Journal of Network and Computer Applications 146 (2019) 102425

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Three-dimensional robot localization using cameras in wireless multimedia sensor networks☆

Sheng Feng [a,*], Shigen Shen [a], Longjun Huang [a], Adam C. Champion [b], Shui Yu [c], Chengdong Wu [d], Yunzhou Zhang [d]

[a] Department of Computer Science and Engineering, Shaoxing University, Shaoxing, 312000, China
[b] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, USA
[c] School of Software, University of Technology Sydney (UTS), NSW, Australia
[d] Faculty of Robot Science and Engineering, Northeastern University, Shenyang, 110819, China

ARTICLE INFO

Keywords:
Indoor environment
Wireless multimedia sensor networks (WMSNs)
Recursive least squares localization (RLS)
Real-time data aggregation
Distributed architecture

ABSTRACT

We consider three-dimensional (3D) localization in wireless multimedia sensor networks (WMSNs) and seek optimal localization accuracy in order to ensure real-time data fusion of mobile robots in WMSNs. To this end, we propose a real-time 3D localization algorithm realized by a distributed architecture with various smart devices to overcome network instability and the bottleneck channel at the coordinator. We then employ the recursive least squares (RLS) algorithm to fuse the 2D image coordinates from multiple views synchronously in WMSNs and determine the mobile robot's 3D location in an indoor environment. To minimize wireless data transmission, we also develop a distributed architecture that combines various smart devices by defining the data content transmitted from multiple wireless visual sensors. Moreover, we analyze the factors influencing the network instability of various smart devices, and factors influencing the localization performance of mobile robots in a multiple-view system. Experimental results show the proposed algorithm can achieve reliable, efficient, and real-time 3D localization in indoor WMSNs.

1. Introduction

The recent progress in wireless visual sensors, dynamic localization of mobile robots in wireless multimedia sensor networks (WMSNs), and three-dimensional (3D) localization in computer vision, has also attracted considerable attention to 3D localization in WMSNs. WMSN technology enables various smart devices to connect to such networks and achieve user-defined data transmission and 3D localization of mobile robots (Nandhini et al., 2018; Piasco et al., 2018; Al-Turjman and Radwan, 2017). Typical applications of 3D localization in WMSN include monitoring various areas, public events, private property, and perimeters, all of which can improve the quality of our lives (Al-Turjman and Radwan, 2017).

However, the numerous smart devices in WMSNs generate large volumes of multimedia data, and raise many challenges. Of primary concern is the computational complexity of 3D localization algorithms, because low-cost smart devices can delay and corrupt WMSNs, causing incomplete data delivery, insufficient memory, and reduced computing capability (Cui et al., 2018). To address this issue, a desirable design strategy for a distributed computing is to achieve real-time localization under QoS constraints (C. Labs; You et al., 2015).

The triangulation algorithm is a typical method for 3D localization in computer vision. It has been widely used in fields such as industrial applications in multi-sensor networks (Maisano and Mastrogiacomo, 2016), Unmanned Aerial Vehicles (UAVs) (Tanathong and Lee, 2014; Rupnik et al., 2015; Osa et al., 2013), industrial applications of non-contact 3D metrology and inspection (Mavrinac et al., 2015), laser light techniques (Hahne et al., 2018), reconstruction of 3D points (Kang et al., 2014), calibration (Otero and Sánchez, 2015), and practical applications of calibration and triangulation (Bouguet, 2008). However, triangulation solutions, which requires exactly three 2D coordinates from three cameras' fields of view, are unsuitable for WMSNs due to their

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail addresses: fengsheng_13@aliyun.com (S. Feng), fengsheng_13@163.com (S. Feng), shigens@usx.edu.cn (S. Shen), hlj_jlh@163.com (L. Huang), champion.17@osu.edu (A.C. Champion), shui.yu@uts.edu.au (S. Yu), wuchengdong@mail.neu.edu.cn (C. Wu), zhangyunzhou@ise.neu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.jnca.2019.102425
Received 30 November 2018; Received in revised form 14 May 2019; Accepted 21 August 2019
Available online 27 August 2019

particular characteristics. First, methods for deploying visual sensors and monitoring areas of interest are the essential problems. Unlike triangulation algorithms that can select 2D coordinates from cameras with cables, many wireless visual sensors in WMSNs have unstable network communications; thus, making it difficult to select exactly three 2D coordinates for data fusion. Second, it is very difficult to flush video streams generated by multiple views into sink nodes via wireless communications, forming a new challenge for WMSN architecture design. This characteristic makes triangulation solutions better suited to cable video networks in computer vision.

The least squares (LS) algorithm can be considered suitable for solving the issue of computational complexity in WMSNs (Wu et al., 2013; Feng et al., 2014). The LS algorithm can work within a distributed architecture consisting of various smart devices and compensate for the weaknesses of triangulation solutions in the WMSNs' visual sensors because it can fuse data from multiple cameras' views without restricting the number of cameras or environmental noise levels (Piasco et al., 2018). Integrating WMSNs with the LS algorithm realizes large-scale capabilities for processing and transmitting wireless data as well as fusing data synchronously from multiple views in WMSNs. This integration enables the 3D localization of mobile robots in WMSNs; thus eliminating the computational burden bottleneck produced by triangulation solution.

In this paper, we employ a distributed architecture consisting of various smart devices to achieve real-time 3D localization of mobile robots in WMSNs. To the best of our knowledge, synchronous real-time data fusion of detected 2D image coordinates of mobile robots from multiple views of wireless visual sensors has not been well studied. We list our main contributions below:

First, we propose a recursive least squares (RLS) algorithm to determine mobile robots' 3D locations accurately in real time by fusing the detected 2D image coordinates from multiple wireless visual sensor views.

Second, we introduce the collaborative calibration method to determine the intrinsic and extrinsic parameters of multiple cameras synchronously with the assistance of two planes from the calibration board.

Third, we develop a distributed architecture that includes various smart devices and assign computational tasks to many sensors in order to minimize the computational burden.

The rest of this paper is structured as follows. In Section 2, we review related works. In Section 3, we describe our RLS algorithm for 3D localization in WMSNs. In Section 4, we construct our RLS localization of mobile robots in indoor environments. In Section 5, we validate our experimental results and present the distributed architecture for WMSNs. Finally, Section 6 concludes the paper.

2. Related works

3D Localization: Various approaches to 3D localization have been proposed. For multiple-view scenes, Pahwa et al. (2018) focused on 3D object localization in a RGB-D video sequence that integrated depth information with state-of-the-art 2D object localization techniques to improve object localization per frame. For generic scenes, using only 2D object detections from multiple view images, Rubino et al. (2018) considered the 3D localization problem as the estimation of 3D quadrics given known matrices and a set of 2D ellipses fitted to the object detection bounding boxes in multiple views. The authors showed that object localization in 3D can be instantiated as a conics optimization problem with a closed-form solution in the dual space.

WMSNs: Initially, Bhatt and Datta (2016) defined surveillance with wireless multimedia sensors. They considered critical event surveillance using audio and video sensors to monitor events occurring in a given area. Next, Ahmed (2017) introduced compression complexity and real-time data routing. He achieved adaptive traffic shaping of multimedia streaming using multipath forwarding with dynamic cost calculation for next-hop selection. Heng et al. (2017) studied the critical conditions for bandwidth-intensive multimedia data processing and proposed a distributed image compression architecture for WMSNs to prolong the overall network lifetime. Kim et al. (2017) studied the collapse of constructed WMSNs due to sensor failure and showed that network could recover with a minimum number of sensors. Moreover, Villamizar et al. (2018) considered rapidly building discriminative classifiers to learn and detect object categories. The authors' boosting strategy, which picked the most discriminative object combination, enabled them to classify objects as categories rather than just instances.

Nandhini et al. (2018) adopted compressed-sensing-based background subtraction for energy-efficient anomaly detection with single and multiple objects using reduced complexity and energy in WMSNs by applying a novel strategy using frame measurement differencing with an adaptive threshold to increase the network lifetime. Their approach can reduce the volume of information to be transmitted by representing the video signal via a small number of measurements. The authors' evaluation of the performance of their approach found that it achieved better detection accuracy and required 90% fewer samples. Ur Rehman et al. (2016) studied how to transmit complete images in WMSNs while minimizing the node energy and image content. The authors considered two variants: a robust object appearance detector that transmits image data only when required and removing redundant features from an image to minimize energy consumption during data transmission.

Simultaneous Localization and Mapping (SLAM): However, due to recent advances in simultaneous localization and mapping (SLAM) technology, mobile sensors can be applied to improve 3D localization. Kim et al. (2018) considered the problem of occlusions with conventional static laser scanning methods. They proposed a 2D Hector SLAM technique to perform real-time estimates of a robot's position and orientation in the x-y plane. The information is used to create 3D point clouds of unknown environments in real time to determine possible robot's navigation paths. Given a set of RGB-D sensors, Chen et al. (2018) introduced a prototype indoor mapping solution that calibrates multiple RGB-D sensors to construct a sensor array with a large field of view (FoV). The proposed RGB-D camera array overcomes the shortcomings of single-RGB-D-camera indoor mapping systems resulting in a greater degree of coverage and efficient collection of a 3D indoor point cloud.

Structure from Motion (SFM): For structure from motion (SFM) problems, Hauke et al. (Strasdat et al., 2012) presented visual SLAM, a real-time SFM method, and analyzed sequential visual SLAM's performance rigorously. Considering various scenes and motion patterns, they concluded that key frame bundle adjustment outperformed filtering methods. For monocular SLAM applications, Raúl et al. (Mur-Artal et al., 2015) chose ORB features, which performed well for place recognition tasks, and proposed the ORB-SLAM method that could operate in various environments in real-time. They designed a system using the same features for all SLAM tasks that featured excellent robustness and achieved unprecedented performance compared to other state-of-the-art monocular SLAM approaches. To address problems with omnidirectional or wide-FoV fisheye cameras, David et al. (Caruso et al., 2015) presented an extension of large-scale direct SLAM (LSD-SLAM) that formulated tracking and mapping for the unified omnidirectional model. LSD-SLAM was suitable for all types of central projection systems, such as fisheye and catadioptric cameras.

In computer vision, SFM research derives from photogrammetry. The SFM and SLAM methods have the same characteristics: they estimate sensor motion by modelling the previously unknown but static environment. SFM can solve problems such as projective geometry and optimization, and SLAM can estimate the robot motion with or without cameras. However, most SFM work suggests that the observed environment is static (Strasdat et al., 2012). For moving objects, localization accuracy can be jeopardized. Muhamad et al. (Saputra et al., 2018) investigated SFM techniques for dynamic environments. They provided comprehensive approaches for the segmentation and tracking of dynamic objects and discussed the advantages and disadvantages of

each approach from the perspectives of practicality and robustness. Our proposed method belongs to the SFM research area because we employ multi-view visual sensors to estimate the 3D locations of color markers using key frames.

Localization with Mobile Sensors/Cameras: Several studies exist that address localization via mobile sensors and cameras. Ham and Yoon (2018) researched the problem of localizing distant objects in any built environment (both indoors and outdoors) in which both the global positioning system (GPS) and wireless networks are unavailable. The authors integrated information from mobile devices' accelerometers, gyroscopes and cameras to determine the 3D locations of distant objects. Wang and Ji (2018) considered replacing intrusive and subject-unfriendly calibration with their proposed implicit calibration method. Champion et al. (2013) studied how to identify significant topics of common interest among proximate people. They proposed a distributed mobile communications system to discover such topics to foster more significant conversations. Regarding the feasibility of camera self-calibration, Javier et al. (Civera et al., 2009) presented a sequential Bayesian structure to estimate full camera calibration from a sequence of fixed but unknown calibrations. The authors employed a sum of Gaussians filter to address the nonlinearities in unknown calibration parameters; their approach achieved complete fixed camera calibration using a sequence of uncalibrated monocular images. The authors also successfully applied their achievement to multi-view situations such as monocular mapping (Concha et al., 2015), visual odometry (Civera et al., 2010) and multi-view depth fusion (Fácil et al., 2017).

3. Recursive least squares (RLS) localization

The complexity of real-world environments makes it challenging to detect potential targets while producing minimal false alarms. PIXY sensors adopt the color algorithm to detect single color information (Rowe et al., 2007; Djelouah et al., 2015). Due to rotation and shape invariance, we localize a red helmet that we fix atop the tripod on the mobile robot. The helmet's color information discriminates it from other objects in the environment, and the helmet's elevation is above most obstacles in the area. Even when the mobile robot remains in one place, the color information can be detected successfully. The signature $s = 1$ indicates that multiple visual sensors can identify the mobile robot without image matching.

3.1. Collaborative calibration with multiple views

We adopt four cameras in each room, each with a 60° angle of view, for which the pinhole imaging principle applies.

The αth visual sensor's intrinsic parameters are denoted as follows: the focal length is fc^{α}, the principal point is cc^{α}, the skew coefficient is ℓc^{α}, and the distortion is kc^{α}, where $\alpha \in$ (Otero and Sánchez, 2015).

We briefly review the method for projecting from the image coordinates to normalized coordinates.

If the image coordinates of the detected centroid from the αth visual sensor are $p = (x_p^{\alpha}, y_p^{\alpha})$, the normalized image projection is (x^{α}, y^{α}).

