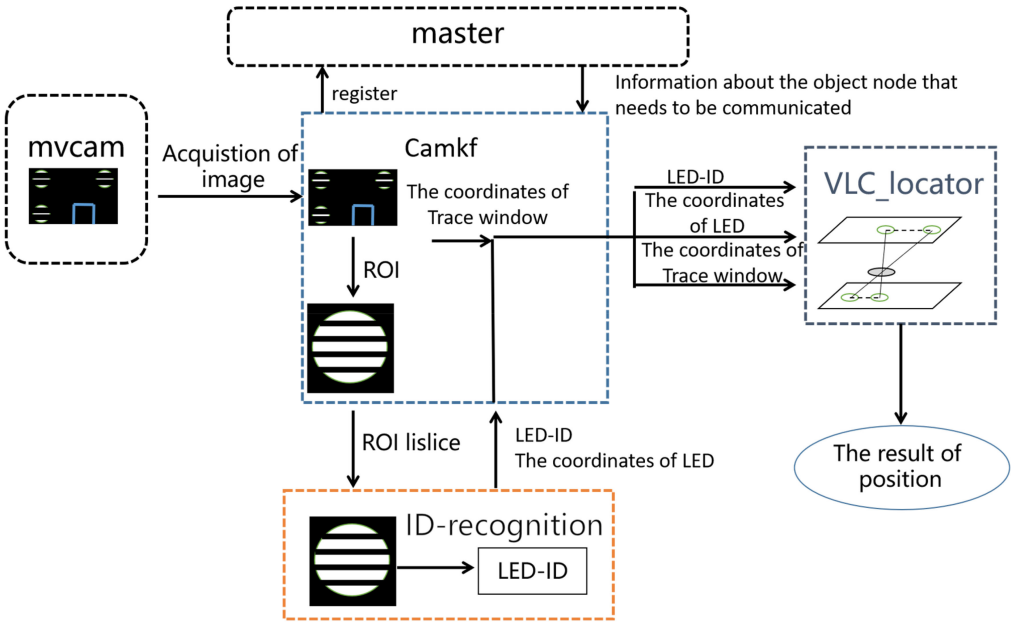


Fig. 4. The structure and message communication methods of the proposed VLP package.

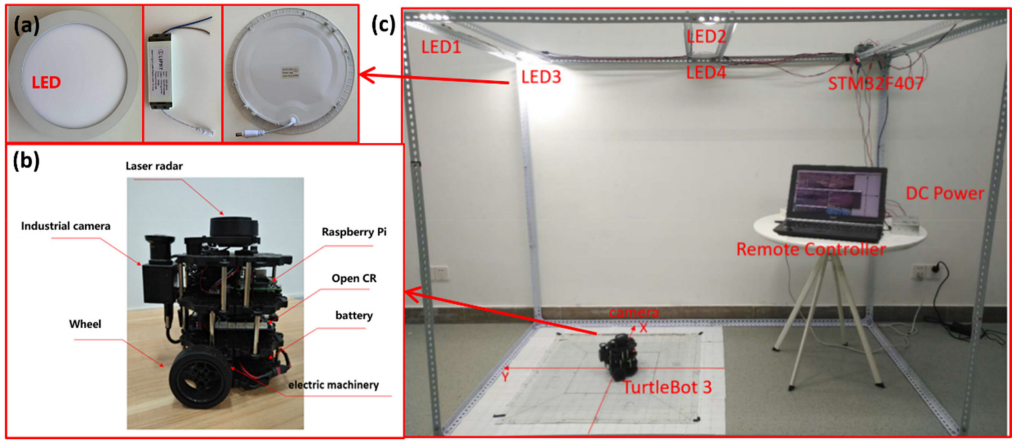


Fig. 5. Experimental platform of the proposed VLP package based on robot. (a) The LED lamp; (b) The Turtlebot 3 robot; and (c) The Environment 1.

previous work [18]. The control and program of the CMOS camera to get the images, which is the source of the LED-ID and position information, is the basic of the whole VLP package. The pseudo-code and design concept of this node is shown in Code 1. The CMOS industrial camera is installed vertically on the turtlebot3 (as shown in Fig. 5(b)) to capture the light and dark stripes from the LEDs installed on the ceiling, through the rolling shutter effect by designing the related Application Programming Interface (Line 2-8 in Code 1). We name processing and control program of this CMOS camera as MindVision camera node (MVCAM), which converts the video to ROS image messages and then publishes it to the “camera/image” topic. The topic in ROS is an asynchronous method which uses the same type of message for both publisher and subscriber [9], [27]. The subscriber node receives the information of publisher node corresponding to the identical topic name (the “camera/image” topic) registered in the master. As topic is unidirectional

and remain connected to continuously send or receive messages, it is suitable for sensor data that requires publishing messages periodically. Based on this information, we use the “camera/image” topic to transmit the message from the CMOS camera, and the other nodes can directly connect the MVCAM Node to receive the image messages.

