

Noninteractive Localization of Wireless Camera Sensors with Mobile Beacon

Yuan He, *Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*,
Xingfa Shen, *Member, IEEE*, Lufeng Mo, and Guojun Dai, *Member, IEEE*

Abstract—Recent advances in the application field increasingly demand the use of wireless camera sensor networks (WCSNs), for which localization is a crucial task to enable various location-based services. Most of the existing localization approaches for WCSNs are essentially interactive, i.e., require the interaction among the nodes throughout the localization process. As a result, they are costly to realize in practice, vulnerable to sniffer attacks, inefficient in energy consumption and computation. In this paper, we propose LISTEN, a noninteractive localization approach. Using LISTEN, every camera sensor node only needs to silently listen to the beacon signals from a mobile beacon node and capture a few images until determining its own location. We design the movement trajectory of the mobile beacon node, which guarantees to locate all the nodes successfully. We have implemented LISTEN and evaluated it through extensive experiments. Both the analytical and experimental results demonstrate that it is accurate, cost-efficient, and especially suitable for WCSNs that consist of low-end camera sensors.

Index Terms—Wireless camera sensor network, localization, mobile

1 INTRODUCTION

WIRELESS sensor networks (WSNs) have gained fast development in the past few years and become increasingly popular in many different application fields. More recently, the availability of low-cost hardware such as CMOS cameras and flash memory has fostered the development of wireless camera sensor networks (WCSNs) [1], which provide unprecedented advantages in a wide variety of applications [2], [3].

On one hand, WCSNs expand the application field of sensor networks. Most WSNs measure scalar physical phenomena like temperature, pressure, and humidity. WSN applications are thus restricted to simple purposes, such as environmental surveillance, traffic monitoring, and industrial control. WCSNs offer visual information (images & videos), which enable advanced applications, such as object tracking, advanced health care, person locator service, etc. On the other hand, WCSNs enhance the ability and efficiency of sensor networks. Because images and videos contain much more information than simple scalar sensor readings, WCSNs are able to make more accurate, intelligent, and efficient monitoring, judgments, and decisions.

Localization is a crucial issue in WCSNs, which involves determining the *locations* and *orientations* of camera sensor nodes.¹ Location information is a basic element in almost all the WCSN applications in practice. For example, the object tracking application requires the location information of sensor nodes to coordinate and schedule the tasking of sensor nodes along the object's movement trajectory. The person locator application requires the location information of sensor nodes to accurately report where the located person is.

For many location-based services in WCSNs, their design correctness and effectivity are highly sensitive to the location accuracy of sensor nodes. Thus localization in WCSNs demands high accuracy. Nevertheless, localization is a nontrivial task for WCSNs. Camera sensor nodes are generally ad hoc deployed. Global Positioning System (GPS) is deemed as a ready-to-use solution for civil applications, but GPS devices do not work in some environments, not to mention the prohibitive cost to equip every camera sensor node with a GPS module. Traditional localization approaches can't offer sufficient accuracy for WCSNs. Thus, localization is a significant issue to be studied in the context of WCSNs.

Recently, many localization approaches for WCSNs have been proposed. We find they are all essentially *interactive*. In other words, the localization process largely relies on interactions among the nodes to be located. "Interaction" in this paper means two-way communication between a pair of nodes. "Noninteractive" means only one side of a communicating pair sends signals. The other side keeps silent. Those interactive approaches have the following drawbacks.

First, most of the existing approaches are difficult to realize in practice. Some of them require expensive and complex hardware, such as mobile objects with specially

- Y. He is with TNLIST and the School of Software, Tsinghua University, Beijing 100084, China. E-mail: he@greenorbs.com.
- Y. Liu is with TNLIST and the School of Software, Tsinghua University, Beijing 100084, China, and the Hong Kong University of Science and Technology. E-mail: yunhao@greenorbs.com.
- X. Shen and G. Dai are with the College of Computer Science, Hangzhou Dianzi University, No. 2 Street, Xiasha, Hangzhou 310018, China. E-mail: shenxf@gmail.com, daigj@hdu.edu.cn.
- L. Mo is with the School of Information Engineering, Zhejiang Agriculture and Forestry University, Room 154, University Library, Zhejiang 311300, China, and Xi'an Jiaotong University. E-mail: molufeng@gmail.com.

Manuscript received 30 Jan. 2011; revised 28 Aug. 2011; accepted 22 Dec. 2011; published online 28 Dec. 2011.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-01-0049. Digital Object Identifier no. 10.1109/TMC.2011.278.

1. For simplicity, we use "camera sensor," "node," and "sensor node" interchangeably to represent "camera sensor node" in this paper.



Fig. 1. The forest deployment environment of GreenOrbs.

designed appearances or an intelligent robot [4], [5]. Some of them have unrealistic assumptions, e.g., assuming neighboring nodes always have overlapping Field of View (FOVs) [6], [7]. In practice, however, the deployment might be sparse so that FOVs of nodes do not necessarily overlap. Some other approaches demand complicated image processing [8], which are too resource-consuming to be executed on the low-end sensor nodes.

Second, interactive localization is vulnerable to malicious behavior such as sniffer attack [9]. For example, in the mobile-assisted localization, the information of locations where the mobile object receives controlling signals from the camera sensors actually imply the FOVs of the cameras. In the case of localization, two-way communication requires camera sensor nodes to send certain signals. Meanwhile, a malicious sniffer may masquerade as a normal mobile object and sniffs in the network. If the signals sent by camera sensor nodes are captured and collected by sniffer attackers, the network will face the risk of leaking location information of sensor nodes to undesired intruders. Thus, noninteractive localization is more suitable to many WCSN applications, especially the security-related ones.

Last but not least, most interactive approaches require every node to capture many images before locating itself. Note that the power consumption of image sensing is much higher than that of scalar sensing (e.g., to sense temperature or humidity) [10]. Besides, computation and processing over the numerous images also incur large energy consumption on the sensor nodes. Interactive localization appears to be energy inefficient, reducing the lifespan of a WCSN.