Using the pinhole imaging model, the image coordinates are

$$\begin{cases} x_p^{\alpha} &= \dfrac{fc^{\alpha} X_c^{\alpha}}{Z_c^{\alpha}} \\ y_p^{\alpha} &= \dfrac{fc^{\alpha} Y_c^{\alpha}}{Z_c^{\alpha}}, \end{cases} \tag{1}$$

where $(X_c^{\alpha}, Y_c^{\alpha}, Z_c^{\alpha})$ is the camera coordinate of the 3D point P.

Assuming $r^2 = (x^{\alpha})^2 + (y^{\alpha})^2$, the new normalized point coordinate, x_d, with lens distortion is

$$x_d = (1 + kc^{\alpha}(1)r^2 + kc^{\alpha}(2)r^4 + kc^{\alpha}(5)r^6)x_n^{\alpha} + dx, \tag{2}$$

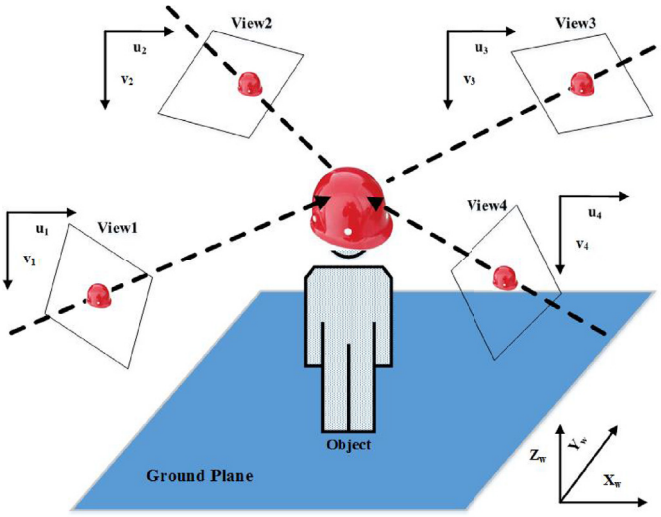


Fig. 1. Recursive least squares localization.

where the tangential distortion vector dx is

$$dx = \begin{bmatrix} 2kc^{\alpha}(3)x^{\alpha}y^{\alpha} + kc^{\alpha}(4)(r^2 + 2(x^{\alpha})^2) \\ kc^{\alpha}(3)(r^2 + 2(y^{\alpha})^2) + 2kc^{\alpha}(4)x^{\alpha}y^{\alpha} \end{bmatrix}. \tag{3}$$

After applying the lens distortion, we compute the image coordinates using Equation (4):

$$\begin{cases} x_p^{\alpha} &= fc^{\alpha}(1)(x_d(1) + \ell c^{\alpha} x_d(2)) + cc^{\alpha}(1) \\ y_p^{\alpha} &= fc^{\alpha}(2)x_d(2) + cc^{\alpha}(2). \end{cases} \tag{4}$$

Collaborative calibration for multiple views is an add-on function in the toolbox (Bouguet, 2008). We calibrate multiple-view cameras via two calibration planes synchronously. The recursive least squares (RLS) algorithm determines the 3D locations of the corner points of these planes, which we describe next.

3.2. Least-squares (LS) localization

Fig. 1 shows the geometric relationship between the image and world coordinate systems. Based on the pinhole imaging model, the straight line between the 2D image coordinate of the centroid for the detected bounding box and the helmet on the mobile robot is the specific ray of the view through the optical center of the camera. However, it cannot reach the center of the helmet because of the errors caused by lens distortion. Thus, there is no common intersection point of the multiple rays from multiple views, which makes it necessary to find the centroid of the least-squares (LS) solution with the shortest distance to all rays. Then, that centroid can be regarded as the estimated 3D location of the robot.

Definition 1. The projection matrix of the specific visual sensor for WMSNs with the LS algorithm is denoted by a 3-tuple $(M^{\alpha}, B_M^{\alpha}, A_M^{\alpha})$. Here, M^{α} denotes the projection matrix of the αth visual sensor ($\alpha \in \{1, 2, 3, 4\}$) obtained by the linear model, and m_{ij}^{α} is the value at the ith row and jth column in the projection matrix M^{α}, which is obtained via the LS algorithm as follows:

$$Z_{ci}^{\alpha} \begin{bmatrix} x_i^{\alpha} \\ y_i^{\alpha} \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^{\alpha} & m_{12}^{\alpha} & m_{13}^{\alpha} & m_{14}^{\alpha} \\ m_{21}^{\alpha} & m_{22}^{\alpha} & m_{23}^{\alpha} & m_{24}^{\alpha} \\ m_{31}^{\alpha} & m_{32}^{\alpha} & m_{33}^{\alpha} & 1 \end{bmatrix} \begin{bmatrix} X_{wi} \\ Y_{wi} \\ Z_{wi} \\ 1 \end{bmatrix} \tag{5}$$

Table 1
Notations of the variables used in this paper.

Notation	Description
s	The signature for identifying the mobile robot without image matching
fc^{α}	Focal length
cc^{α}	Principal point
lc^{α}	Skew coefficient
kc^{α}	Distortion
α	Identification of wireless visual sensors in each room
$(x_p^{\alpha}, y_p^{\alpha})$	Image coordinate of the detected centroid from the αth visual sensor
(x^{α}, y^{α})	Normalized image projection of $(x_p^{\alpha}, y_p^{\alpha})$
$(X_c^{\alpha}, Y_c^{\alpha}, Z_c^{\alpha})$	Camera coordinate of the 3D point P
x_d	Normalized point coordinate with lens distortion
dx	Tangential distortion vector
M^{α}	4×3 projection matrix with the intrinsic and extrinsic parameters
$(x_i^{\alpha}, y_i^{\alpha}, 1)$	Homogeneous normalized image coordinate of the ith corner point in the αth visual sensor
$(X_{wi}, Y_{wi}, Z_{wi}, 1)$	Homogeneous world coordinate of the $w$ith corner point
m_{ij}^{α}	Value in the ith row and jth column of M^{α}
Z_{ci}^{α}	Z axis coordinate of the αth camera coordinate system
B_M^{α}	Matrix transpose of the normalized coordinates
A_M^{α}	Linear model of the normalized image and world coordinates
A_w	First parameter matrix of the αth visual sensor in M^{α}
B_w	Second parameter matrix of the αth visual sensor in M^{α}
$\widehat{P}_w(1)$	Estimated 3D world coordinate via the LS algorithm
$\widehat{P}_w(k)$	3D localization result of the kth iteration
k	Iteration number from 2 to $(rn + 1)$
rn	Row number of A_w
$z(k)$	Observation equation
$K(k), P(k), h(k)$	Iterative parameters in the iterative equations
I	Identity matrix

$$M^{\alpha} = \begin{bmatrix} m_{11}^{\alpha} & m_{12}^{\alpha} & m_{13}^{\alpha} & m_{14}^{\alpha} \\ m_{21}^{\alpha} & m_{22}^{\alpha} & m_{23}^{\alpha} & m_{24}^{\alpha} \\ m_{31}^{\alpha} & m_{32}^{\alpha} & m_{33}^{\alpha} & 1 \end{bmatrix}, \tag{6}$$

where $(x_i^{\alpha}, y_i^{\alpha}, 1)$ is the homogeneous normalized image coordinate of the ith corner point in the αth visual sensor; $(X_{wi}, Y_{wi}, Z_{wi}, 1)$ is the homogeneous world coordinate of the $w$ith corresponding point; m_{ij}^{α} is the value in the ith row and jth column of the projection matrix M^{α}; and Z_{ci}^{α} is the Z axis coordinate of the αth camera coordinate system. Because the two calibration planes are monitored by four visual sensors simultaneously, the world coordinates are the same as those in Eq. (1).

The projection matrix M^{α} can be obtained via the LS algorithm as follows. Because the normalized coordinates $(x_i^{\alpha}, y_i^{\alpha})$ and the world coordinates (X_{wi}, Y_{wi}, Z_{wi}) in LS can be obtained from the procedures for the two calibration planes, B_M^{α} is the matrix transpose of the normalized coordinates,

$$M^{\alpha} = ((A_M^{\alpha})^T A_M^{\alpha})^{-1} (A_M^{\alpha})^T B_M^{\alpha}, \tag{7}$$

where B_M^{α} denotes the matrix transpose of the normalized coordinate, and the normalized image coordinate of the ith corner point in the αth visual sensor is as follows:

$$B_M^{\alpha} = \begin{bmatrix} x_i^{\alpha} & y_i^{\alpha} \end{bmatrix}.^T \tag{8}$$

A_M^{α} denotes the linear model of the normalized image and world coordinates, which can be obtained via LS from the two calibration plane procedures, and (X_{wi}, Y_{wi}, Z_{wi}) are the predefined world coordinates of the corresponding point in these two planes and are monitored by the αth visual sensor as follows:

$$A_M^{\alpha} = \begin{bmatrix} A_1^{\alpha} & A_2^{\alpha} \end{bmatrix} \tag{9}$$

$$A_1^{\alpha} = \begin{bmatrix} X_{wi} & Y_{wi} & Z_{wi} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_{wi} & Y_{wi} & Z_{wi} & 1 \end{bmatrix} \tag{10}$$

$$A_2^{\alpha} = \begin{bmatrix} -x_i^{\alpha} X_{wi} & -x_i^{\alpha} Y_{wi} & -x_i^{\alpha} Z_{wi} \\ -y_i^{\alpha} X_{wi} & -y_i^{\alpha} Y_{wi} & -y_i^{\alpha} Z_{wi} \end{bmatrix}. \tag{11}$$

3.3. Recursive least squares (RLS) localization

We consider the normalized coordinates, world coordinates, and projection matrix to be correlated according to Eq. (9) because the projection matrix M^{α} is obtained from Eq. (7). If the normalized coordinates of the detected blobs are located, the world coordinates can be determined accordingly.

Definition 2. 3D localization for WMSNs based on the RLS localization algorithm is denoted by a 7-tuple $(A_w, B_w, \widehat{P}_w(1), \widehat{P}_w(k), K(k), P(k), h(k))$, where

A_w denotes the first parameter matrix of the αth visual sensor in M^{α}:

$$A_w = \begin{bmatrix} m_{11}^{\alpha} - m_{31}^{\alpha} x_i^{\alpha} & m_{12}^{\alpha} - m_{32}^{\alpha} x_i^{\alpha} & m_{13}^{\alpha} - m_{33}^{\alpha} x_i^{\alpha} \\ m_{21}^{\alpha} - m_{31}^{\alpha} y_i^{\alpha} & m_{22}^{\alpha} - m_{32}^{\alpha} y_i^{\alpha} & m_{23}^{\alpha} - m_{33}^{\alpha} y_i^{\alpha} \end{bmatrix}, \tag{12}$$

and B_w denotes the second parameter matrix of the αth visual sensor in M^{α}:

$$B_w = \begin{bmatrix} x_i^{\alpha} - m_{14}^{\alpha} & y_i^{\alpha} - m_{24}^{\alpha} \end{bmatrix}'. \tag{13}$$

Then, $\widehat{P}_w(1)$ denotes the estimated 3D world coordinate via the LS algorithm, which is also the initial value of the RLS algorithm,

$$\widehat{P}_w(1) = (A_w^T A_w)^{-1} A_w^T B_w, \tag{14}$$

where $\widehat{P}_w(k)$ denotes the 3D localization result of the kth iteration, which is computed based on the $(k - 1)$th result, k is an iteration index ranging from 2 to $(rn + 1)$, and rn is the row number of matrix A_w, in which $z(k)$ is the following observation equation:

$$\widehat{P}_w(k) = \widehat{P}_w(k-1) + K(k)[z(k) - h'(k)\widehat{P}_w(k-1)]. \tag{15}$$

Finally, $K(k)$, $P(k)$, and $h(k)$ denote the iterative parameters in the iterative equations shown below. The initial value of $P(1)$ is $10I$, where I is the identity matrix:

$$K(k) = P(k-1)h(k)[h'(k)P(k-1)h(k) + 1]^{-1} \tag{16}$$

$$P(k) = [I - K(k)h'(k)]P(k-1) \tag{17}$$

$$h(k) = A_w(k-1, :). \tag{18}$$

For clarity, we list the notations of the variables used in this paper in Table 1.

To clarify the contribution of our proposed RLS localization algorithm, we consider the localization errors of target detection and time synchronization of the key frames from multi-view wireless visual sensors as follows. In A_w and B_w, we use B_M^{α} as the parameter of the matrices, which are $[x_i^{\alpha} \quad y_i^{\alpha}]$. In computer vision, the state-of-the-art RLS localization methods employ the 2D coordinates of the detected target from the αth cable camera (Bouguet, 2008). We assume that the multi-view cable camera systems are precisely synchronized and that the key frames from multiple views have the same time stamps. However, in our proposed WMSN system, the key frames from multi-view wireless visual sensors cannot have the exact same timestamps because synchronization in WMSNs remains a challenging task.