Code 1 MVCAN

```
Step 1. Setup the camera
CameraSdkInit(1);//Camera initialization
CameraSetAeState (hCamera,FALSE); //Set to custom mode
CameraSetAnalogGain (hCamera,100); //Set the analog gain to 100
CameraSetExposureTime (hCamera,200);//Set the exposure time to 200 microseconds
CameraSetIspOutFormat(hCamera,CAMERA_MEDIA_TYPE_MONO8);//Set the image output format to 8-bit gray
CameraSetMonochrome (hCamera,TRUE); //Enable the color to black and white function
CameraSetFrameSpeed (hCamera,2000); //Set camera output image frame rate to very high.
Set the Camera parameters;
Step 2. Get the image and publish it as the ROS message
image_transport::Publisher pub =it.advertise("camera/image", 1);
//Set the image publisher and the topic is camera/image
ros::Rate r(fps);
while (nh.ok())
{
   CameraGetImageBuffer(hCamera,&sFrameInfo,&pbyBuffer,1000);  //Gets a frame of image data.
   CameraImageProcess(hCamera, pbyBuffer, g_pRgbBuffer,&sFrameInfo);
   //Process images according to preset parameters.
   resize(cvarrToMat(iplImage),frame,Size(800,600),0,0,INTER_NEAREST);
   //Scale the image to 800×600 pixels
   sensor_msgs::ImagePtr msg = cv_bridge::CvImage(std_msgs::Header(), "bgr8", frame).toImageMsg();
   //Convert images from OpenCV format to ROS image messages via CvBridge
   pub.publish(msg); //publish  a  message
   ros::spinOnce();
   r.sleep();
}
```

3.2 The Design of CamKF Node

After receiving the raw image message using rolling shutter effect from the CMOS camera, we obtain the LED-ROI through the proposed CamKF node whose pseudo-code and design concept are shown in Code 2. The proposed CamKF node represents “LED-ROI Tracking and Extraction Method using Improved Camshift-Kalman Algorithm”, which has been firstly proposed in our prior work [22]. A structure is defined in this node, whose members include Vector tracking window, LED-ID, and the x and y coordinates of the tracking window (detected LED-ROI) in the image. The proposed CamKF Node is the central administration of the proposed VLP package based on ROS (as shown in Fig. 4). Using the CamKF Node to extract the LED-ROI areas in the images based on the “camera/image” topic which is published from the MVCAM Node. After that, the extract LED-ROI areas are published, and the ID identification request is immediately issued to the LED-ID detection node (ID-Detection Node in Code 3) through the “ID_recognition_srv” service. Service is a bidirectional synchronous communication between the service client requesting a service and the service server responding to the request [27]. A service consists of a service server (ID-Detection Node) that responds only when there is a request and a service client (CamKF Node) that can send requests (ID identification request) as well as receiving responses (the result of LED-ID detection). Service does not maintain the connection, so we use the service here to reduce the load of the network rather than using the topic. During the whole process, the LED-ID message (from the ID-Detection Node) and the coordinates of the tracking window (LED-ROI message from the CamKF Node) are encapsulated in the customized “LED_info _srv” service, which can send positioning calculation requests to the VLC positioning node (VLC2Locator Node in Code4). As the robot is moving, if a new unmarked LED appears in the subsequent process, the ID identification request is immediately issued to the ID-Detection Node again

through the “ID_recognition_srv”, which can update the LED-ID message in the “LED_info _srv” service.

Code 2 CamKF

```
Struct LED A, B, C; //Three lamps A,B and C are defined by the structure;
Int main //Image processing main function
{
    image_sub_ = it_.subscribe("/camera/image", 1, & IMAGE_LISTENER_and_LOCATOR::convert_callback,
    this); //Define the image receiver
  ros::Rate loop_rate();
  while (ros::ok())
  {
     {
       cv_ptr =   cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::RGB8);
       While
          {
             ROI_process; // traversing all ROI, looking for ROI;
             ros::ServiceClient client = nh.serviceClient< ID_recognition:: ID_recognition >(“get_image”);
             ID_recognition:: ID_recognition ID_recognition_srv;
             // Declare a service called ID_recognition_srv that USES the ID_recognition service file
             ID_recognition_srv.request::sensor_msgs::ImagePtr msg = cv_bridge::CvImage(std_msgs::Header(),
             "bgr8", frame).toImageMsg(); //OpenCV image conversion to ROS image format
             srv.response.ID = ID; // get the ID by responding
           }
     }
     Tracking
     {
       ros::ServiceClient ros_tutorials_service_client =nh.serviceClient<LED_info:: LED_info >("LED_info_srv");
LED_info:: LED_info LED_info _srv; // declare a service called LED_info_srv that USES the LED_info service file
Tracking(frame, trackObjectNum, frameIndex); // track lamp image, calculate tracking window
 LED_info_srv.request.A.ID = A.ID
     }
ros::spin();
   }
}
```

3.3 The Design of ID-Detection Node

This node responds to the ID identification request of the CamKF Node, using the proposed LED-ID detection method and the efficient LED-ID detection scheme (mentioned in Section 2) to recognize the LED-ID in LED-ROI, and return the results to the CamKF Node. The pseudo-code of the node is as follows:

Code 3 ID-Detection

```
bool get_image(ID-Detection::rgbd_image::Request &req, ID-Detection::ID::Response &res)
// service response function
{
   ID = Image_process（req. rgbd_image);
   res.ID=ID； // take the ID information as the service response
   ROS_INFO("success");
   return true;
}
int main
{
   ros::ServiceServer service = n.advertiseService("get_image", get_image);// define service server "get_image"
   //Process the image and get the ID
   ros::Rate loop_rate();
   while (ros::ok())
   {
    ros::spin();// waiting for service request
   }
```

3.4 The Design of VLC2Locator Node

The VLC2Locator Node is the positioning coordinate calculation node. It subscribes the messages from the “LED_info _srv” service of the CamKF Node, and then, returns the positioning result and as the interface of the proposed VLP package to the basic operation control of the robots through the XML/RPC calls under the TCP/IP protocol. The pseudo-code of the VLC2Locator node is as follows:

Code 4 VLC2Locator

```
Struct LED A, B, C;
//Three lamps A,B and C are defined by the structure;
bool position(ros_tutorials_service::SrvTutorial::Request &req, ros_tutorials_service::SrvTutorial::Response &res)
{
   Req.A.ID=A.ID
   Req. A_ Img_local_X =A. A_ Img_local_X
   ROS_INFO("sending back response", (int)res.result);// response information
   return true;
}
ros::ServiceServer service = nh.advertiseService("LED_info _srv", position);
// define the service server and execute the position function.
Get_coordinate(LED A, LED B, LED C) // coordinates are calculated according to the position of the luminaire.
{
   Global_coordinate_judgment (LED A, LED B, LED C);//Choose two lamps with different X and Y coordinates.
   //Assign to LED U1, LED U2.
   pose_value= Calculate (LED U1, LED U2);
   return pose_value;
}
int main() // locate the main function
{
   ros::Publisher pub = nh_.advertise ("location", 1000); // define a location publisher
   {
      ros::Rate loop_rate();
      while (ros::ok())
      {
         pose_value=Get_coordinate(LED A, LED B, LED C);// positioning according to lamp information
         pub.publish(msg); // publish location information;
      }
   ros::spin();
   }
}
```

4. Experiment

4.1 Experiment Setup

Experiments with field tests have been conducted to evaluate the performance of the proposed indoor navigation system in terms of positioning accuracy and positioning speed. The experimental platform and the related specifications are shown in Fig. 5 and Table 1. A MindVision UB-300 industrial CMOS camera is installed vertically on the Turtlebot3 whose processor is Pi 3 Model B with Quad ARM Cortex-A53 Core 1.2 GHz Broadcom BCM2837 64 bit CPU and 1 GB RAM. In general, we need two computers for the ROS development because the netbook in the robot generally doesn’t have sufficient resources to handle the graphic requirements. In the Turtlebot3, the Raspberry Pi 3 is the netbook in the robot, as its weak processing power, most of the nodes of the VLP package including the image processing, and the basic operation control of the robots are run at the remote control laptop (the Workstation), except the MVCAM Node and the motor drive node. Therefore, the video captured by the MVCAM Node should be transmitted to the laptop through the Local Area Network (LAN), and the size of the captured image influence the real-time

TABLE 1
Parameters of the Experimental Platform

CMOS Camera	
Model	MindVision UB-300
Spectral Response Range/nm	400~1030
Pixel(H × V)	2048 × 1536
Resolution used in the package	Downsample to 800 × 600
Pixel Size/μm2	3.2 × 3.2
Time of Exposure/ms	0.02
Type of Shutter Acquisition Mode	Electronic Rolling Shutter
Acquisition Mode	Successive and Soft Trigger
Turtlebot3 Robot	
Module	Raspberry Pi 3 B
CPU	Quad Core 1.2 GHz Broadcom BCM2837
RAM	1 GB
operating system	Ubuntu mate 16.04
Remote Controller	
Module	Acer VN7-593G
CPU	Quad Core Intel® Core™ i7-7700HQ
RAM	16 GB
operating system	Ubuntu 16.04
The Size of the Experimental Platform (L ×W× H)/ cm^3	
Environment 1	100 × 100 × 150
Environment 2	170 × 170 × 180
Environment 3	190 × 190 × 285
LED Coordinates for Environment 1	
Coordinates (x, y, z)/cm	LED1 (-465,495,150) LED2 (-460,-420,150) LED3 (460,490,150) LED4 (480,-425,150)

ability, which would be discussed in Section 4.3. Last but not the least, the LED lamp (Fig. 5(a)) used in our experiment is from the LIPHY COMPANY in Hong Kong. It is an integrated LED lamp with VLP function and the Bluetooth, which makes the VLC data (LED-ID) / frequency can be easily configured by the Android mobile phone.

Robot needs a map for positioning and navigation, so we need to create a map and give it to the robot, or the robot should be able to create a map by itself. In this paper, we use the Laser Distance Sensor (LDS) in the Turtlebot 3 to convert 3D information into 2D information of the XY plane. After building the map, the VLP package and the other nodes related to motion control can be run directly in the Turtlebot 3. Fig. 6 is the user interface (UI) of the proposed VLP package based on ROS robot in Rviz which is the primary 3D visualization tool for ROS [27]. The Display setting is used to control the displayed information, and the Perspective setting in the right of the UI is used to set the viewing angle. The image shown at the bottom left of the UI is the real-time images of two LEDs captured by CMOS camera installed in the robot. The middle part of this UI is the main interface of the robot visualization, showing the current map and the VLP position estimation results. The map was constructed by SLAM and indicated the locations of the four columns and walls corresponding to the experimental Environment 1. The purple dots are the VLP positioning results of the mobile robot, the gray block is the model of robot with camera. The latest position estimation results can be seen in the local close-up, which is exactly in line with the position of the industrial camera model on the robot. While the green arrow indirectly reflects the position of the robot obtained by the lidar and IMU using the Adaptive Monte Carlo localization (AMCL) fusion localization.