Our work is motivated by the need of location information of camera sensors in GreenOrbs [11]. GreenOrbs is a large-scale sensor network system that supports a wide variety of forestry applications [12]. Camera sensors are deployed in GreenOrbs to enable fire detection and rescue in the wild forest. Fig. 1 shows the deployment environment of GreenOrbs. It is a challenging task for GreenOrbs to accurately locate the camera sensors. According to our deployment experience in the forest, GPS does not work at many positions under the tree crowns, where the GPS antenna cannot receive sufficient satellite signals. Besides, typical price of an ordinary GPS module for sensor networks is generally around US\$100. It is prohibitive cost to equip a GPS module to every sensor node in a large-scale

WCSN. Besides, conventional localization approaches for WSNs yield unsatisfactory results in the complex forestry environments, due to the signal irregularity and environmental dynamics.

To address the above issues, in this paper we propose LISTEN, a noninteractive localization approach for WCSNs. LISTEN employs a mobile beacon node with very simple appearance. The whole localization process does not incur any interactions. While the mobile beacon node traverses the deployment area, every node only needs to silently listen to the beacon signals and capture a few images until successfully locating itself. Our main contributions are summarized as follows:

First, we propose the noninteractive lightweight localization approach, LISTEN. To locate itself, every node needs only a few times of image sensing and simple image processing. LISTEN has no complicated requirements on hardware or specific assumptions on the network deployments.

Second, we design the trajectory of the mobile beacon node, which guarantees successful localization of all the nodes. A node only needs to receive the beacons while sends nothing, thus never leaks location information to malicious sniffers.

Third, we have implemented LISTEN on the camera sensor nodes produced by ourselves. The experimental results demonstrate that LISTEN outperforms other approaches with low energy cost and high accuracy, using only commercial off-the-shelf devices.

The remainder of this paper is organized as follows: Section 2 briefly introduces the background of localization in WCSNs and reviews related work. Section 3 presents the design of LISTEN, followed by the theoretical proofs, analysis, and discussion in Section 4. Section 5 presents the implementation and the experimental results. We conclude this paper in Section 6.

2 BACKGROUND AND RELATED WORK

2.1 Localization in Wireless Camera Sensor Networks

WCSNs differ from the conventional WSNs with some distinct characteristics. Generally, multimedia data occupy much larger memory storage on the sensor nodes. The available network bandwidth in a WCSN, however, is rather limited. Real-time data collection is already a challenging issue in conventional WSNs [13], [14], not to mention the data collection in WCSNs. Besides, the power consumption of sensing once on a camera sensor (i.e., capture an image) is much higher than that on a scalar sensor, such as thermometer sensor. Such facts necessitate innovative designs of localization, sensing control and coordination, data collection, routing, and query processing techniques [2], [3], [15], [16], [17], [18].

Sensing model. Many existing works in conventional WSNs assume disk-based sensing [19], while WCSNs employ the directional sensing model. The FOV of a camera sensor is usually based on the pinhole model and shaped as a cone in 3D space or a sector in 2D plane, as shown in Figs. 2a and 1b, respectively. The FOV of a camera sensor is determined by the camera's extrinsic parameters (i.e.,

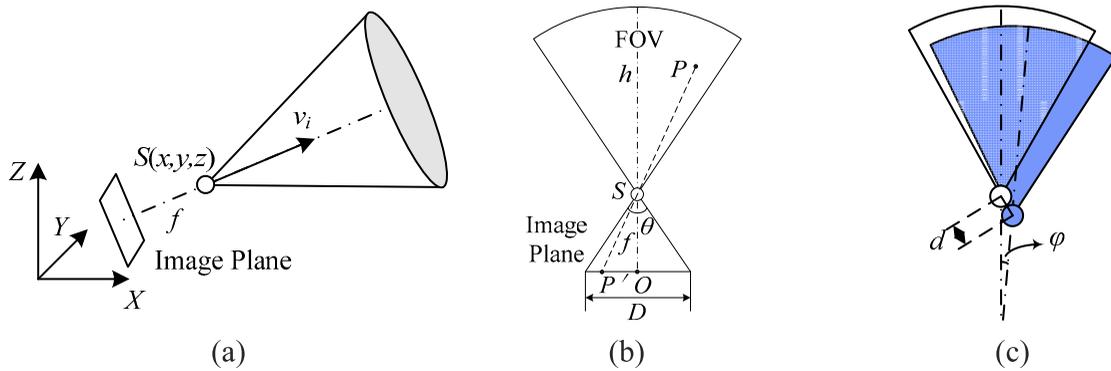


Fig. 2. (a) FOV of $S(x, y, z)$ based on the pinhole model. v_i is the unit vector denoting the orientation of S ; f is the focal length; θ is the angle of view. (b) The 2D pinhole model. h is the depth of view; D is the image plane. (c) Comparison between the real (transparent) and estimated (blue) FOVs.

location and orientation) and intrinsic parameters (including focal length, image format, principal point, etc.). Generally the intrinsic parameters are fixed for the camera sensors.

Localization of a camera sensor refers to determining its location and orientation. In Fig. 2a, locating node S means determining its coordinates (x, y, z) and unit orientation vector v_i . To function effectively, camera sensors demand very accurate localization. Fig. 2c shows an illustrative example in the 2D plane. The tiny estimation errors of the location (d) and the orientation (φ) can result in a significant estimation error of the FOV. In Fig. 2c, less than 80 percent of the real FOV is covered by the estimated one. The relative error is over 20 percent. Specifically, in the motivating applications of this work (such as fire detection, rescue, and intrusion detection), the localization accuracy with below 1 meter location error is desired. Most of those application purposes are security related. The application requirements are thus very critical on the precision, correctness, and effectiveness of WCSN operation. The effective FOV depth of a camera sensor is generally below 20 meters. One meter localization error generally leads to a relative FOV error of 5 percent or even larger. That means for at least 5 percent of the entire deployment area, the WCSN cannot offer reliable and effective monitoring service, which is clearly unacceptable for security-related applications.

2.2 Related Work

As far as we know, conventional localization approaches in WSNs yield errors of 1 meter at least [20], [21], [22], which is unacceptable for localization in WCSNs. GPS-based solutions are too costly and only work for outdoor applications. Range-based approaches like TOA, TODA, and AOA provide better accuracy but require extra hardware support [23].