Therefore, we believe that B_M^{α} has two problems that can affect the localization accuracy of our proposed WMSN system. The first problem is localization error of the image coordinates for the 2D detected

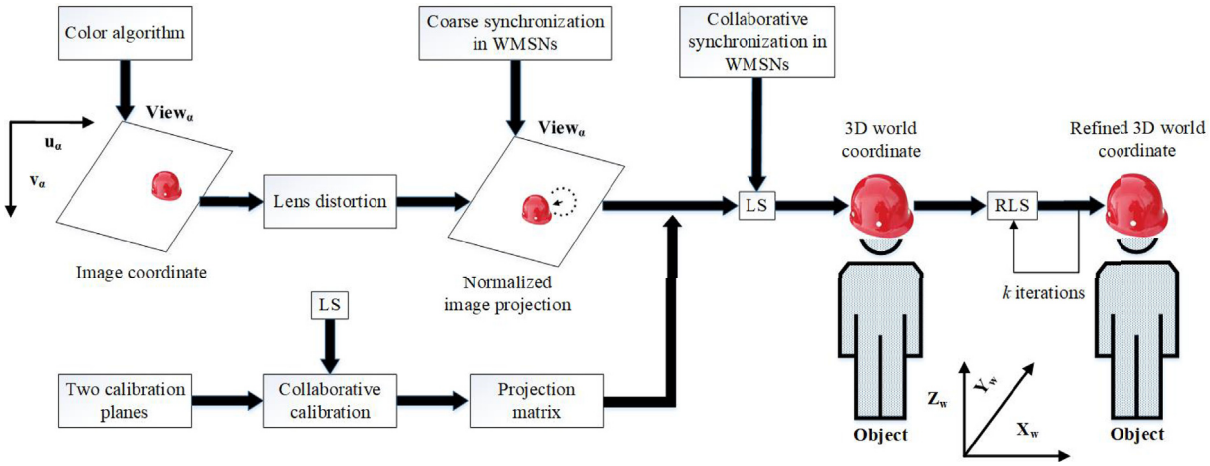


Fig. 2. System architecture of our 3D localization algorithm in WMSNs.

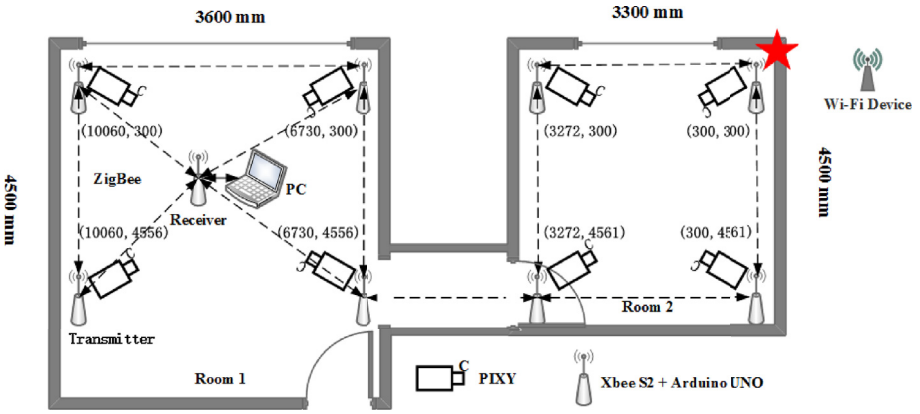


Fig. 3. The mesh network of the proposed WMSNs.

target. For localization error problems, we present the color algorithm to provide accurate 2D target detection: we also compare its localization performance with the well-known KITTI vision benchmark suite (Geiger et al., 2012) in Section 5. Because the color algorithm can precisely localize the color marker on the mobile robot, its localization accuracy is better than those of state-of-the art methods such as BM-NET and SAITv1, as shown in Table 5. Furthermore, our RLS localization algorithm uses the normalized image coordinates defined by Equations (1)–(4) as replacements for the image coordinates in Bouguet (2008) to reduce the effects of lens distortion caused by cameras with 60° angles of view. Our method achieves better localization performance than does the Triangulation Algorithm (TA) (Bouguet, 2008), as shown in Figs. 5–10. The second problem involves the lack of key frame time synchronization from the multi-view wireless visual sensors. To address synchronization problems, we redesigned the collaborative synchronization mechanism in WMSNs (Feng et al., 2017) and

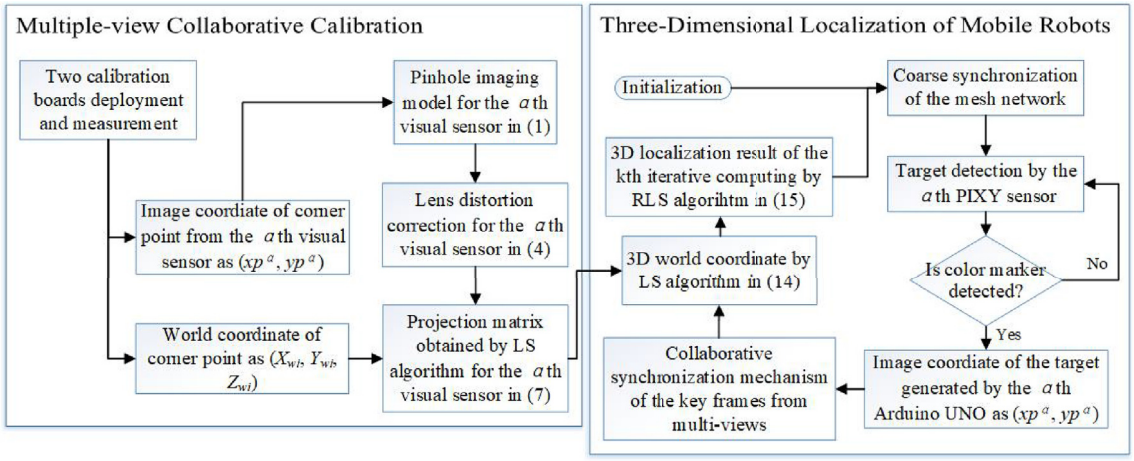
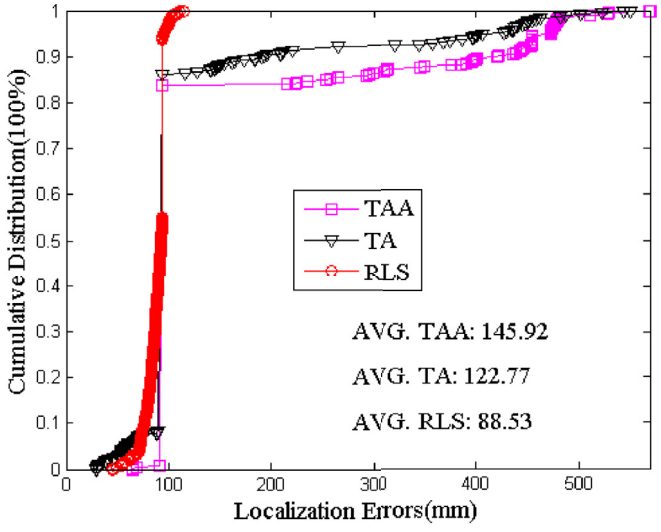
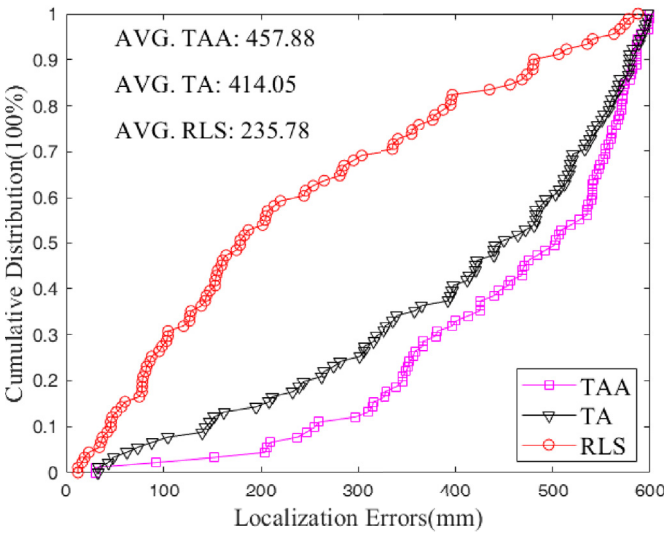


Fig. 4. RLS localization of mobile robots.

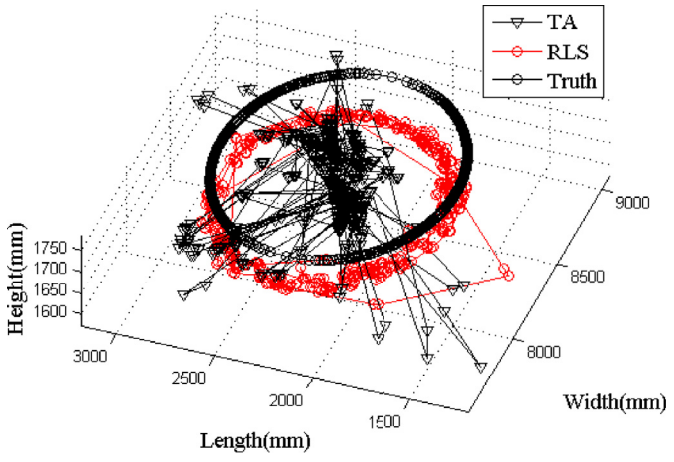


(a)

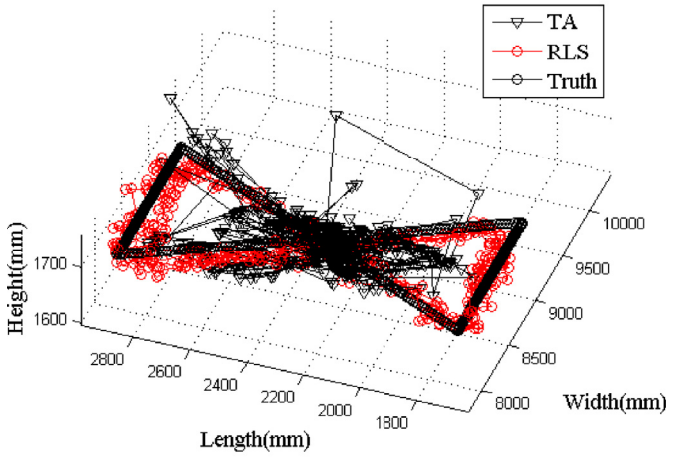


(b)

Fig. 5. The cumulative distributions of two trajectories of the mobile robot in room 1.



(a)



(b)

Fig. 6. The 3D localization results of two trajectories of the mobile robot in room 1.

employed it on the mobile robot to obtain target detection results from multiple-view visual sensors synchronously. Section 5 details this process. The experimental results in Figs. 11–13 show that the mean time difference among groups in the data received from multi-view visual sensors is approximately 10 ms; however, some time differences among key frames exceed 100 ms. Nonetheless, this is acceptable for our RLS algorithm because Equations (12) and (13) can adopt the normalized image coordinates despite their imperfect synchronization and yield a unique solution in Equation (14). Moreover, our proposed RLS algorithm improves the localization accuracy through the kth iteration and reduces the side effects caused by the key frame time differences via Equation (14).

We show the advantages of our proposed RLS algorithm even when time differences among key frames exceed 100 ms by comparing it with TA (Bouguet, 2008); the performance evaluation in shown in Figs. 5–10. Fig. 6(a) and 6(b) and Fig. 9(a) and 9(b) describe the circle and cross trajectories of the mobile robot in rooms 1 and 2. The black triangle markers represent TA's localization results: the markers are distributed in a disordered manner that does not follow the circle and cross trajectories. However, the red circle markers represent the localization results of our proposed RLS algorithm, which follows both trajectories in sequence. The experimental results show RLS functions even when the time differences among key frames exceed 100 ms, and the localization results approximate the ground truth, which is represented by the black circles in the figures. In contrast, TA has very poor localization accuracy, and the localization results deviate significantly from the ground truth when the time differences reach approximately 100 ms.

To discuss our 3D localization algorithm in WMSNs more clearly, Fig. 2 depicts the system architecture of our proposed algorithm. In our established WMSNs, multiple views of the wireless visual sensors in each room monitor the target activity and generate the target detection data. Each wireless visual sensor transmits its data to the sink node connected to the mobile robot via the mesh network. The computing device on the mobile robot is responsible for the data processing and fusion required to localize mobile robots in 3D space. However, the numerous wireless visual sensors in WMSNs generate large volumes of multimedia data, which they transmit to the mobile robot in real-time, causing many challenges. Target detection accuracy is the primary concern because a large number of false alarms can lead to

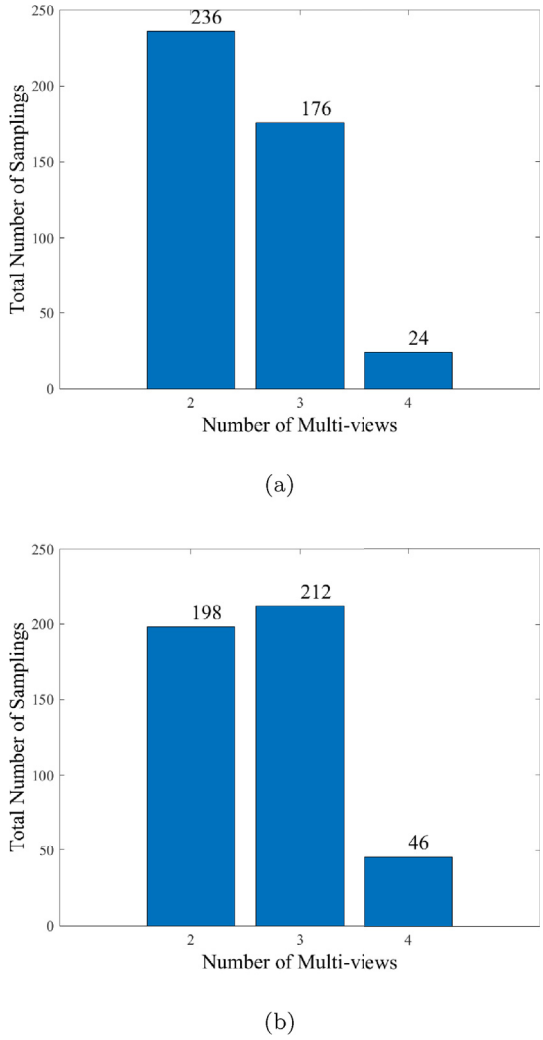


Fig. 7. The wireless communication performance of two trajectories of the mobile robot in room 1.