It is worth to mention that, without loss of generality, in order to test the performance of our proposed VLP package based on ROS, we also setup another two robot indoor positioning system in different environments (as can be shown in Fig. 7). Environment 2 is the scene on the left side of

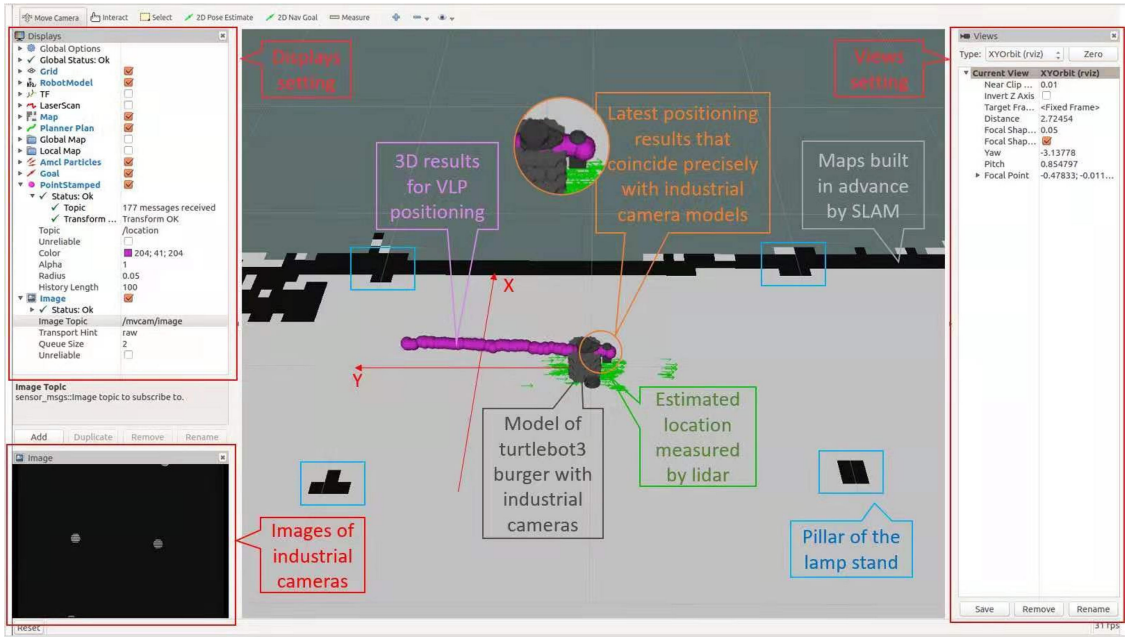


Fig. 6. The UI of the proposed VLP package based on ROS robot.