As we mentioned in Section 1, the existing approaches for localization in WCSNs are all interactive, which are classified into two main categories: collaboration based and mobile assisted. The collaboration-based approaches borrow the idea of [8], [24] from the field of computer vision. Nodes locate themselves by collaboratively interpret the common visual information in their overlapping FOVs. Mobile-assisted approaches employ mobile objects (robots or beacon nodes) to assist the localization process. It is assumed the mobile objects always know their own coordinates, have

distinctive appearance, and can be controlled by the other nodes via wireless controlling signals.

Devarajan et al. [8] address the issue of calibrating distributed cameras. They model a camera sensor network with two undirected graphs: a communication graph and a vision graph. The communication graph is mostly determined by nodes' locations and the topography of the environment. In the vision graph, an edge appears between two nodes if they observe some of the same scene points from different perspectives. Edges in the vision graph can be automatically established by detecting and matching corresponding features between images. Nodes that image part of the same scene collaboratively interpret visual information among themselves. Then, each node calibrates itself independently based on information shared by nodes adjacent to it in the vision graph, using structure-from-motion techniques [24] from the computer vision literature. As a result of calibration, each node has an estimate of its own location and orientation.

Barton-Sweeney et al. [6] propose another collaboration-based approach for 3D localization. They assume a WSN includes camera sensors and a number of normal sensor nodes. The normal sensor nodes use modulated light emissions from a bright red LED to uniquely identify themselves to the cameras. Pairs of camera sensors then exchange information about the normal nodes observed in their FOVs, so as to compute their relative rotation and translation matrices. As reported in [6], generally a minimum of five nodes in the common FOV of two cameras is required to realize the above localization. The implementation in [6] adopts the normalized eight-point algorithm [26], which requires eight or more points in the common FOV of two cameras to compute the translation matrix.

Lee and Aghajan in [7] propose collaborative localization based on observation of a noncooperative moving target. A camera sensor node with an opportunistic observation of a passing target broadcasts a synchronizing packet and triggers image capture by its neighbors. In the cluster of participating nodes, the triggering node and a helper node construct a relative coordinate system. When a small number of joint observations of the target are made by the nodes, the proposed model allows for a decentralized or a cluster-based solution for the localization problem.

The above schemes have apparent limitations in the context of WCSNs. For example, the scheme in [8] requires the nodes to form local calibration clusters, each of which

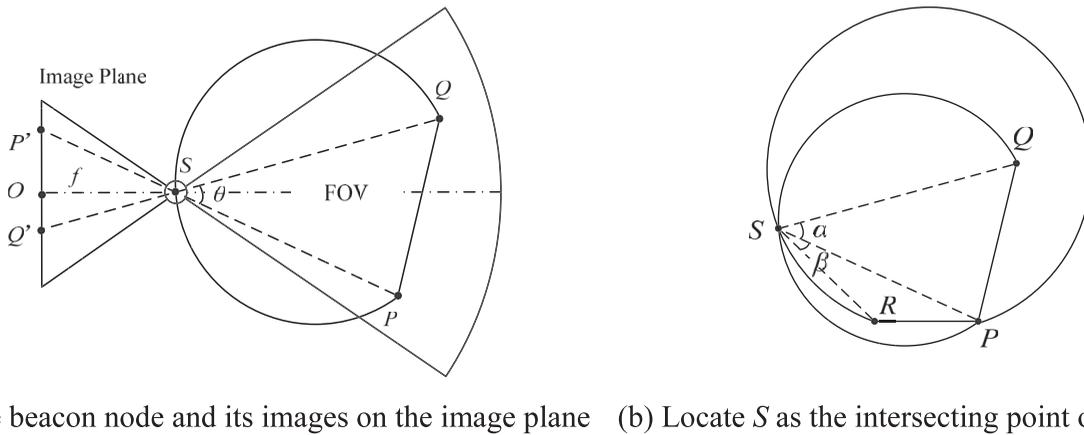


Fig. 3. Localization of node S in the 2D plane.

has a minimum of three nodes with eight common scene points. Schemes in [6] and [7] require camera sensors to identify common objects in their FOVs. The practical WCSNs, however, are very unlikely to satisfy such requirements. Camera sensors are often ad hoc and sparsely deployed, without dense clusters or overlapping FOVs among them. As a result, many nodes still cannot locate themselves using these schemes. The process to extract featured points or common objects from the images is also too costly for the low-end camera sensors motes.

Liu et al. propose in [25] a self-calibration protocol but cannot guarantee the successful localization for all the nodes in a WCSN. In [4], the authors propose a scheme of robot-assisted localization. In their targeted scenarios, camera sensors are deployed on the same plane (e.g., the ceiling of a room), which is parallel to the robot's motion plane (e.g., the floor). Their solution only addresses localization in the 2D plane. The robot knows its own coordinates and acts as a reference object in the captured images. The sensor network topology is modeled as a forest. Camera sensors in the same tree collaborate with each other to control the patrolling routes of the robots, so that every node can obtain sufficient observations to localize itself. When a tree of sensors is localized, the root node of that tree initiates a complicated process to instruct the robot to discover other adjacent trees. Both the localization and discovery processes incur large amount of communication cost on the sensors and the robot. The total energy consumption is easily affected by the node densities and network topologies. When the camera sensors are sparsely deployed and the entire network consists of many weakly connected components, the energy drains even faster because the discovery processes are triggered frequently.

3 DESIGN OF LISTEN

3.1 Assumptions

First we assume the intrinsic parameters of all the camera sensors are identical, fixed, and known beforehand, because we generally use the same type of camera sensors in a WCSN.

Second, we assume the communication range of the sensor node is not shorter than the depth of the camera's FOV. This is often true in practice. For example, the communication range of CC2420 radio using the maximum

transmission power is 100 meters in an outdoor environment [10], while a distance of 100 meters is in general longer than what needs to be monitored by a single camera sensor. Therefore, a beacon node and a camera sensor can communicate with each other as long as the former is in the FOV of the latter.

As for the mobile beacon used in LISTEN, it is a common sensor mote enabled with mobility. It always knows its own coordinates, as many existing proposals assume [4], [20]. Specifically, the mobile beacon's initial location is measured in two ways. In indoor environments, its initial location is manually assigned. In outdoor environments, its initial location should be a position where the reception of satellite signals is good. Then, we measure the initial location with a precise GPS device. After initial localization, the beacon node is programmed, in which the planned trajectory of its movement is preconfigured. Based on the initial location and the programmed trajectory, the beacon node always knows its own location. A featured tag (e.g., LED or a piece of colored paper) is attached to the beacon node to make it easily identified by the camera sensors and differentiated from the surroundings.