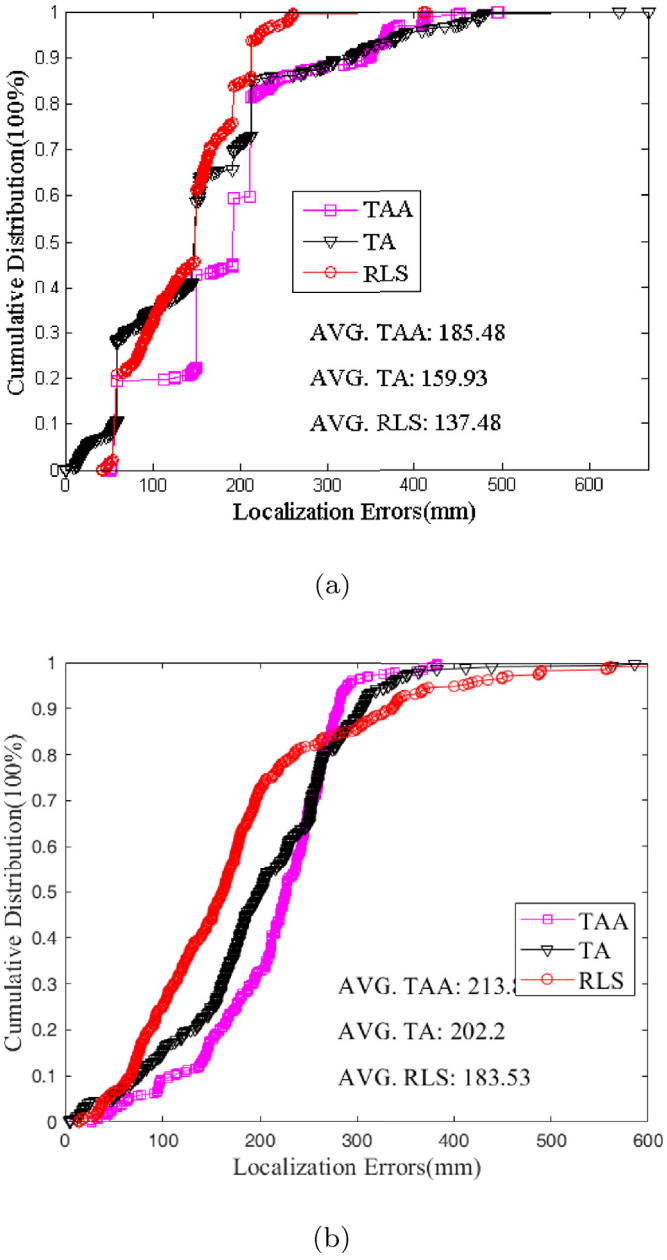


Fig. 8. The cumulative distributions of two trajectories of the mobile robot in room 2.

mobile robot localization failure. Therefore, we employ the color algorithm with the PIXY sensors to identify the colors and bounding boxes of the color markers carried by the mobile robots to generate the image coordinates of the mobile robots in each view of the wireless visual sensors. Second, to overcome the lens distortion of cameras with a 60° angle of view, we refine the image coordinates by fully considering the skew coefficient and lens distortion, which results in new normalized image coordinates with lens distortion compensation. Third, each wireless visual sensor has its own local time, but the system architecture does not recognize the meaningless local time; it requires a unified system time. Thus, the course synchronization mechanism in WMSNs periodically broadcasts the system time, allowing time collation for all sensors. The mechanism can also achieve reliable, efficient synchronization of the key frames and asynchronous packet transmission from multiple views. Fourth, in our 3D localization algorithm, we need to obtain the intrinsic and extrinsic parameters of multiple-view cameras synchronously. Therefore, we propose a collaborative calibration method that uses two calibration boards to calibrate multiple-view cameras in one room collaboratively using the LS algorithm. The robot stores the projection matrix of all the cameras for the localization algorithm. Fifth, on the mobile robot side, the sink node collects the packets from the multiple-view visual sensors and delivers them to the computing device, which implements collaborative synchronization to sample and cluster the received messages in real-time. Furthermore, the computing device performs data fusion of the normalized image coordinates and the projection matrix from multiple views to estimate the robot's 3D world coordinates via the LS algorithm. Finally, this device improves the localization accuracy by performing k iterations of the RLS algorithm.

4. Recursive least squares localization of mobile robots

We consider the WMSN's distributed architecture as a mesh network of XBee sensors with four components: PIXY sensors (C. Labs), Arduino UNO sensors, XBee S2 sensors and a PC for synchronously aggregating the data from multiple visual sensor views. Fig. 3 shows the WMSN's deployment and the mesh network. The PIXY sensors, which monitor target activities and generate detection data—including the targets' colors, bounding boxes and each target's unique number—are placed at different locations in the indoor environment. They transmit these detection data to the Arduino UNO, which conducts data analysis and

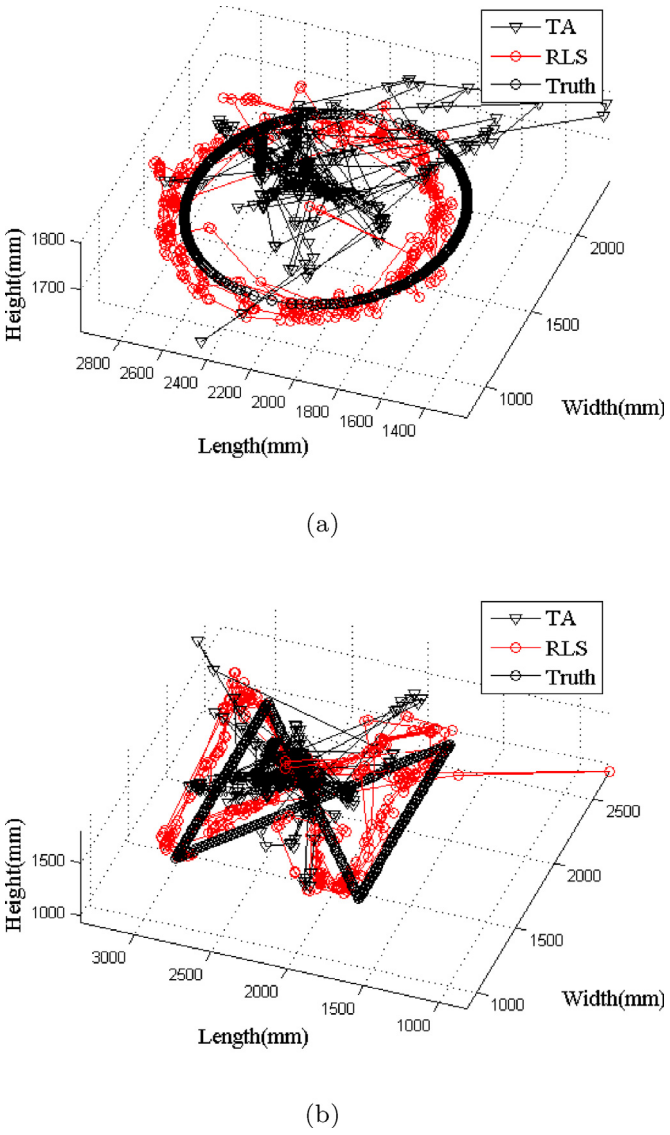


Fig. 9. The 3D localization results of two trajectories of the mobile robot in room 2.

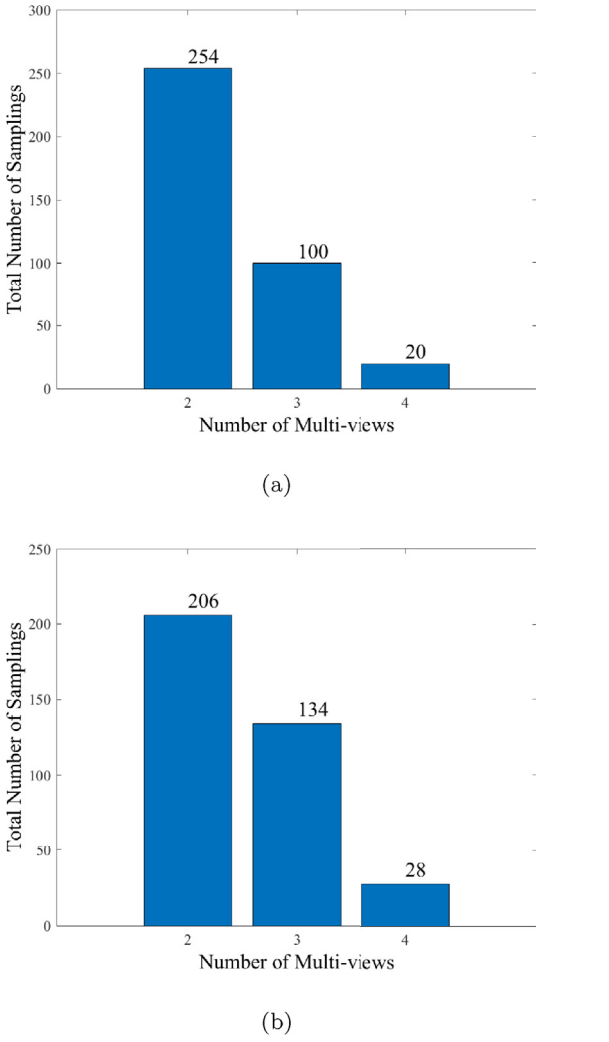
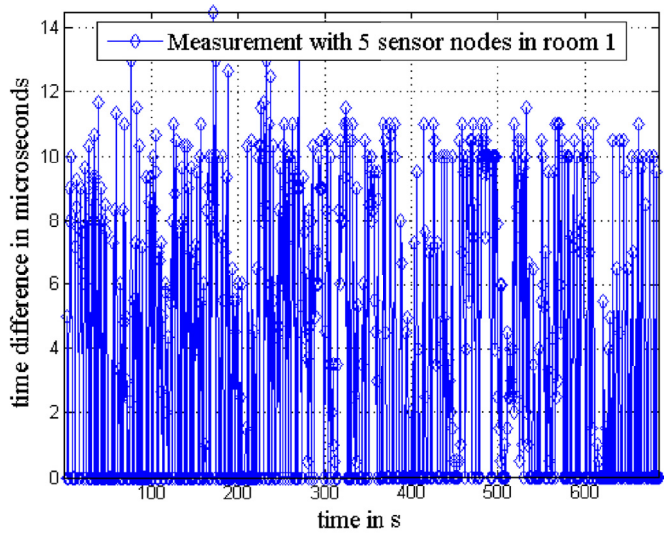


Fig. 10. The wireless communication performance of two trajectories of the mobile robot in room 2.

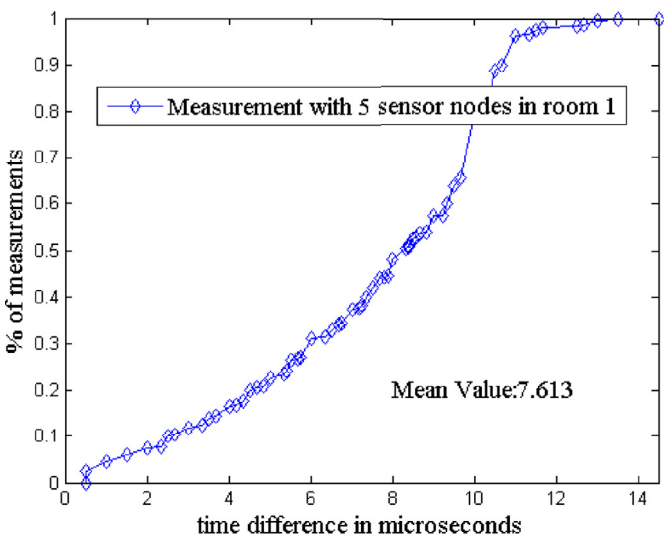
encapsulation. Then, the XBee S2 sensors transmit the packets to the sink node via the mesh network. The sink node generates a trigger message to the connected PC, which is responsible for data processing and fusion to localize mobile robots in 3D space. As shown in Fig. 1, eight visual sensors are deployed at the four corners of two rooms. The red star is the coordinate origin; the ground plane coordinates of the visual sensors are given in millimeters. The four visual sensors cover a room area of approximately a 1.5×1.5 m^2; fewer than four sensors overlap the remaining areas. The mobile robot carries the PC that connects to the coordinator and moves in the space between the rooms. When the mobile robot cannot be detected by any of the visual sensors, we localize it dynamically to provide the navigation service (Wu et al., 2013; Feng et al., 2014).