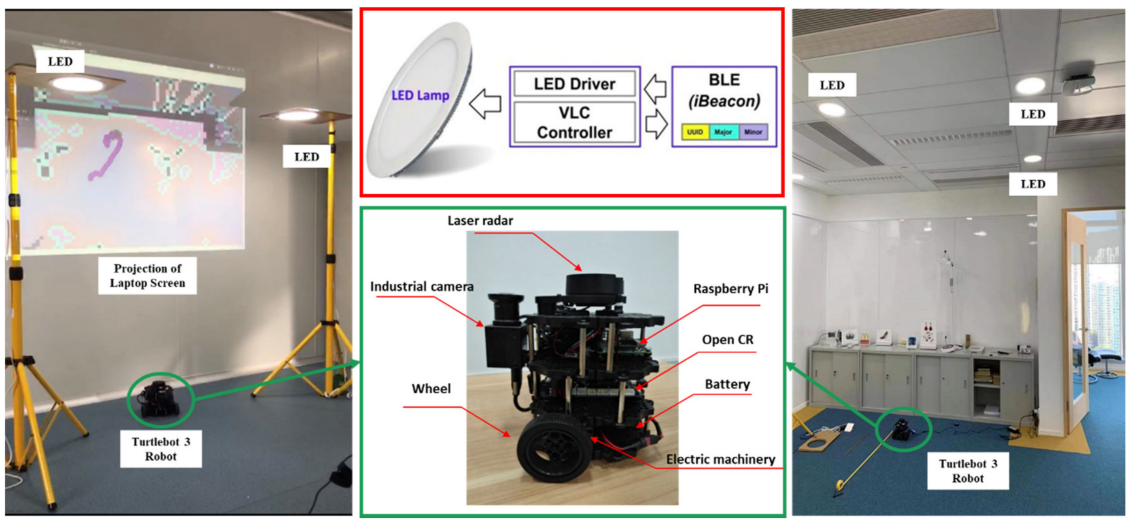
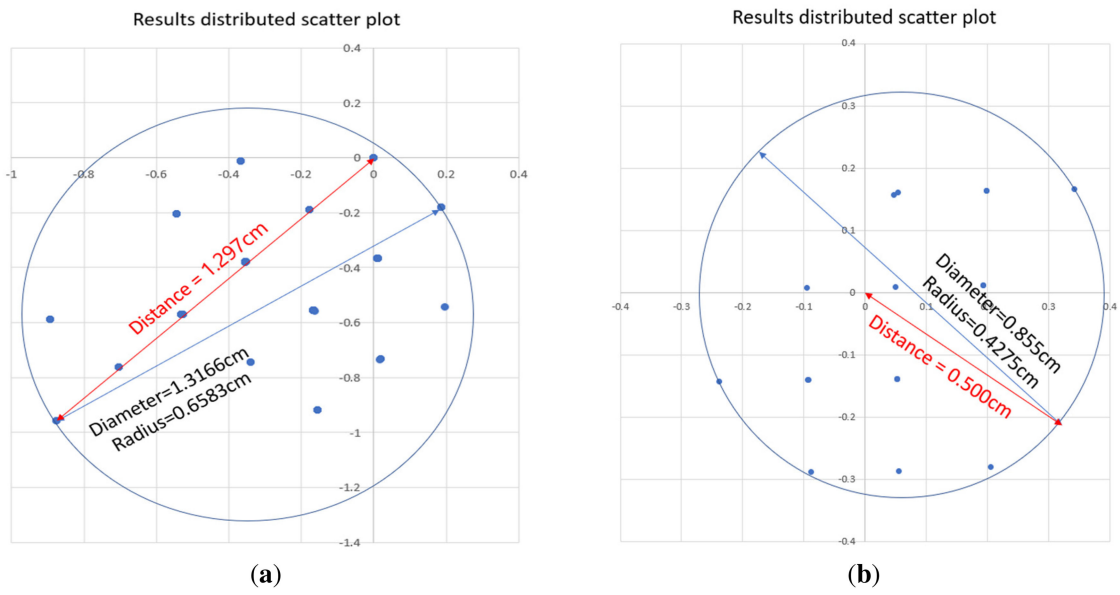


Fig. 7. Other robot localization systems established using the proposed VLP package based on ROS.

this figure, which has been accepted for the Demo Zone at the 2020 Optical Fiber Communication Conference (OFC). Environment 3 is the scene on the right side of this figure, which is a bigger room with the height of 2.85 m in an office. For those two environments, we just list the cases to show reproducibility and effectiveness of our proposed VLP package based on ROS. For the readers who are interested in those two setups, please refer to our OFC demo in March 2020. In this paper, we just analysis the positioning performance of the Environment 1. A video demo is



(a) (b)

Fig. 8. The stability of stationary positioning. (a) Without positioning error calibration method; (b) With positioning error calibration method.

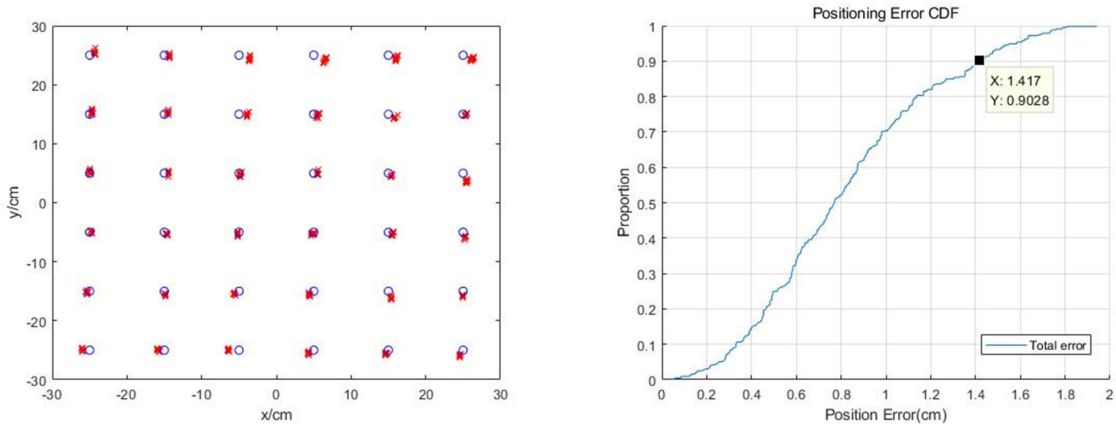
shown at the footer of the first page of this paper, which shows the demo of the proposed VLP package in robot localization and navigation.

4.2 Positioning Accuracy

In general, numbers of locations were randomly chosen in the experimental field of the VLP accuracy verification, then, the positioning error for each location was calculated by comparing the actual spatial position and the estimated position. We name it as stationary positioning. Here, we innovatively proposed to use Dispersion Circle (DC) to test the accuracy and stability of the stationary positioning. Since all the positioning results of one test point are distributed in the DC, which can be used to evaluate the distribution of the positioning result. The Dispersion Radius of the DC is the maximum fluctuation of positioning results, while the Euler distance between the center of the DC and the actual position of the test point is the average positioning accuracy. The locations of the robot is kept still at the coordinate of (0, 0) to test the DC performance, which can be seen in Fig. 8(a). The red line in Fig. 8(a) represents the positioning accuracy of the proposed VLP package is 1.297 cm, and the fluctuation positioning error (the radius of the DC) would be within 0.6583 cm. Through the positioning error calibration method proposed in our previous work [14], the positioning accuracy of the proposed VLP package can be improved to 0.5 cm with a fluctuation of 0.4275 cm(as shown in Fig. 8(b)).