3.2 2D Localization

A mobile beacon node is employed to assist localization, which traverses the deployment area of a WCSN, passing a set of *beacon positions*. At each beacon position, it broadcasts a beacon signal that includes its current coordinates. On receiving the signal, a camera sensor captures an image and then tries to extract the featured tag of the beacon node from the image. We name the image of the featured tag as *beacon image*. The coordinates of beacon images are then used to calculate the angular distances of beacon positions, e.g., $\angle PSQ$ in Fig. 3a. In the next sections, we will elaborate how to utilize such angular information to locate the camera sensors.

3.2.1 The Localization Scheme

First we introduce the algorithm of LISTEN when all the nodes including the mobile beacon are in the same 2D plane. The orientation vectors of the cameras are in the plane too.

As shown in Fig. 3, S is a camera sensor. Abusing notations, we use $S(x_0, y_0)$ to denote the camera pinhole. O is the projection of S on the image plane, i.e., the image

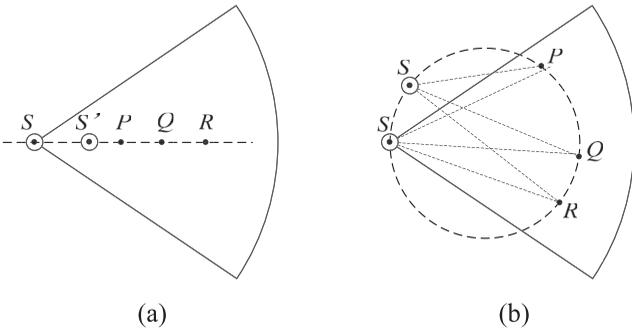


Fig. 4. Two special cases that don't yield unique solutions.

center. Suppose points $P(x_1, y_1)$ and $Q(x_2, y_2)$ are two beacon positions where the beacon node broadcasts beacon signals. P' and Q' are the corresponding beacon images. $\angle PSQ$ is called the *angular distance* between beacon positions P and Q . A camera sensor node maintains a coordinate system for its captured images, in which the image center is $(0, 0)$. Every point on the image has a pair of coordinates, measured in pixel. On receiving a beacon signal, a camera sensor node captures an image and then extracts the featured tag of the beacon node from the image. The output of extraction is the coordinates of the beacon image in the node's coordinate system. For example, Q' and P' are two beacon images. Suppose their coordinates are $(X_{Q'}, Y_{Q'})$ and $(X_{P'}, Y_{P'})$. Meanwhile, O is the center of the captured image. Its coordinates is $(0, 0)$. Thus, we have

$$|OQ'| = \sqrt{X_{Q'}^2 + Y_{Q'}^2}, \quad |OP'| = \sqrt{X_{P'}^2 + Y_{P'}^2},$$

$$\begin{aligned} \angle \alpha &= \angle QSP = \angle Q'SP' = \angle Q'SO + \angle OSP' \\ &= \tan^{-1} \left(\frac{|OQ'|}{|OS|} \right) + \tan^{-1} \left(\frac{|OP'|}{|OS|} \right) \\ &= \tan^{-1} \left(\frac{|OQ'|}{f} \right) + \tan^{-1} \left(\frac{|OP'|}{f} \right). \end{aligned} \quad (1)$$

Note that f is known. Since $|OQ'|$ and $|OP'|$ can be measured from the captured image, $\angle \alpha$ can be determined. Thus, S lies on arc \widehat{QP} , whose circumferential angle is $\angle \alpha$. Thus, we have

$$\frac{\overrightarrow{SP}}{|\overrightarrow{SP}|} e^{i\alpha} = \frac{\overrightarrow{SQ}}{|\overrightarrow{SQ}|}. \quad (2)$$

Similarly, by introducing a new beacon position R , we draw another arc \widehat{PR} with circumferential angle $\angle \beta$, as shown in Fig. 3b. Node S lies on \widehat{PR} . We have

$$\frac{\overrightarrow{SR}}{|\overrightarrow{SR}|} e^{i\beta} = \frac{\overrightarrow{SP}}{|\overrightarrow{SP}|}. \quad (3)$$

Thus, S is located as an intersecting point (the other is P) of arcs \widehat{QP} and \widehat{PR} . Solving Formulas (2) and (3) generally yields the unique coordinates of S .

In two special cases, however, the coordinates of S cannot be uniquely determined. In Fig. 4a, P, Q, R , and S are collinear. Any point left to P or right to R on the line is an eligible solution. In Fig. 4b, P, Q, R , and S are concyclic. Any

point on the arc \widehat{PSR} is an eligible solution. In Section 3.2.2, we will present the design of the mobile beacon trajectory to guarantee unique localization for every node.

Now we continue to calculate the orientation of node S , which is denoted by the direction of vector \overrightarrow{OS} in Fig. 3a. Recall that f and the coordinates of S and P are already known, while $|OP'|$ can be measured from the captured image. Thus, the unit orientation vector of S is calculated as follows:

$$\frac{\overrightarrow{OS}}{|\overrightarrow{OS}|} = \frac{\overrightarrow{SP}}{|\overrightarrow{SP}|} e^{i \sin \angle P'SO}, \quad \text{where } \angle P'SO = \tan^{-1} \left(\frac{|OP'|}{f} \right). \quad (4)$$

3.2.2 The Mobile Beacon Trajectory

According to the above scheme of LISTEN, we obtain the sufficient and necessary condition for a node to be uniquely located in the 2D plane: The node has captured three beacon images and the three beacon positions are not collinear or concyclic with it.

Recall that the localization process using LISTEN is completely noninteractive. The mobile beacon node doesn't have any prior or posterior knowledge of the other nodes' FOVs. When traversing the deployment area, the beacon node doesn't know whether a beacon position is captured by any node. Hence, it is nontrivial to satisfy the above sufficient and necessary condition.

This section presents the design of the mobile beacon trajectory. As long as the mobile beacon broadcasts beacon signals at selected beacon positions along the trajectory, it is guaranteed that every node in the WCSN, no matter where the node is, can be uniquely located.