We design the collaborative synchronization mechanism in WMSNs to obtain target detection results from multiple-view visual sensors synchronously (Feng et al., 2017), and we use it in this work with some modifications. The PIXY sensors identify the colors and the bounding boxes of the color markers carried by mobile robots. The Arduino UNO sensors encapsulate the detection results into packets and transmit them over the XBee sensor network. However, the mobile robot needs to localize itself in 3D space. Therefore, we deploy the sink node directly on the robot. The computing device carried by the mobile robot processes the fused data and determines its own 3D locations. We present a method to perform reliable, efficient synchronization of the key frames and asynchronous packet transmission from multiple views. In the 3D localization algorithm, we need to obtain the intrinsic and extrinsic parameters of the multiple-view cameras synchronously. We use two calibration boards to calibrate the multiple-view cameras in each room collaboratively. The robot stores the intrinsic and extrinsic parameters of all the cameras in both rooms. When the robot is moving, the PIXY sensors detect the target, identify the robot by its color, and determine its bounding box. Next, the Arduino UNO sensors extract the target's image coordinates and coarsely synchronize the collaborative synchronization mechanism (Feng et al., 2017). The Arduino sensors encapsulate the packets and transmit them via the XBee sensor network to the sink node on the mobile robot. The sink collects the packets from the multiple-view visual sensors and delivers them to the computing device via a serial port. The computing device implements the collaborative synchronization method, which samples and clusters the received messages in real-time (Feng et al., 2017). Finally, the computing device fuses the multiple views to estimate the robot's 3D world coordinates via the LS algorithm. This approach improves the localization accuracy by performing k iterations of the RLS algorithm.

Fig. 4 shows the flowchart of the RLS algorithm for mobile robot localization. The first stage involves multiple-view collaborative cali-

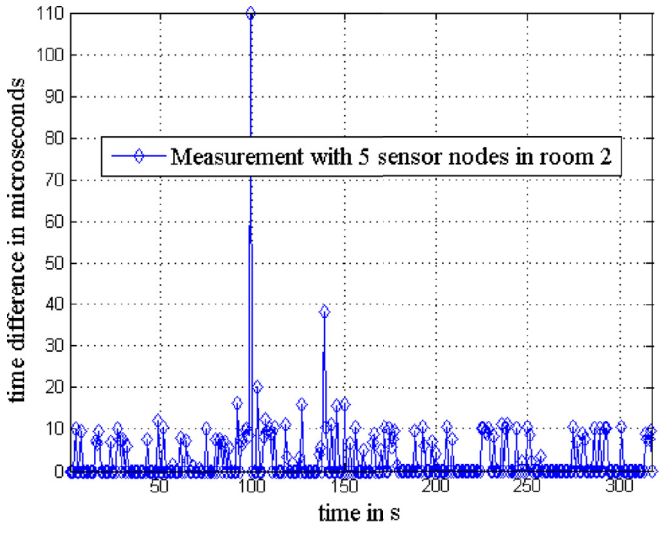


(a)

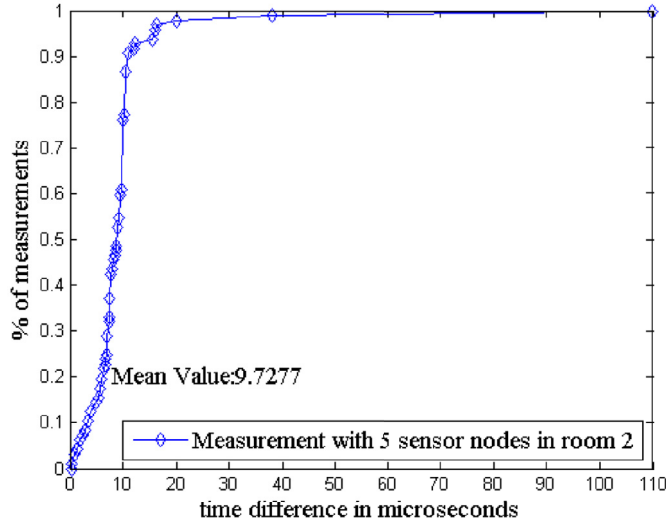


(b)

Fig. 11. Measurements of five sensor nodes in room 1 for collaborative synchronization. (a) t^{μ} for the five nodes in room 1; (b) CDF of t^{μ} in room 1.



(a)



(b)

Fig. 12. Measurements for five sensor nodes in room 2 for collaborative synchronization. (a) t^{μ} for the five nodes in room 2; (b) CDF of t^{μ} in room 2.

bration. First, we manually deploy the two calibration boards at the centers of the rooms and obtain the world coordinates of the corner points. Next, the αth PIXY sensor localizes the corner points in its image and autonomously determines the image coordinates of the corner points. These image coordinates form the input to the pinhole imaging model for the αth visual sensor. We further improve the localization accuracy by performing lens distortion correction. Finally, we use the LS algorithm to obtain the projection matrix, which includes the intrinsic and extrinsic parameters of the αth visual sensor. The projection matrices of all the visual sensors are stored at the mobile robot for further application.

The second stage involves mobile robot 3D localization. The sink node periodically sends unicast synchronization packets to predefined visual sensors. If the visual sensors detect no target, they remain in sleep mode to promote energy efficiency. When the αth PIXY sensor detects the color marker of a robot, it identifies the target's color and bounding box and delivers that information to the αth Arduino UNO, which extracts the target's image coordinates in chronological order. However, the timestamps of the image coordinates from the multiple views do not match exactly. We apply the collaborative synchronization mechanism to match the timestamps of the key frames from multiple views. Then, the αth Arduino UNO encapsulates the data in packets and transmits them over the XBee sensor network to the sink node. On the mobile robot side, the computing device first clusters the received data into groups in real time. Each group includes the target's image coordinates from multiple views with the same timestamp. Then, the computing device fuses the group data to obtain the mobile robot's 3D world coordinates via the LS algorithm. Furthermore, the computing device refines the 3D localization result by performing k iterations of the RLS algorithm. Finally, the entire system enters standby mode in preparation for the next round of processing.

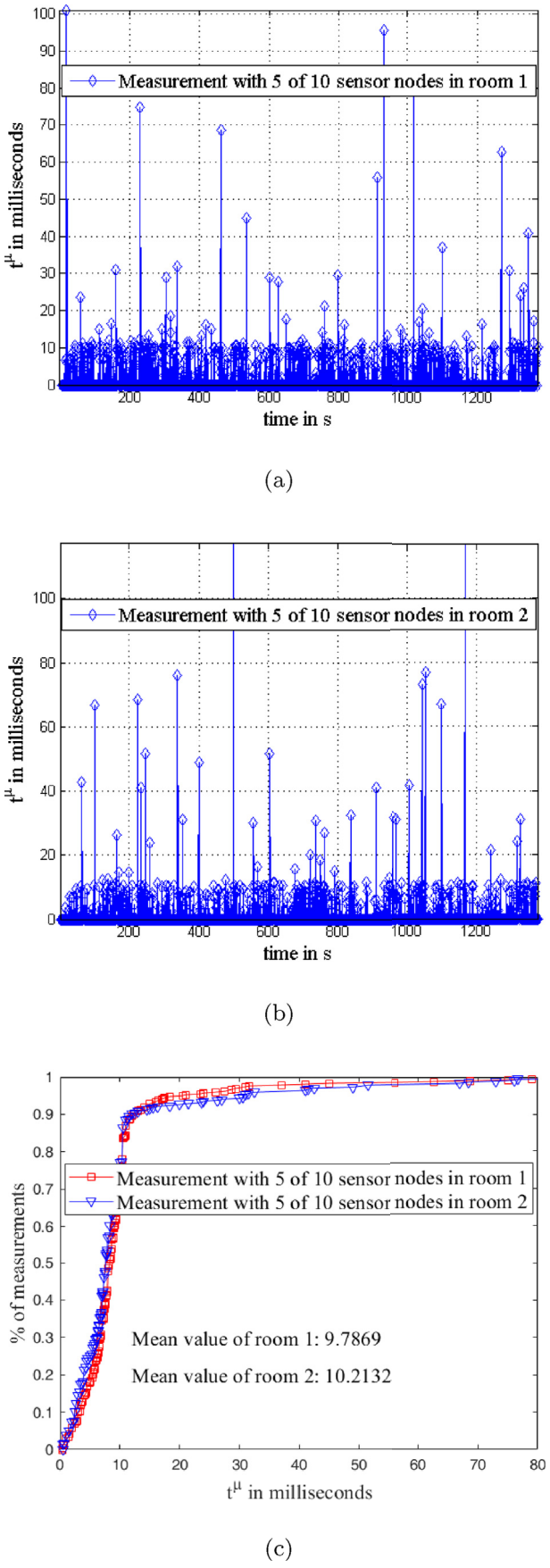


Fig. 13. Measurement of ten sensor nodes in two rooms for collaborative synchronization. (a) t^{μ} for five of ten nodes in room 1. (b) t^{μ} for five of ten nodes in room 2. (c) The CDF of t^{μ} in the two rooms.

5. Experimental results and analysis

We implemented the distributed WMSN architecture on a self-developed platform, as Fig. 3 shows. Each PIXY sensor has 24 clock cycles per pixel and captures 320 × 200 images at 50 fps. To localize the 2D image coordinates, we use an image size of 320 × 200, which we scale to 640 × 400.

In our platform, the coordinator transmits the synchronization message to every visual sensor at 5-s intervals. The visual sensors synchronously sample the key frame at 500-ms intervals. Two key frames are encapsulated into the packet and transmitted to the coordinator in each visual sensor's time slot asynchronously at 1-s intervals. The real-time received data are clustered into discriminative groups using our rules for spatial and temporal correlation.

We use the triangulation algorithm (TA) (Bouguet, 2008) in the performance evaluation because the proposed platform is a multiple-view system. We present the triangulation algorithm that uses all 4 cameras in each room (TAA), which fuses the mean values of each pair of multiple cameras' estimated 3D space locations via triangulation.

Figs. 5–7 depict the mobile robot's circle and cross trajectories in room 1. The cumulative distribution function (CDF) in Fig. 5(a) shows that the proposed RLS algorithm achieves the smallest localization error: its average error is approximately 88 mm, and 90% of the errors are below 100 mm compared to the ground truth. However, 20% of the errors in the TA and TAA algorithms are worse than those of other algorithms, and some errors are over 500 mm. The CDFs in Fig. 5(a) and (b) indicate that the circle trajectory's average localization error is much lower than that of the cross trajectory. Because the robot is moving at the center of the area where the four cameras overlap, the lens distortion at the image edge in Fig. 5(a) is less than that in Fig. 5(b). Most of the errors in both trajectories are within 100 mm and 240 mm. RLS achieves the best localization accuracy for both trajectories. Fig. 6(a) and (b) show that RLS accords with the ground truth of the circle and cross trajectories, whereas TA's estimated locations are scattered about the trajectories. Fig. 7(a) and (b) depict the quality of wireless communications. Very few samples gather 2D image coordinates from all four visual sensors at the same time; the numbers of sampling groups from 2 or 3 views are roughly similar. This indicates good communication quality for dynamic mobile robot localization in WMSNs. Hence, we can ensure that the proposed RLS algorithm provides excellent performance in the circle trajectory with high localization accuracy and overcomes network instability successfully even when data fusion can be performed on data from only two or three multi-view camera images.

Figs. 5(a) and 6(a) clearly present our proposed RLS algorithm's better localization performance. The RLS algorithm performs stably and achieves the optimal solution for multiple rays in 3D space. However, the TA and TAA algorithms do not perform very well in this situation because the deviations between multiple rays are large, which occurs because the time synchronization among the multiple wireless visual sensors is not perfectly matched. The large time differences lead to large deviations between multiple rays and increases the errors of the TA and TAA algorithms. Approximately 20% of the localization errors exceed 200 mm, and the localization results are scattered in a disorderly manner in 3D space.

Figs. 5(b) and 6(b) depict the effect of lens distortion on the localization accuracy. Because the mobile robot is moving along the edge of multiple views in the cross trajectory, the target image coordinates deviate considerably from the real coordinates, and the normalized image coordinates after the lens distortion correction retain large errors. Moreover, the errors are superimposed with the time synchronization deviation, which leads to even worse localization errors. The average error of RLS in the cross trajectory is 235.78 mm, nearly three times as high as the average error in the circle trajectory. This problem occurs with the TA and TAA algorithms at the same time. The localization results of all

Table 2
The easy category results for one robot in two rooms.