What's more, 36 uniformly distributed test locations are randomly chosen in the experimental field. Each test point is measured 12 times, obtaining in a total of 432 stationary positioning results, which can be seen in Fig. 9. More than 90% positioning errors are less than 1.417 cm, and the average positioning error is 0.82 cm.

As for the mobile robot which is moving, it is really hard to know the exact actual position of the robot without using complexity motion capture system. As we just give the command for straight movement at a fixed speed to the robot in our experiments, which means that if the initial position and orientation of the robot are not exactly aligned, the actual trajectory of the robot would deviate from the given command trajectory. Then the estimated trajectory of the VLP package would deviate from the given command trajectory, which is a normal experimental phenomenon.



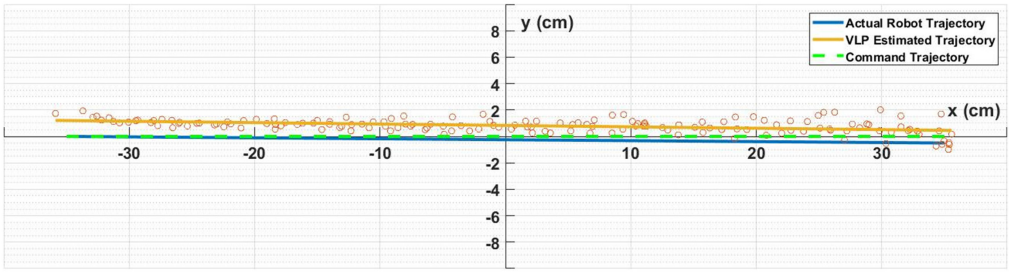
(a) (b)

Fig. 9. The accuracy of stationary positioning. (a) The actual position of the test points and the estimated position of the proposed VLP package; (b) The cumulative distribution function (CDF) curves of positioning errors.

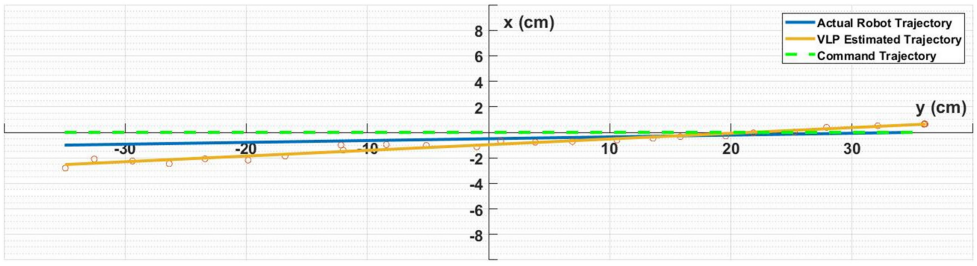
Therefore, we record the starting position and the ending position of the robot to represent the actual trajectory of the robot. We calculate the positioning error through comparing the command trajectory, actual trajectory of the robot and the estimated trajectory estimated by the proposed VLP package. For the dynamic positioning, we just test two different commands for the robot: moving from (−35, 0) to (35, 0) and moving from (0, −35) to (0, 35), which can be seen in Fig. 10(a) and (b) (More dynamic positioning results can be seen in the video demo). The robot is moving along the given command trajectory (the green line) at a speed of 4 cm/s, while blue line is the actual trajectory of the robot obtained by measuring the start and end points of the robot. The yellow line is the estimated trajectory of the proposed VLP package, which is obtained by fitting the VLP estimated results with the least square method. To evaluate the accuracy of the dynamic positioning, we sampled the test points from the green line and the yellow line. More specifically, as the exact actual position of a moving robot is not available, we just only use the command trajectory which controls the actual trajectory of the robot as the reference to calculate the dynamic positioning errors. Although the actual trajectory of the robot have errors comparing the given command, the errors between the VLP estimated trajectory and the command trajectory is still persuasive. The CDF of the dynamic positioning errors can be seen in Fig. 10(c). It is worth to mention that, the position results of the VLP package is the coordinates of the center of CMOS camera, which can be converted to the center of the robot through the Transform Frame (TF) transformation.