We assume the nodes as well as their FOVs are included in a rectangle area without any obstacle. Recall that all camera sensors are assumed to have identical intrinsic parameters. We use h and θ to denote the depth and opening angle of the FOV, respectively. Let r denote the radius of the inscribed circle of the cameras' FOV. We have

$$r = \frac{h}{(1 + 1/\sin \frac{\theta}{2})}. \quad (5)$$

Fig. 5 shows the beacon trajectory in our design. Then, the deployment area is partitioned with equilateral triangles, whose side lengths all equal r . We select the vertices of the triangles as beacon positions. The beacon node starts from the upper left corner, moves along the trajectory, and broadcasts its coordinates at those beacon positions. Based on such a trajectory, every node is able to capture at least three beacon images, which correspond to three beacon positions on the plane that are neither collinear nor concyclic with the node itself.

Recall the deployment environment shown in Fig. 1. It is worth noticing that the forest application scenario does include some obstacles (e.g., trunks and shrubs). In a typical forest, the area of ground occupied by obstacles usually accounts for less than 5 percent of the area of the entire forest ground. In most cases the obstacles do not obstruct the movement of the mobile beacon node. According to the trajectory design, the mission of the beacon node is to broadcast beacon signals at the selected beacon positions. As long as the beacon node can reach those positions, the

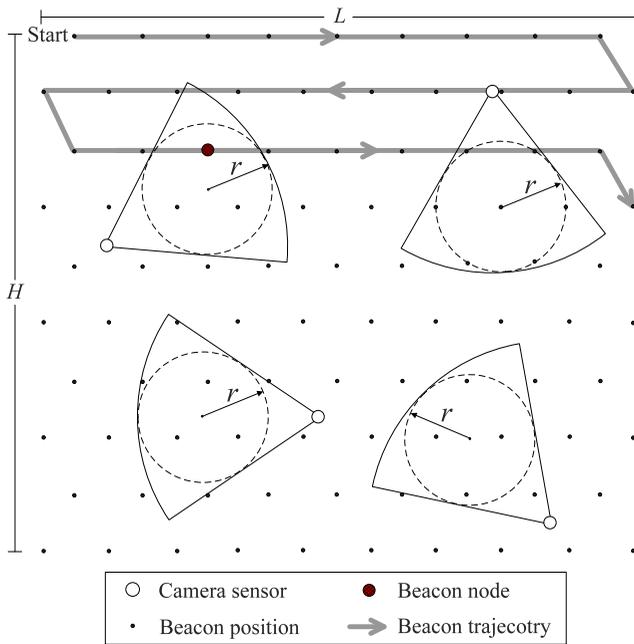


Fig. 5. The beacon trajectory.

localization process works. It is not necessary to make the beacon node move along straight lines.

Nevertheless, if an obstacle exactly stands on a selected beacon position, the beacon node may skip that beacon position and move to the next. It's an extremely rare case in practice that most beacon positions in a camera node's FOV are occupied by obstacles. When that happens, the camera sensor node might be unable to locate itself. We may need to reconfigure new beacon positions so as to make the node localizable.

3.3 Extension of LISTEN to 3D Localization

This section presents the design of LISTEN in the 3D space. We first present the calculation of angular distances between beacon positions, followed by the discussion on the uniqueness of localization. We then propose a selection method of beacon positions to guarantee locating an entire network of camera sensors in the 3D space.

3.3.1 Calculation of Angular Distance

The angular distances between beacon positions in the 3D space can be calculated similarly as that in the 2D plane. Fig. 6 shows an example. P and Q are two beacon positions. P' and Q' are the corresponding beacon images on the image plane of camera sensor S . The angular distance between P and Q , i.e., $\angle PSQ = \angle P'SQ'$.

We have

$$\begin{cases} |SQ'|^2 = |SO|^2 + |OQ'|^2 = f^2 + |OQ'|^2, \\ |SP'|^2 = |SO|^2 + |OP'|^2 = f^2 + |OP'|^2, \\ |P'Q'| = |SQ'|^2 + |SP'|^2 - 2|SQ'||SP'| \cos \angle P'SQ'. \end{cases} \quad (6)$$

Since $|OQ'|$ and $|OP'|$ can be measured from the captured image, solving (6) yields $\angle P'SQ'$.

3.3.2 Uniqueness of Localization

Suppose $\angle PSQ = \angle \beta$, then in the 3D space, S lies on a rotating surface. Taking any node on such a surface as

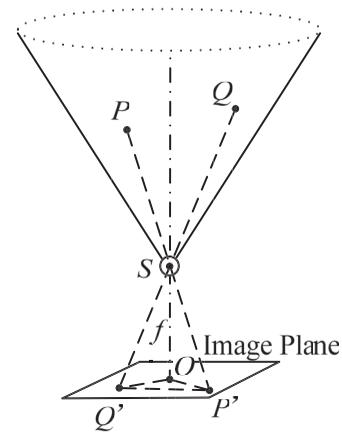


Fig. 6. Angular distance between 3D beacon positions.

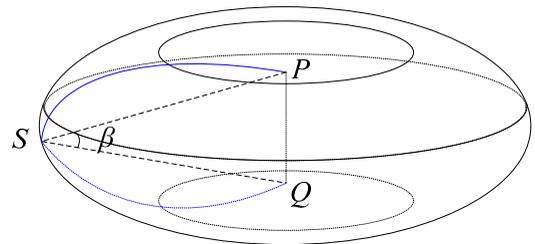
reference, the angular distance between P and Q identically equals to $\angle \beta$, as shown in Fig. 7. The generatrix of the rotating surface is an arc with circumferential angle of $\angle \beta$.

Now we examine whether three beacon positions are sufficient to uniquely locate a node. Suppose P , Q , and R are three different beacon positions. According to Fig. 7, we get three different rotating surfaces. Any intersecting point of the three surfaces is a candidate position of the camera sensor S .

As shown in Fig. 8a, using only three beacon positions generally yields more than one solution to the coordinates of S , namely S, S_1, S_2, S_3 , and their counterpoints across the plane PQR (for clear display, we do not show all of them). For $i = 1, 2, 3$, $\angle PSQ = \angle PS_iQ$, $\angle PSR = \angle PS_iR$, $\angle RSQ = \angle RS_iQ$. In other words, three beacon positions in the 3D space are insufficient to uniquely locate a sensor node.