Rooms	Trajectory	Category	1st Cam	2nd Cam	3rd Cam	4th Cam
Room1	Circle	Easy	5	213	225	214
		Total	19	214	233	233
		Percentage	26.32%	99.53%	96.57%	91.85%
		Average	**93.99%**			
	Cross	Easy	66	219	287	299
		Total	85	220	304	304
		Percentage	77.65%	99.55%	94.41%	98.36%
		Average	**95.40%**			
	Circle + Cross	Average	**94.79%**			
Room2	Circle	Easy	104	174	308	178
		Total	110	220	308	250
		Percentage	94.55%	94.55%	100%	71.20%
		Average	**86.04%**			
	Cross	Easy	148	176	227	143
		Total	176	258	267	223
		Percentage	84.09%	68.22%	85.02%	64.13%
		Average	**75.11%**			
	Circle + Cross	Average	**80.46%**			
Room1+2	Circle + Cross	Average	**90.02%**			

the algorithms in the cross trajectory deviate further from the ground truth than in the circle trajectory.

Fig. 7(a) and (b) show the results after clustering the received data. The computing device clusters the received data into groups based on the timestamps. Due to wireless communication instability, some groups contain data from all 4 visual sensors and have similar timestamps, but some groups contain data from only 2 or 3 visual sensors. As the figures show, 24 and 46 sampling groups have samples from 4 views, 176 and 212 sampling groups have samples from 3 views, and 236 and 198 sampling groups have samples from 2 views in the two trajectory scenarios, respectively. The number of sampling groups with 2 and 3 views are similar. These data represent the best wireless communication quality in the tested WMSNs.

Figs. 8–10 both depict the same trajectory of the mobile robot in room 2. The CDFs in Fig. 8(a) and (b) show that the average localization error in Fig. 8(a) is smaller than that in Fig. 8(b); however, the average localization error is larger than that in Fig. 5(a) for room 1. The reason is that the localization performance of the eight visual sensors varies in different experimental settings. Table 2 details the localization accuracy for one robot in both rooms. For the circle trajectory, the localization accuracies for one robot in rooms 1 and 2 are 93.99% and 86.04%, which are similar to the results in Figs. 5(a) and 8(a), respectively. The average localization error in room 2 (137.48 mm in Fig. 8(a)) is worse than that in room 1 (88.53 mm in Fig. 5(a)). Because the cross trajectory accuracy is only 75.11%—worse than that of the circle trajectory—the average localization error in Fig. 8(b) exceeds that in Fig. 8(a) (183.53 mm vs. 137.48 mm, respectively). As shown in Fig. 7(a) and (b), we obtain twice as many samples from two cameras than from three cameras, which directly impacts the accuracy of all the localization algorithms. Further, Fig. 3 illustrates that room 2 is smaller than room 1; thus, there are more opportunities for the robot to move at edges of images, where more lens distortions occur, in room 2 than in room 1. However, the average error for all algorithms in Fig. 8(b) is much lower than that in Fig. 5(b) for the cross trajectory; the smaller room size enables better localization accuracy near the corners of this trajectory. Fig. 9(a) and (b) show that localization results are consistent with the ground truth of both trajectories. Moreover, Fig. 9(b) shows better localization results than Fig. 6(b), which means that room size impacts the accuracy and that smaller rooms yield better localization results. From these results, we are confident that our RLS algorithm can achieve good performance even in a worse network environment despite the data fusion limitations of only two multiple-view cameras. RLS overcomes lens distortion quite well and provides stable localization accuracy.

Figs. 8(a) and 9(b) provide strong evidence that our proposed RLS algorithm can perform very well even when environmental noise increases. Compared to room 1, room 2 includes two more serious factors that affect the localization results. First, the wireless communication quality in room 2 is worse than that in room 1. As shown in Fig. 7(a) and (b), the numbers of sampling groups with 2 and 3 views are roughly similar. However, there are twice as many sampling groups with only 2 views as those with 3 views in Fig. 10(a) and (b). This result indicates that fewer packets were successfully transmitted to the sink node. Second, the localization performances of the eight visual sensors vary in different settings. Table 2 shows that the localization accuracies for one robot in rooms 1 and 2 are 93.99% and 86.04% for the circle trajectory. Thus, under the influence of these two factors, our RLS algorithm can still achieve a stable performance and find the optimal solution in multiple rays in 3D space even under a larger deviation between the rays—137.48 mm in this case. However, the TA and TAA algorithms do not perform well in this situation: 20% of the localization errors exceed 300 mm, and the localization results show disordered scattering in 3D space.

Figs. 8(b) and 9(b) depict the effects of lens distortion and worse wireless communication quality on the localization accuracy. Because the mobile robot is moving along the edges of multiple views in the cross trajectory, the image coordinates of the targets deviate considerably from the real coordinates. However, as Fig. 3 shows, room 2 is smaller than room 1, and the smaller room size enables better localization accuracy near the corners of this trajectory. After the lens distortion correction, the normalized image coordinates retain smaller errors, and the errors are superimposed with the time synchronization deviation. Furthermore, the majority of the sampling groups have fewer views than room 1. The average RLS error in the cross trajectory is 183.53 mm, which is larger than the 137.48 mm in Fig. 8(a) but smaller than 235.78 mm in Fig. 5(b). This result occurs with the TA and TAA algorithms at the same time.

To evaluate the performance of our proposed method, we employ the well-known KITTI vision benchmark suite (Geiger et al., 2012) for which several localization algorithms have been tested. To apply KITTI's evaluation indicators in our experiments, we define the evaluation metrics based on the KITTI indicators. For all our metrics, the minimum bounding box height is 25 pixels (because robots carry red helmets, which are tracked objects with low heights). Our metrics are as follows:

Table 3
The moderate category for two robots in two rooms.

Rooms	Trajectory	Category	1st Cam	2nd Cam	3rd Cam	4th Cam
Room1	Square	1st Moderate	246	257	256	152
		1st Total	271	271	272	158
		2nd Moderate	161	263	249	158
		2nd Total	269	271	272	160
		Percentage	75.37%	95.94%	92.83%	97.48%
		Average	**89.61%**			
Room2	Square	1st Moderate	242	509	600	509
		1st Total	244	551	630	671
		2nd Moderate	218	474	359	600
		2nd Total	228	550	633	670
		Percentage	97.46%	89.28%	75.93%	82.70%
		Average	**84.06%**			
Room1+2	Square	Average	**85.82%**			

Table 4
The hard category for two robots in two rooms.

Rooms	Trajectory	Category	1st Cam	2nd Cam	3rd Cam	4th Cam
Room1	Cross	1st Hard	299	68	322	322
		1st Total	342	69	348	350
		2nd Hard	157	69	314	331
		2nd Total	329	69	348	351
		Percentage	67.96%	99.28%	91.38%	93.15%
		Average	**85.31%**			
Room2	Cross	1st Hard	269	528	626	512
		1st Total	277	528	646	654
		2nd Hard	221	492	358	592
		2nd Total	243	516	629	641
		Percentage	94.23%	97.70%	77.18%	85.25%
		Average	**87.03%**			
Room1+2	Cross	Average	**86.44%**			

- **Easy**: Maximum occlusion level: Fully visible. In this category, one robot moves along the circle and cross trajectories in both rooms; four visual sensors per room detect the helmet.
- **Moderate**: Maximum occlusion level: Partly occluded. In this category, two robots move along square trajectories in both rooms because the robots can obscure each other, and the square trajectory's four right-angle turns increase the occlusion probability.
- **Hard**: Maximum occlusion level: Difficult to see. In this category, two robots move along cross trajectories in both rooms because the robots can obscure each other, and the cross trajectories' four sharp corner turns greatly increase the occlusion probability.

Table 2 describes the easy category (one robot, both rooms). The robot follows the circle and cross trajectories, and four visual sensors per room detect and localize the helmet carried by the robot. In room 1, the first camera provides very poor detection results because the lighting change from this orientation is worse than the changes from other cameras. For the circle trajectory, only 5 of the 19 data samples involve heights over 25 pixels (26.32%). The remaining three cameras have over 200 data samples, each of which score over 91%. In particular, the second camera achieves over 99%. Each of the four cameras scored 93.99% in room 1 for this trajectory. For the cross trajectory, the results are slightly better, even though the first camera has the worst results. Each of the four visual sensors scores 95.40%. For the circle and cross trajectories in room 1, all the data samples achieve scores of 94.79%. The results in room 2 are worse than those in room 1, because the former is smaller than the latter (see Fig. 3). In the cross trajectory, the robot moves along the edges of room 2, which increases the object detection difficulty. The cross and circle trajectories average 75.11% and 86.04%, respectively. The fourth camera achieves the worst results (71.20% for the circle trajectory and 64.13% for the cross trajectory) because the lighting change in that area is worse than the lighting in the other orientations. In room 2 all data samples for both trajectories achieve 80.46%, while the average for both trajectories in both rooms is 90.02%. In other words, when a robot moves through both rooms without occlusion, 90.02% of the results have bounding box heights that exceed 25 pixels.

Table 3 illustrates the moderate category (two robots in both rooms). Each robot moves along a square trajectory; as before, four cameras in each room detect and localize helmets that the robots carry. Again, the first camera achieves the worst detection results in room 1, where the lighting change from the camera's orientation is the most adverse. For the square trajectory, only 246 out of 271 data samples from the first robot and 161 out of 269 samples from the second robot have heights over 25 pixels, both of which average 75.37%. Each of the remaining three cameras has over 300 data samples and scores over 92%; in particular, the second and fourth cameras score over 95%. When the two robots move along the square trajectory in room 1, the data samples score 89.61% overall. The results in room 2 are worse than those of room 1 due to room 2's smaller size (see Fig. 3). In the square trajectory, the robot moves along the edges of room 2, which hinders object detection. The third camera has the worst results (75.93%) due to lighting changes. When both robots move along the square trajectory in room 2, each data sample scores 84.06%; however, when both robots move along the square trajectory in two rooms, the average score is 85.82%. Hence, when two robots are moving in two rooms and may obscure each other, 85.82% of the detection results have bounding box heights over 25 pixels.

Table 4 describes the hard category (two robots in two rooms). Here, the two robots move along cross trajectories in two rooms; as before, in each room, four visual sensors detect and localize the colored helmet carried by each robot. Again, the first camera in room 1 achieves the worst detection results due to the unfavorable lighting changes

Table 5
Comparison with state-of-the-art methods.

Method	Moderate	Easy	Hard	Runtime	Environment
DH-ARI	91.48%	90.87%	82.25%	4 s	GPU @ 2.5 Ghz (C/C++)
HRI-SH	90.71%	91.34%	84.28%	3.6 s	GPU @ > 3.5 Ghz (Python + C/C++)
BM-NET	90.50%	90.81%	83.92%	0.5 s	GPU @ 2.5 Ghz (Python + C/C++)
SAITv1	90.36%	90.78%	80.48%	0.18 s	GPU @ 2.5 Ghz (Python, C/C++)
TuSimple	90.33%	90.77%	82.86%	1.6 s	GPU @ 2.5 Ghz (Python + C/C++)
Proposed	**85.82%**	**90.02%**	**86.44%**	**0.5 s**	**No GPU (C/C++)**

Table 6
The easy category: one robot in each room with five cameras.

Rooms	Category	1st Cam	2nd Cam	3rd Cam	4th Cam	5th Cam
Room1	Easy	158	180	602	648	290
	Total	158	180	602	648	290
	Percentage	100%	100%	100%	100%	100%
	Average	**100%**				
Room2	Easy	78	94	112	112	62
	Total	78	94	112	112	62
	Percentage	100%	100%	100%	100%	100%
	Average	**100%**				
Room1+2	Average	**100%**				

based on the camera's orientation. For the cross trajectory, only 299 out of 342 and 157 out of 329 data samples from the first and second robots, respectively, have bounding box heights over 25 pixels, yielding 67.96%. The third and fourth cameras have over 700 data samples, and each scores over 91%. Although the second camera can detect only approximately 140 data samples due to the lighting changes, it still achieves the best average percentage: 99.28%. When two robots are moving along the cross trajectory in room 1, the average score is 85.31%. The detection results in room 2 are better than those of room 1 because the lighting changes based on the orientations of the second and fourth cameras are more favorable than in previous cases. When the robot follows this trajectory, it moves along the edge of room 2, which hampers object detection. The third camera achieves the worst detection results due to lighting changes as two robots move along the cross trajectory in room 2, with an average score of 87.03%. Finally, when the two robots move along the cross trajectory in both rooms, the average is 86.44% because the robots move between the two rooms and obscure each other frequently.

For evaluation purposes, we compared our method with other state-of-the-art localization methods on the well-known KITTI vision benchmark suite (Geiger et al., 2012). Table 5 shows the results of this comparison. In the runtime category, our proposed method requires only 0.5 s to collect a data sample, which is considerably faster than DH-ARI (4 s), HRI-SH (3.6 s), or TuSimple (1.6 s). In the moderate and easy categories, our proposed method performs slightly worse than the other methods because our PIXY sensors have no GPU units, and we adopt an image size of 320 × 200 (scaled to 640 × 400), which is the worst configuration shown in the environment category. However, for the hard category, our proposed method achieves the best detection results. These results indicate that the PIXY sensors can use the color algorithm to perform precise target detection and localization under occlusion.