4.3 Real-Time Performance

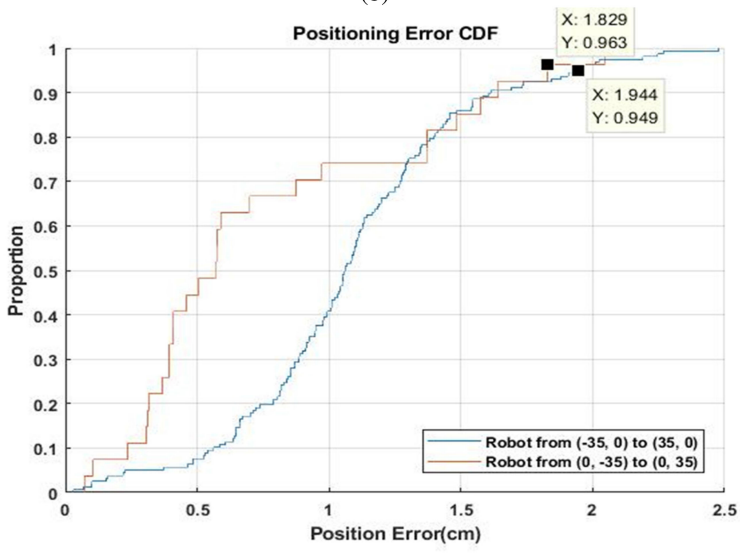
Positioning speed is another important factor that affects positioning system. Low positioning speed hinders the efforts of adopting indoor positioning in practice, especially for the mobile robot. Therefore, lightweight and effective process algorithm is very important to reduce the positioning latency. In this section, the average positioning time is tested to evaluate the effectiveness of the structure and message communication methods of the proposed VLP package. As what has been mentioned in Section 4.1, due to the weak processing power of the Raspberry Pi in the Turtlebot 3, most of the proposed VLP package process including: the positioning calculation, image process, and so on, can't be proposed in the Raspberry Pi directly. Therefore, the image captured by the CMOS camera in the robot need to be transmitted to the remote laptop to process. However, the transmission of the "camera/image" topic to the laptop through the network would cause time-cost which lead to the positioning latency. To test this conjecture, the average positioning time is tested



(a)



(b)



(c)

Fig. 10. The accuracy of dynamic positioning. (a) The experiment result of robot moving from (−35, 0) to (35, 0); (b) The experiment result of robot moving from (0, −35) to (0, 35); (c) The CDF curves of positioning errors in (a) and (b).

in different setup (two variables: the resolution of the captured image and the transmission medium between the robot and laptop), whose results can be seen in Table 2.

The positioning time of the proposed VLP package based on robot was measured carefully. The positioning time is started at image captured in the CMOS camera node and ended at the position estimation, including all of the algorithm process and messages communication among different nodes in the whole ROS system. We set the timer and counter in the program, the average positioning time is obtained by 5 minutes positioning and navigation experiment. As can be seen in Table 2, when the USB line is used as the transmission medium between the robot and laptop, the average positioning time can be reduced from 0.4 s to 0.08 s. It means that if a more powerful processer is used in the robot rather than the Raspberry Pi, the all of the nodes of

TABLE 2
Average Positioning Time

Experiment	(A)	(B)	(C)
Resolution of the Captured Image	800×600	800×600	2048×1536
Transmission Medium between the robot and laptop	WiFi	USB Cable	USB Cable
Average Positioning Time (s) / each poistioning	0.4018	0.0780	0.3253

the proposed VLP package can directly run in the robot platform rather than the laptop, then the average positioning time for each positioning of the proposed VLP package should be within 0.08 s. With the improvement of the processing capacity of Turtlebot based on ROS, the proposed VLC package will be processed directly on the robot. Only the positioning results will be fed back to the remote control book.

5. Conclusion

In this paper, an indoor robot VLP positioning package based on ROS is proposed. Detailed explanations are given to its design of the VLP package and the message communication among different nodes in the ROS. What's more, an efficient LED-ID detection scheme for rolling shutter-patterning is proposed, which not only can be used for the features of LED-ID recognition but also is suitable for the thresholding of conventional OCC. A prototype system has been implemented on a Turtlebot3 Robot, and the experiment evaluate that the proposed system can provide robot indoor positioning accuracy within 1 cm and an average computational time of only 0.08 s. This study pushed forward the development of VLP technology application in robots and lay a foundation for transplanting to other ROS robot platforms, showing the great potential of VLC localization for indoor robot positioning.