Subsequently, at least four beacon positions are needed in 3D localization. For example, in Fig. 8b, there is an additional beacon position T on the line PR . Based on the angular distances among P, T , and R , one can locate node S on a circle (denoted by C). Line PR is perpendicular to the plane of C , and the center of C is on PR .

On the other hand, S is located on another two rotating surfaces. One of them has PQ as its central axis. The other has QR as its central axis. Obviously, at least one of the two rotating surfaces does not contain circle C . There must be two intersecting points, i.e., point S and its counterpoint S' between C and the rotating surfaces. Now the number of candidate solutions is reduced to two.

Fig. 7. The rotating surface: taking any node S on the surface as reference point, the angular distance between P and Q is identical.

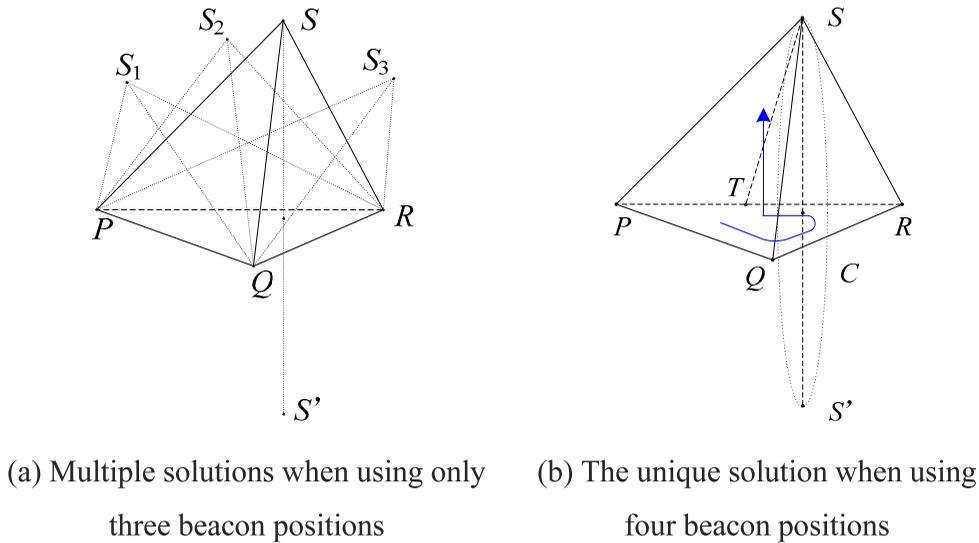


Fig. 8. On the uniqueness of 3D localization.

We further introduce *the right-hand rule* to filter out point S' . Specifically, the node decides a counterclockwise sequence of the beacon positions P, T , and R , simply based on the node's view of the beacon images. In Fig. 8b, the sequence is $Q \rightarrow R \rightarrow P$. When fitting the right hand with the sequence, the pollex points to the real position of the camera sensor.

To sum up, in the 3D space, generally a camera sensor S can be uniquely located by using four different beacon positions together with the right-hand rule. Meanwhile, the orientation of S can be calculated similarly as the case of 2D localization. We skip this part due to the limit of the paper length.

3.3.3 Selection of Beacon Positions

According to the result in Section 3.3.2, we can use four beacon positions to uniquely locate a node as long as the four positions satisfy the following condition: exactly three of them are collinear. Such a condition serves as general guidance in selection of beacon positions for 3D localization.

Selection of beacon positions in practice depends on the specific conditions of the deployment area. Here, we propose an option for selecting beacon positions.

We assume all the camera sensors and their FOVs are included in a cuboid deployment area without any obstacle. Recall that the FOV of a camera sensor in the 3D space is a shaped as a cone. As shown in Fig. 9, let r denote the radius of the inscribed sphere of the FOV. We use h and θ to denote the depth and opening angle of the FOV, respectively. In this case, (5) still holds.

Further, the deployment area is partitioned with cubes whose side lengths are all equal to l , such that the diagonal length of a cube is r . We have

$$l = \frac{h}{\sqrt{3}(1 + 1/\sin \frac{\theta}{2})}. \quad (7)$$

The vertices and the diagonal joins of the cubes are selected as the beacon positions. It is easy to see that the FOV of an arbitrary camera sensor covers at least nine beacon positions, namely the eight vertices and the diagonal join of a cube. Because every pair of diagonal vertices

(e.g., A and B in Fig. 9) and the diagonal join (K in Fig. 9) are collinear, there exist four covered beacon positions while exactly three of them are collinear. Using the beacon images corresponded to the four beacon positions, any camera sensor can uniquely locate itself.

4 ANALYSIS AND DISCUSSION

In this section, we mainly analyze the sensing cost and message overhead. We further discuss the error factors of localization using LISTEN and illuminate the relation between the location and orientation errors.

4.1 Sensing Cost

Different existing localization approaches in their implementations require different processors, different communication modules, and different camera modules. For example, the work in [8] adopts Canon G5 digital camera as the image source and uses a centralized server for processing. The work in [6] adopts iMote2 and OV7649 VGA camera. Thus, we do not directly compare their power consumption with LISTEN. Instead, we compare their sensing cost, which is counted as the number of image sensing. We use I_{min} and I_{max} to denote the minimum and maximum sensing cost, respectively.

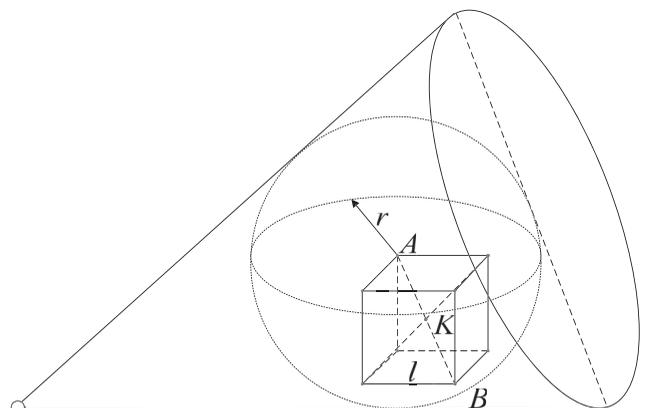


Fig. 9. The beacon positions in the 3D space.