To discuss the scalability of our proposed method, we deploy an extra visual sensor in each room to assess the way that different numbers of sensors influence the wireless communication. Figs. 11 and 12 show measurements for five sensor nodes in each room for collaborative synchronization. First, we deployed the sensors in room 1. A robot moves within the room carrying a coordinator that can communicate only with the sensors in that room. Fig. 11(a) (Feng et al., 2017) shows the time differences t^{μ} among each group in the data received from these sensors under collaborative synchronization. The results clearly indicate that the t^{μ} values are largely under 12 ms. Even in the worst case, t^{μ} is approximately 14 ms, which is excellent frame synchronization for a 3D localization algorithm. Fig. 11(b) shows the CDF of t^{μ} in room 1. Over 95% of the t^{μ} values below 12 ms, and the mean value is 7.613 ms. However, when a robot carrying a coordinator moves inside room 2 and the coordinator can communicate only with the five sensors there, the maximum value of t^{μ} can reach 110 ms. Some t^{μ} values exceed 20 ms, as shown in Fig. 12(a). The reason is that the quality of the wireless communication from the sensors to the coordinator in room 2 is worse than that in room 1, and the quality drop is caused by a bottleneck at the coordinator. Table 6 describes the wireless packets received from five visual sensors by the coordinator in the two rooms. In room 1, the third and fourth visual sensors have the best wireless communication with the coordinator and send most packets to it. The remaining three sensors have similar wireless communication with the coordinator and send similar numbers of packets to it. However, in room 2, while the third and fourth visual sensors send the same numbers of wireless packets to the coordinator; the first and second sensors send fewer packets to it. The fifth visual sensor has the worst wireless communication with the coordinator; the sensor sends only 62 packets. This means that no visual sensor has stable, continuous wireless communication with the coordinator, which results in larger mean t^{μ} values for frame synchronization. Furthermore, Fig. 8 in Feng et al. (2017) shows a time-channel diagram of a multi-channel packet analyzer and clearly indicates that the analyzer receives 10,239 packets in total. However, our proposed architecture uses 9483 packets in channel 9; thus, 756 packets do not belong to this channel; they constitute wireless signal interference. Based on Fig. 7 in Feng et al. (2017), which lists the packets in the multi-channel packet analyzer, it follows that the analyzer can capture only wireless signals at 2435 MHz. Therefore, we conclude that any other wireless signals operating in the same frequency range (such as Wi-Fi signals from neighboring rooms) interfere with the quality of the wireless communication of our proposed WMSNs. Fig. 3 illustrates, the WMSN deployment. Room 1 has a size of 3.6 m × 4.5 m × 2.5 m, while room 2 is 3.3 m × 4.5 m × 2.5 m. Wi-Fi devices are located in neighboring rooms and can interfere with wireless communications in our WMSN. As Fig. 12(b) shows, although 90% of the t^{μ} values are under 10 ms, 5% exceed 20 ms; thus, the mean value of t^{μ} can reach 9.7277 ms. This means that some of the frame

Table 7
The easy category for one robot in two rooms with ten cameras.

Rooms	Category	1st Cam	2nd Cam	3rd Cam	4th Cam	5th Cam
Room1	Easy	534	580	576	550	546
	Total	534	580	576	550	546
	Percentage	100%	100%	100%	100%	100%
	Average	**100%**				
Room2	Easy	418	522	622	574	174
	Total	418	522	622	574	174
	Percentage	100%	100%	100%	100%	100%
	Average	**100%**				
Room1+2	Average	**100%**				

synchronization operations are insufficient for the 3D localization algorithms.

Table 6 describes the localization performances for the five visual sensors in each room. Because we designed this experiment to analyze collaborative synchronization, the sensors do not perform collaborative calibration, and the robot is motionless at the center of the room. Hence, they can detect the color marker on the robot, determine the 2D image coordinates of the detected targets, and transmit data over the wireless network to the coordinator. This experiment fits in the easy category, with one robot in each room and no occlusion. First, in both rooms, because the robot remains motionless in the region of interest, no dynamic lighting changes or image edge effects occur. All the data samples from all the cameras have bounding box heights over 25 pixels, which indicates 100%. Second, in both rooms, the third and fourth cameras transmit 602 out of 648 samples in room 1 and 112 out of 112 samples in room 2. Accordingly, these two visual sensors have better wireless communications than do the others. Third, the fifth camera in each room have different performances. The fifth camera in room 1 transmits 290 data samples, which indicates better wireless communication than the first and second sensors. The fifth camera in room 2 transmits only 62 samples, which is the worst wireless communication performance among all the sensors. Hence, this camera constitutes a source of interference for the original system.

To discuss the maximum number of visual sensors our proposed method can accommodate, we design an experiment in which two robots move inside two rooms separately. The robot in room 1 carries a coordinator that can communicate with the ten visual sensors in both rooms. Fig. 13(a) depicts the time differences (t^{μ}) among each group in the data received from the five visual sensors in room 1 during collaborative synchronization. The figure indicates that most t^{μ} values are under 12 ms, and some exceed 20 ms. In the worst case, t^{μ} can reach 100 ms, and frame synchronization is slightly worse for the 3D localization algorithms. Fig. 13(b) shows the measurements for the five visual sensors in room 2; the figure indicates that most t^{μ} values are under 15 ms. However, the number of t^{μ} values exceeding 20 ms in room 2 is much greater than that in room 1. The maximum value of t^{μ} reaches approximately 150 ms, which is very poor for frame synchronization. Fig. 13(c) shows the CDF of t^{μ} in the two rooms. Over 90% of t^{μ} values are under 15 ms. The mean value of t^{μ} in room 1 is 9.7869 ms, which is slightly worse than that of the five sensors in room 2, as shown in Fig. 12(b). The mean value of t^{μ} in room 2 is 10.2132 ms, which is slightly worse than that in room 1. However, over 8% of the t^{μ} values exceed 20 ms; some exceed 50 ms, and in the worst-case, the t^{μ} value exceeds 100 ms. This result occurs because a larger number of visual sensors increases the wireless interference at the coordinator and causes the bottleneck problem. Therefore, considering the coordinator's capabilities, we collect data from each visual sensor only every 5 s, even though the PIXY sensors have cameras with high frame rates (>100 Hz). Furthermore, we recommend deploying four visual sensors in each room to prevent the coordinator from becoming a bottleneck and achieve rapid frame synchronization for 3D localization algorithms.

Table 7 describes the localization performance measured for ten cameras in two rooms. Because we designed the experiment for collaborative synchronization analysis, the cameras did not perform collaborative calibration, and each robot remained motionless at the center of each room. The robot in room 1 carries the coordinator and communicates with the ten cameras synchronously in the two rooms. Therefore, the visual sensors can detect the color marker on the robot in each room, determine the 2D image coordinates of the detected targets, and transmit the data over the wireless network to the coordinator synchronously. This fits in the easy category: one robot in two rooms with no occlusion. First, in both rooms, because each robot remains motionless in the region of interest, no dynamic lighting changes or image edge effects are present. All the data samples from all the cameras have bounding box heights over 25 pixels, indicating scores of 100%. Second, in one room, all five cameras transmit nearly the same number of data samples (approximately 570). However, in room 2, the cameras achieve very different sample numbers. The third camera transmits the largest number of samples (622), but the fifth camera transmits the smallest number (174). This means that room 2 has worse wireless communications than does room 1 because the five cameras in room 2 must travel longer distances to the coordinator, which is in room 1. Third, the fifth camera in each room has a different performance. The fifth camera in room 1 transmits 546 data samples, which is similar to the other cameras there. However, the fifth visual sensor in room 2 achieves only 174 data samples, which is the worst wireless communication among all the sensors in both rooms. Again, this camera acts as a source of interference for the original system.

Motion capture systems have achieved solid performances in various applications, including digital animation, biomechanical analysis, and clinical diagnosis. Currently, the most common techniques use reflective markers on human skin. To avoid motion impediments from the markers, Stefano et al. (Corazza et al., 2006) developed a model-based motion capture system that does not require such markers. The goals of the system are to circumvent the absence of markers and achieve usability due to numerous synchronized calibrated cameras. The authors successfully evaluated their system's tracking capability in a virtual environment for a complete gait cycle. Despite the precision and popularity of motion capture systems, our proposed method has several advantages over the current systems: (i) Our method can use common color markers to localize targets precisely, achieving an average 3D localization error of 88.53 mm in a 3.6 m × 4.5 m × 2.5 m room, as shown in Figs. 1 and 5(a), whereas motion capture systems require reflective skin markers that can migrate to underlying bones, leaving visible skin artifacts; (ii) Our method uses wireless visual sensors to detect targets and transmit data over wireless networks, but motion capture systems require cabled cameras whose placement is laborious and has a greater infrastructure demand; (iii) Our proposed RLS localization algorithm is robust to asynchronous time stamps for key frames from multi-view cameras. Fig. 13(a) shows that although such time stamps from five visual sen-

Table 8
A comparison with the state-of-the-art architecture.

System	OESP	Proposed architecture
Image processing block	Xilinx Spartan-3	PIXY
Networking processor	XBee	XBee S2
Number of clock cycles	2,457,600	1,536,000
Image size (pixels)	640 × 480	320 × 200
Time (ms)	49.15	20
Transmission data type	Imagery	2D image coordinates
Transmission data size	At least 160 × 100 Byte	At least 72 Byte
Multiple views	Image stream from 1 view	Key frames from multi-views
Time synchronization	No	Yes
Target image	Gray scale image	Color image
Object extraction	Background subtraction	Color algorithm
Localization results	2D localization	3D localization
Status of coordinator	Motionless	Mobile sensor

sors may vary by up to 100 ms (as in room 1), Equation (12) in our RLS algorithm can fuse normalized coordinates of detected blobs from multi-view cameras even when large timestamp differences exist. However, motion capture systems typically require over 18 synchronized cable cameras to localize targets, which increases the infrastructural and equipment requirements; and (iv) our proposed method divides computing tasks into multiple subtasks assigned to various devices, such as PIXY sensors (which perform object detection and extraction subtasks), Arduino UNO sensors (which perform time synchronization and packet encapsulation subtasks), and XBee S2s (which perform message analysis and wireless WSN transmission subtasks (Feng et al., 2017)). In contrast, the current motion capture systems must process computationally intensive tasks on a central server.

To illustrate the advantages of our proposed distributed architecture, we compare its performance with another state-of-the-art WMSN architecture (Pham and Aziz, 2013) in Table 8. Duc et al. proposed an Object Extraction Scheme and Protocol (OESP) for WMSNs using Xilinx Spartan-3 FPGA (Pham and Aziz, 2013); we employ PIXY sensors for image processing. Both approaches use commercial off-the-shelf equipment. Our networking processor is the Digi XBee S2, which is newer than the XBee processor used in OESP. The PIXY and Xilinx FPGAs provide 1,536,000 and 2,457,600 clock cycles, respectively. PIXY requires 6 cycles per pixel for 1280 × 800 images at 25 fps, 12 cycles per pixel for 640 × 200 at 50 fps, and 24 cycles per pixel for 320 × 200 images at 50 fps. For the purpose of real-time 3D localization, we adopt a 320 × 200 image (scaled to 640 × 400 resolution). Because the PIXY sensor requires only 20 ms for target detection, the proposed system is twice as fast as OESP, which used a 640 × 480 image size. Because Duc et al. focus on image transmission to monitor the human-specified region of interest, OESP can transmit imagery over wireless networks; the transmitted data packet size is at least 160 × 100 bytes. However, because our proposed architecture focuses on transmitting 2D image coordinates of detected targets for the robot to monitor the region of interest, our transmitted packet size is 72 bytes, considerably less than that of OESP. Hence, our proposed architecture can achieve time synchronization of key frames from multiple views and estimate the 3D location of the detected targets. In contrast, OESP can transmit imagery from only one view at a time to the coordinator to localize the 2D image coordinates of the detected targets. To reduce the computational burden of background subtraction, Duc et al. reduced the image quality in OESP to gray-scale images. However, our system employs the color algorithm to process high-quality color images, which have good localization performance, as Table 4 describes. Furthermore, because Duc et al. design the architecture for humans to monitor regions of interest, they did not require coordinator mobility. In contrast, we design the proposed architecture so that the mobile robot can monitor the region of interest. Therefore, the robot carries the coordinator that communicates with the wireless visual sensors to determine the 2D image coordinates of the detected targets.

6. Conclusions

By combining WMSNs and distributed computing, we have proposed a distributed architecture that performs target detection and 3D localization of mobile robots in WMSNs via wireless communications. Based on computer vision, we constructed an RLS localization algorithm for mobile robots that appropriately reflects the network uncertainty that occurs during wireless communication among smart sensors and the sink node. We also developed a multiple-view collaborative calibration method to determine the intrinsic and extrinsic parameters of multiple cameras in WMSNs. Our approach improves the localization accuracy of the RLS algorithm. In addition, we implemented our distributed architecture and algorithm in a real-world experimental setting. The results demonstrated that our approach can achieve real-time 3D localization for mobile robots. The RLS algorithm fuses data from multiple visual sensor views, effectively correcting positioning errors, and localizing robots in 3D space. In future work, we plan to extend our approach to allow it to monitor multiple color helmets, rescuers' clothes, and mobile robots.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgments

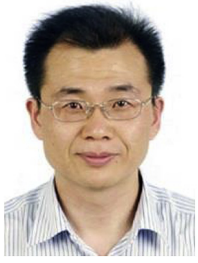
This research was funded by the National Natural Science Foundation of China under grant number 61772018, the Public Welfare Technology Research Project of Zhejiang Province under grant number LGG19F020007, the General Research Projects of Zhejiang Provincial Department of Education under grant number Y201839944, the Public Welfare Technology Application Research Project of Shaoxing City under grant number 2018C10013, and the Research Foundation for Talented Scholars of Shaoxing University under grant number 20185001.