References

[1] P. Keikhosrokiani, N. Mustaffa, N. Zakaria, and M. I. Sarwar, "Wireless positioning techniques and location-based services: A literature review," in *Lecture Notes in Electrical Engineering*, Berlin, Germany: Springer, 2013.
[2] Y. Chouchang and S. Huai-rong, "WiFi-based indoor positioning," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 150–157, Mar. 2015.
[3] F. Topak, M. K. Pekericli, and A. M. Tanyer, "Technological viability assessment of bluetooth low energy technology for indoor localization," *J. Comput. Civil Eng.*, vol. 32, no. 5, Sep. 2018, Art. no. 04018034.
[4] S. Park and S. Hashimoto, "Autonomous mobile robot navigation using passive RFID in indoor environment," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2366–2373, Jul. 2009.
[5] Z. Dongfang, C. Gang, and A. F. Jay, "Joint measurement and trajectory recovery in visible light communication," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 247–261, Jan. 2017.
[6] T. Do and M. Yoo, "An in-depth survey of visible light communication based positioning systems," *Sensors*, vol. 16, no. 5, pp. 1–40, May 12, 2016.
[7] G. Weipeng, W. Yuxiang, W. Shangsheng, C. Hao, C. Yang, and Z. Zhang, "A novel three-dimensional indoor positioning algorithm design based on visible light communication," *Opt. Commun.*, vol. 392, pp. 282–293, Jun. 1, 2017.
[8] W. Guan, S. Wen, L. Liu, and H. Zhang, "High-precision indoor positioning algorithm based on visible light communication using complementary metal–oxide–semiconductor image sensor," *Opt. Eng.*, vol. 58, no. 2, 2019, Art. no. 024101.
[9] W. P. Guan, "Research on high precision indoor visible light positioning algorithm based on image sensor," M.S. thesis, Dept. Control. Eng., South China Univ. of Technology, Guangzhou, China, 2019.
[10] J. Fang *et al.*, "High-speed indoor navigation system based on visible light and mobile phone," *IEEE Photon. J.*, vol. 9, no. 2, Apr. 2017, Art. no. 8200711.
[11] W. Guan, S. Wen, H. Zhang, and L. Liu, "A novel three-dimensional indoor localization algorithm based on visual visible light communication using single LED," in *Proc. IEEE Int. Conf. Autom., Electron. Elect. Eng.*, 2018, pp. 202–208.
[12] Y. Q. Ji *et al.*, "A single LED lamp positioning system based on CMOS camera and visible light communication," *Opt. Commun.*, vol. 443, pp. 48–54, 2019.
[13] R. Zhang, W. Zhong, Q. Kemao, and S. Zhang, "A single LED positioning system based on circle projection," *IEEE Photon. J.*, vol. 9, no. 4, Aug. 2017, Art. no. 7905209.

[14] S. Chen *et al.*, "High accuracy and error analysis of indoor visible light positioning algorithm based on image sensor," Nov. 2019, *arXiv:1911.11773*.
[15] B. J. Stephens, "State estimation for force-controlled humanoid balance using simple models in the presence of modeling error," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3994–3999.
[16] R. Miyagusuku, A. Yamashita, and H. Asama, "Data information fusion from multiple access points for WiFi-based self-localization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 269–76, Apr. 2019.
[17] S. Xu *et al.*, "A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with monte carlo localization," *Sensors*, vol. 19, no. 2, pp. 1–19, Jan. 2019.
[18] W. Guan *et al.*, "High precision indoor visible light positioning algorithm based on double LEDs using CMOS image sensor," *Appl. Sci.*, vol. 9, no. 6, 2019, Art. no. 1238.
[19] X. Canyu, G. Weipeng, W. Yuxiang, F. Liangtao, and C. Ye, "The LED-ID detection and recognition method based on visible light positioning using proximity method," *IEEE Photon. J.*, vol. 10, no. 2, Apr. 2018, Art. no. 7902116.
[20] J. Li and W. Guan, "The optical barcode detection and recognition method based on visible light communication using ML," *Appl. Sci.*, vol. 8, no. 12, 2018, Art. no. 2425.
[21] Guan W *et al.*, "The detection and recognition of RGB-LED-ID based on visible light communication using convolutional neural network," *Appl. Sci.*, vol. 9, no. 7, 2019, Art. no. 1400.
[22] W. Guan, Z. Liu, S. Wen, H. Xie, and X. Zhang, "Visible light dynamic positioning method using improved camshift-kalman algorithm," *IEEE Photon. J.*, vol. 11, no. 6, Dec. 2019, Art. no. 7906922.
[23] W. Guan, X. Chen, M. Huang, Z. Liu, Y. Wu, and Y. Chen, "High-speed robust dynamic positioning and tracking method based on visual visible light communication using optical flow detection and Bayesian forecast," *IEEE Photon. J.*, vol. 10, no. 3, Jun. 2018, Art. no. 7904722.
[24] C. W. Chen *et al.*, "Efficient demodulation scheme for rolling-shutter-patterning of CMOS image sensor based visible light communications," *Opt. Express*, vol. 25, no. 20, pp. 24362–24367, 2017.
[25] C. W. Chow, C. Y. Chen, and S. H. Chen, "Visible light communication using mobile-phone camera with data rate higher than frame rate," *Opt. Express*, vol. 23, no. 20, pp. 26080–26085, 2015.
[26] G. Weipeng, W. Yuxiang, X. Canyu, F. Liangtao, L. Xiaowei, and C. Yingcong, "Performance analysis and enhancement for visible light communication using CMOS sensors," *Opt. Commun.*, vol. 410, pp. 531–551, Mar. 1, 2018.
[27] A. Martinez and E. Fernández, *Learning ROS for Robotics Programming [M].* Birmingham, U.K.: Packt Publishing Ltd., 2013.
[28] C. H. Chang *et al.*, "A 100-Gb/s multiple-input multiple-output visible laser light communication system," *J. Lightw. Technol.*, vol. 32, no. 24, pp. 4121–4127, Dec. 2014.
[29] Y. Chen *et al.*, "Optoelectronic properties analysis of silicon light-emitting diode monolithically integrated in standard CMOS IC," *Chinese Phys. B*, vol. 28, no. 10, 2019, Art. no. 107801.
[30] K. Xu, "Silicon MOS optoelectronic micro-nano structure based on reverse-biased PN junction," *Physica Status Solidi (a)*, vol. 216, no. 7, 2019, Art. no. 1800868.