In the 2D localization, a node needs at least three beacon images to locate itself. To save energy, the node tries to calculate its coordinates as soon as it captures three images. If the coordinates can be calculated, the nodes will no longer capture any images even if receiving the beacon signals.

Based on the above description, we have $I_{min} = 3$. As for the maximum sensing cost, it equals to the number of beacon positions within the communication range (denoted by r_c) of the node. We still use h and θ to denote the depth and opening angle of the FOV, respectively. We have

$$I_{max} = \left\lfloor \frac{4\pi r_c^2}{\sqrt{3}r^2} \right\rfloor, \text{ where } r = \frac{h}{(1 + 1/\sin\frac{\theta}{2})}. \quad (8)$$

We also evaluate the actual sensing cost of LISTEN in experiment, as shown in Table 2 later in Section 5.2.2. The minimum sensing cost is 3 in both experiments, while the average sensing cost is about 4.

LISTEN outperforms the approaches proposed in [6], [8], which both require a camera sensor node to capture at least eight images to locate itself. In the case of 3D localization, $I_{min} = 4$, while

$$I_{max} = \left\lfloor \frac{3\pi r^3}{2l^3} \right\rfloor, \text{ where } l = \frac{h}{\sqrt{3}(1 + 1/\sin\frac{\theta}{2})}. \quad (9)$$

4.2 Message Cost

Suppose the deployment area is a rectangle with length L and width H as shown in Fig. 5. To ensure every node can be localized, without regard to the locations and orientations of the nodes near the boundary, a large rectangle should be used in the trajectory planning of the mobile beacon.

Since the mobile beacon sends a beacon message at every beacon position, the message cost on the beacon node (denoted by C_M) is equal to the total number of beacon positions in the entire deployment area. Let h and θ to denote the depth and opening angle of the FOV, respectively. C_M is calculated as follows:

$$C_M = \left(\left\lfloor \frac{L}{r} \right\rfloor + 1 \right) \times r \times \left(\left\lfloor \frac{2H}{\sqrt{3}r} \right\rfloor + 1 \right),$$

$$\text{where } r = \frac{h}{(1 + 1/\sin\frac{\theta}{2})}.$$

4.3 Localization Error

The localization errors using LISTEN are mainly introduced from three aspects: the pinhole model, the instrumental error of f , and image processing. Such errors affect the calculated angular distance and in turn affect the localization results. Basically, when the angular distance gets larger, the node is located closer to its corresponding beacon positions.

The pinhole model is an ideal approximation of CMOS imaging. Actually, there are tiny distortions between the object and its image. Pixels at different regions on the image correspond to different refractions. Generally, pixels far from the center of the image are more refracted than the pixels near to the center. As a result, the calculated angular distances are usually smaller than the real ones.

The value of f used in our experiments is 400, measured in pixels. Due to the diversity of instruments,

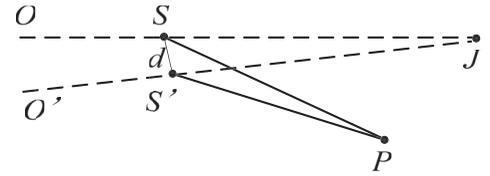


Fig. 10. Error of orientation.

however, the real values of f on the camera sensors differ slightly from each other. Subsequently, the calculated angular distances might be larger or smaller than the real ones. The errors introduced by image processing are also possible to yield larger or smaller angular distances, compared to the real ones.

As for the error of orientation, we examine it based on an example in Fig. 10. Suppose S' is the estimated location of S using LISTEN, $|SS'| = d$. P is a beacon position. Let e denote the maximum error (measured in pixels) in determining the coordinates of the beacon image of P . $\angle SJS'$ is the error of orientation in locating S . Then, we have

$$\begin{aligned} \angle SJS' &= 180^\circ - \angle SS'J - \angle S'SJ \\ \angle SPS' &= 180^\circ - \angle SS'P - \angle S'SP \leq \arctan \frac{d}{|S'P|} \\ \angle SJS' - \angle SPS' &= (\angle SS'P - \angle SS'J) - (\angle S'SJ - \angle S'SP) \\ &= \angle JS'P - \angle JSP \leq \arctan \frac{e}{f} \angle SJS' \\ &\leq \arctan \frac{d}{|S'P|} + \arctan \frac{e}{f}. \end{aligned} \quad (10)$$

Inequality (10) gives the upper bound of orientation error in the localization of LISTEN. Indeed, our experimental results demonstrate that $d/|S'P|$ is always below 0.05. Meanwhile, the practical value of e is no more than 2. Thus, the maximum orientation error is 3.12 degree.

5 IMPLEMENTATION AND EXPERIMENTS

5.1 Hardware

We have implemented LISTEN on our own-produced camera sensor nodes. The hardware design conforms to the paradigm of CMUcam3 [3]. It consists of three main components: a microcontroller, a CMOS camera chip, and a frame buffer. The ARM microcontroller NXP LPC2106 is a 32-bit 60 MHz ARM7TDMI with built-in 64 KB of RAM and 128 KB of flash memory. The image input is provided by an Omnivision OV6620 CMOS camera, which supports a maximum resolution of 352×288 at 50 frames per second. In order to allow the camera to operate at full speed and decouples processing on the CPU from the camera's pixel clock, a 50 MHz, 1 MB AL4V8M440 high-speed video FIFO frame buffer manufactured by Averlogic, is added to the camera sensor node.

5.2 Implementation and Experiments

Every camera sensor is connected with a TelosB mote [10], as shown in Fig. 11. For all the cameras, the depth and opening angle of the FOV are 6 meters and 52 degree, respectively. The mobile beacon node consists of a TelosB

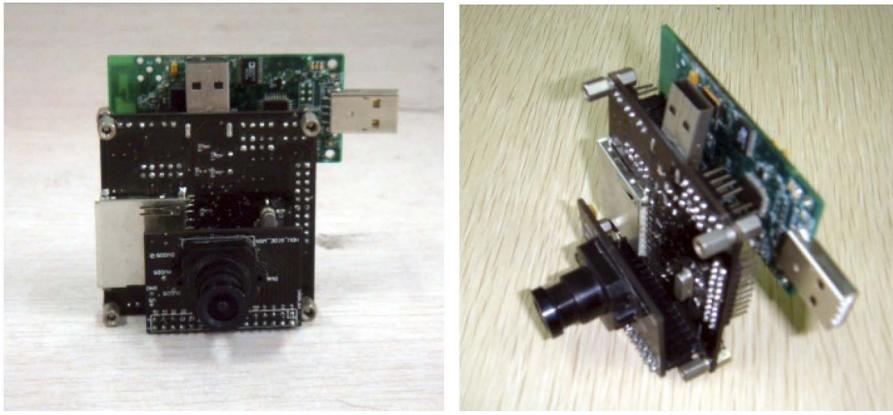


Fig. 11. The camera sensor node used in the experiments.

mote, a bracket on which the mote is mounted, and a vehicle that offers mobile capability. The movement of the vehicles follows a preconfigured beacon trajectory, which may then be realized by manual control or an automated motor. The red (or green) LED of the mote is kept on, which serves as the featured tag of the mobile beacon node.