References

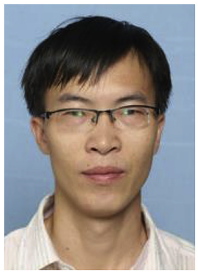
Ahmed, A.A., 2017. A real-time routing protocol with adaptive traffic shaping for multimedia streaming over next-generation of wireless multimedia sensor networks. Pervasive Mob. Comput. 40, 495–511.
Al-Turjman, F., Radwan, A., 2017. Data delivery in wireless multimedia sensor networks: challenging and defying in the iot era. IEEE Wirel. Commun. 24 (5), 126–131.
Bhatt, R., Datta, R., 2016. A two-tier strategy for priority based critical event surveillance with wireless multimedia sensors. Wirel. Netw. 22 (1), 267–284.
Bouguet, J., 2008. Calibration Toolbox. http://www.vision.caltech.edu/bouguetj/calib_doc/index.htm.
C. Labs, C. M. University, cmucam camera, http://www.cmucam.org.
Caruso, D., Engel, J., Cremers, D., 2015. Large-scale direct slam for omnidirectional cameras. In: IROS, vol. 1, p. 2.
Champion, A.C., Yang, Z., Zhang, B., Dai, J., Xuan, D., Li, D., 2013. E-smalltalker: a distributed mobile system for social networking in physical proximity. IEEE Trans. Parallel Distrib. Syst. 24 (8), 1535–1545.

Chen, C., Yang, B., Song, S., Tian, M., Li, J., Dai, W., Fang, L., 2018. Calibrate multiple consumer rgb-d cameras for low-cost and efficient 3d indoor mapping. Remote Sens. 10 (2), 328.
Civera, J., Bueno, D.R., Davison, A.J., Montiel, J., 2009. Camera self-calibration for sequential bayesian structure from motion. In: Robotics and Automation, 2009. ICRA09. IEEE International Conference on. IEEE, pp. 403–408.
Civera, J., Grasa, O.G., Davison, A.J., Montiel, J., 2010. 1-point ransac for extended kalman filtering: application to real-time structure from motion and visual odometry. J. Field Robot. 27 (5), 609–631.
Concha, A., Hussain, W., Montano, L., Civera, J., 2015. Incorporating scene priors to dense monocular mapping. Aut. Robots 39 (3), 279–292.
Corazza, S., Muendermann, L., Chaudhari, A., Demattio, T., Cobelli, C., Andriacchi, T.P., 2006. A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach. Ann. Biomed. Eng. 34 (6), 1019–1029.
Cui, L., Hu, H., Yu, S., Yan, Q., Ming, Z., Wen, Z., Lu, N., 2018. Ddse: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. J. Netw. Comput. Appl. 103, 119–130.
Djelouah, A., Franco, J.-S., Boyer, E., Le Clerc, F., Prez, P., 2015. Sparse multi-view consistency for object segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 37 (9), 1890–1903.
Fcil, J.M., Concha, A., Montesano, L., Civera, J., 2017. Single-view and multi-view depth fusion. IEEE Robot. Automat. Lett. 2 (4), 1994–2001.
Feng, S., Wu, C.-d., Zhang, Y.-z., Jia, Z.-x., 2014. Grid-based improved maximum likelihood estimation for dynamic localization of mobile robots. Int. J. Distributed Sens. Netw. 10 (3), 271547.
Feng, S., Wu, C., Zhang, Y., 2017. Collaborative synchronization mechanism in wireless multimedia sensor networks. Wirel. Pers. Commun. 96 (2), 1929–1943.
Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, pp. 3354–3361.
Hahne, C., Aggoun, A., Velisavljevic, V., Fiebig, S., Pesch, M., 2018. Baseline and triangulation geometry in a standard plenoptic camera. Int. J. Comput. Vis. 126 (1), 21–35.
Ham, Y., Yoon, H., 2018. Motion and visual data-driven distant object localization for field reporting. J. Comput. Civ. Eng. 32 (4).
Sovannarith, H., Chakchai, S.I., Gia, N.T., 2017. Distributed image compression architecture over wireless multimedia sensor networks. Wirel. Commun. Mob. Comput. 2017, 1–21, https://doi.org/10.1155/2017/5471721.
Kang, L., Wu, L., Yang, Y.-H., 2014. Robust multi-view l2 triangulation via optimal inlier selection and 3d structure refinement. Pattern Recognit. 47 (9), 2974–2992.
Kim, H., Oh, H., Bellavista, P., Ben-Othman, J., 2017. Constructing event-driven partial barriers with resilience in wireless mobile sensor networks. J. Netw. Comput. Appl. 82, 77–92.
Kim, P., Chen, J., Cho, Y.K., 2018. Slam-driven robotic mapping and registration of 3d point clouds. Autom. ConStruct. 89, 38–48.
Maisano, D., Mastrogiacomo, L., 2016. A new methodology to design multi-sensor networks for distributed large-volume metrology systems based on triangulation. Precis. Eng. 43, 105–118.
Mavrinac, A., Chen, X., Alarcon-Herrera, J.L., 2015. Semiautomatic model-based view planning for active triangulation 3-d inspection systems. IEEE ASME Trans. Mechatron. 20 (2), 799–811.
Mur-Artal, R., Montiel, J.M.M., Tardos, J.D., 2015. Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Robot. 31 (5), 1147–1163.
Nandhini, S.A., Radha, S., Kishore, R., 2018. Efficient compressed sensing based object detection system for video surveillance application in wmsn. Multimed. Tools Appl. 77 (2), 1905–1925.
Osa, V., Matamales, J., Monserrat, J.F., Lpez, J., 2013. Localization in wireless networks: the potential of triangulation techniques. Wirel. Pers. Commun. 68 (4), 1525–1538.
Otero, J., Snchez, L., 2015. Local iterative dlt soft-computing vs. interval-valued stereo calibration and triangulation with uncertainty bounding in 3d reconstruction. Neurocomputing 167, 44–51.
Pahwa, R.S., Lu, J., Jiang, N., Ng, T.T., Do, M.N., 2018. Locating 3d object proposals: a depth-based online approach. IEEE Trans. Circuits Syst. Video Technol. 28 (3), 626–639.
Pham, D.M., Aziz, S.M., 2013. Object extraction scheme and protocol for energy efficient image communication over wireless sensor networks. Comput. Network. 57 (15), 2949–2960.
Piasco, N., Sidib, D., Demonceaux, C., Gouet-Brunet, V., 2018. A survey on visual-based localization: on the benefit of heterogeneous data. Pattern Recognit. 74, 90–109.
Rehman, Y.A.U., Tariq, M., Sato, T., 2016. A novel energy efficient object detection and image transmission approach for wireless multimedia sensor networks. IEEE Sens. J. 16 (15), 5942–5949.
Rowe, A.G., Goode, A., Goel, D., Nourbakhsh, I., 2007. Cmucam3: an Open Programmable Embedded Vision Sensor. Report. Carnegie Mellon Robotics Institute.
Rubino, C., Crocco, M., Del Bue, A., 2018. 3d object localisation from multi-view image detections. IEEE Trans. Pattern Anal. Mach. Intell. 40 (6), 1281–1294.
Rupnik, E., Nex, F., Toschi, I., Remondino, F., 2015. Aerial multi-camera systems: accuracy and block triangulation issues. ISPRS J. Photogrammetry Remote Sens. 101, 233–246.
Saputra, M.R.U., Markham, A., Trigoni, N., 2018. Visual slam and structure from motion in dynamic environments: a survey. ACM Comput. Surv. (CSUR) 51 (2), 37.
Strasdat, H., Montiel, J.M., Davison, A.J., 2012. Visual slam: why filter? Image Vis Comput. 30 (2), 65–77.
Tanathong, S., Lee, I., 2014. Using gps/ins data to enhance image matching for real-time aerial triangulation. Comput. Geosci. 72, 244–254.
Villamizar, M., Andrade-Cetto, J., Sanfeliu, A., Moreno-Noguer, F., 2018. Boosted random ferns for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 40 (2), 272–288.
Wang, K., Ji, Q., 2018. 3d gaze estimation without explicit personal calibration. Pattern Recognit. 79, 216–227.
Wu, C., Feng, S., Zhang, Y., 2013. Dynamic localization of mobile robot based on asynchronous kalman filter. J. Northeast. Univ. (Nat. Sci.) 34 (3), 312–316.
You, K., Yang, W., Han, R., 2015. The video collaborative localization of a miner's lamp based on wireless multimedia sensor networks for underground coal mines. Sensors 15 (10), 25103–25122.

Sheng Feng received the B.S. in Computing from Greenwich University in 2004, the M.S. degree in software engineering in 2013 and the Ph.D. degree in pattern recognition and intelligent system from Northeastern University, Shenyang, China in 2017. He is currently a Lecturer with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China. His research interests are in the area of intelligent robot, computer vision and wireless sensor networks.

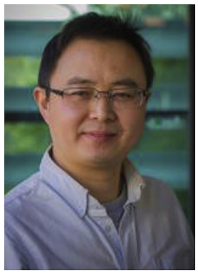


Shigen Shen received the B.S. degree in fundamental mathematics from Zhejiang Normal University, Jinhua, China, in 1995, the M.S. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent systems from Donghua University, Shanghai, China, in 2013. He is currently a Professor with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China. His current research interests include Internet of Things, cyber security, cloud computing and game theory.



Longjun Huang received the B.S. degree in Computer Software from Lanzhou University, Lanzhou, China, in 1999, the M.S. degree in Control Theory and Engineering from Zhejiang University of Technology, Hangzhou, China, in January 2010, and the Ph.D. degree in Control Science and Engineering in Zhejiang University of Technology, Hangzhou, China in January 2017. He is currently a lecturer with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China. His current research interests include electromagnetic nanonetworks, wireless sensor networks and game theory.

Adam C. Champion received the B.S. degree and Ph. D study in Computer Software and Engineering from the Ohio State University, Columbus, Ohio, USA, since 2007. He is currently a senior lecturer with the Department of Computer Science and Engineering, the Ohio State University, Columbus, Ohio, USA. His current research interests include mobile systems, computing networking, system and network security, distributed computing.



Shui Yu (SM'12) received the B.Eng. degree in electronic engineering, the Associate degree in mathematics, and the M.Eng. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1993, 1993, and 1999, respectively, and the Ph.D. degree in computer science from Deakin University, Melbourne, VIC, Australia, in 2004. H He is currently a full Professor with the School of Software, University of Technology Sydney (UTS), NSW, Australia. Before joining UTS, he was a Senior Lecturer with the School of Information Technology, Deakin University, Australia, and a Lecturer with the Computer College in University of Electronic Science and Technology of China. He has authored 2 monographs and edited 2 books, over 250 techni-

cal papers, including top journals and top conferences, such as the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON KNOWLEDGE & DATA ENGINEERING, and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, and for IEEE INFOCOM. His current research interests include security and privacy in networking, big data, cyber-space, and mathematical modeling. He initiated the research field of networking for big data in 2013. He has an H-Index of 25. Dr. Yu actively serves his research communities in various roles. He is currently serving the Editorial Boards of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE ACCESS, the IEEE INTERNET OF THINGS JOURNAL, IEEE Communications Magazine, and a number of other international journals. He has served over 70 international conferences as a member of organizing committee, such as a Publication Chair of IEEE Globecom in 2015 and 2017, IEEE INFOCOM in 2016 and 2017, the TPC Co-Chair of IEEE BigDataService in 2015, IEEE ATNAC in 2014, IEEE ITNAC in 2015, and an Executive General Chair of ACSW2017. He was a member of Deakin University's Academic Board from 2015 to 2016, and is a member of the AAAS and the ACM, and a Vice Chair of the Technical Committee on Big Data Processing, Analytics, and Networking of the IEEE Communication Society.

ChengDong Wu was born in June 1960. He received the B.Sc. degree in electrical automation from the University of Shenyang Architectural and Civil Engineering, Shenyang, China, in 1983, the M.Sc. degree in control theory and its application from Tsinghua University, Beijing, China, in 1988, and the Ph.D. degree in industrial automation from Northeastern University, Shenyang, in 1994. He is currently the Director of the Institute of Artificial Intelligence and Robotics, Northeastern University. He was the Head of more than 20 foundation projects in the fields such as robot control and path planning, image processing, intelligent traffic system, and wireless network communication. He has authored or coauthored nine books and more than 150 of journal and conference papers. His current research interests include the pattern recognition and image processing, robot intelligent control, and wireless sensor network.

YunZhou Zhang received B.S. degree and M.S. degree in Mechanical and Electronic engineering from National University of Defense Technology, Changsha, China in 1997 and 2000, respectively. He received Ph.D. degree in pattern recognition and intelligent system from Northeastern University, Shenyang, China, in 2009. He is now a professor with Northeastern University, China. His research interests include image processing, intelligent robot and wireless sensor networks. He is a member of the IEEE society.