We use three such beacon nodes simultaneously to speed up the localization process. Using three beacons in the experiments is just to divide the beacon trajectory into three segments. Specifically, we partition the whole deployment area from a $H \times L$ rectangle into three small $(H/3) \times L$ rectangles. Then, we let each beacon node traverse one segment. The benefit is the total time to finish locating the whole sensor network is reduced to one third of the original. Because the beacon nodes are identical, the localization result is same with the experiments using only one beacon. As we observe, LISTEN has similar performance in the 2D and 3D cases. In order to compare it with the conventional localization approaches that work in the 2D case, only the results in the 2D case are present here. The camera sensor nodes report their data to the sink node of the WCSN via a multihop data collection protocol.

5.2.1 Comparisons

The first experiment compares LISTEN with PI [20], a state-of-the-art mobile-assisted approach for localization in conventional WSNs. Instead of using the absolute values of RSSI, PI utilizes only the comparison relationship of the measured RSSI values between the mobile beacon and the other nodes to do localization. The resulting accuracy of PI is better than most existing localization approaches, as demonstrated in [20].

We conduct the experiments in a $12\text{ m} \times 12\text{ m}$ classroom. 10 sensor nodes are randomly deployed and located using LISTEN and PI, respectively. Fig. 12 plots the estimated node locations using PI and LISTEN, compared to their real locations. We can see the estimated locations using LISTEN is extremely close to the real ones. The average localization error is a necessary metric to evaluate the consistency of localization, for which there are two options. One of them is to average the localization errors of one node in multiple experiments. The other is to average the localization errors of all nodes in one experiments. Please note that in the evaluation experiments of LISTEN,

when the mobile trajectory is determined, the localization process of a certain sensor node is deterministic as well. On the other hand, considering that the camera sensor nodes are deployed at different positions with different orientations, we believe the average error of locating all sensor nodes in the experiments is a more appropriate metric to evaluate the consistency of localization accuracy. The experimental results in Table 1 also demonstrate that LISTEN apparently outperforms PI with much lower location error and more consistent accuracy. Moreover, the accuracy of RSSI-based localization is sensitive to various factors, such as signal fading, multipath, interference, and environmental dynamics. Image-based localization using LISTEN performs stably against such factors.

5.2.2 Evaluation in Different Environments

In this group of experiments, we evaluate LISTEN in two different environments. Other than the first deployment in the classroom, we further conduct another experiment with 12 nodes in a long corridor ($6\text{ m} \times 30\text{ m}$) in our office building. Due to the relatively long and narrow space in the corridor, the beacon positions used for locating a node are relatively far from the node. The resulting angular distances between beacon images are thus relatively small, which in turn results in larger relative errors.

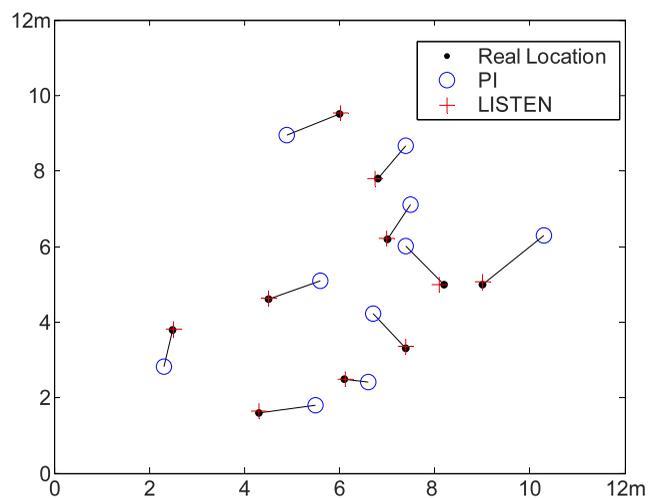
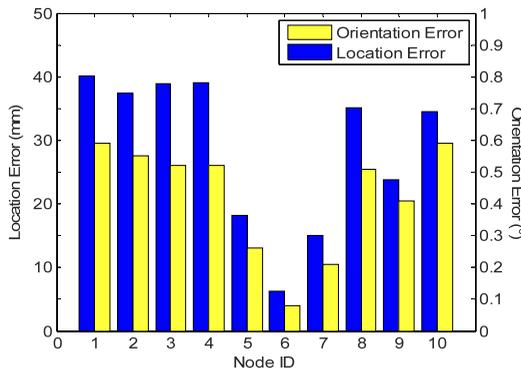


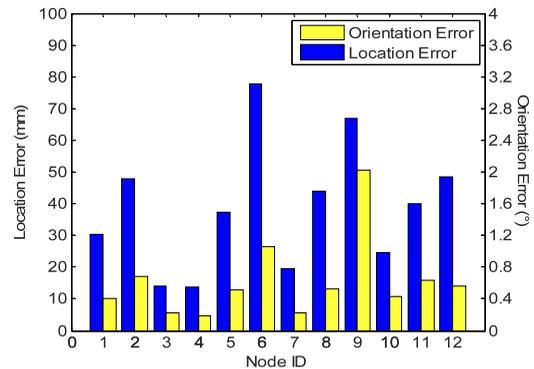
Fig. 12. Comparison between PI and LISTEN in the classroom experiment.

TABLE 1
Summary of the Experimental Results

	Localization Error (mm)		Orientation Error (°)		Relative FOV Error (%)	
	Avg.	S.D.	Avg.	S.D.	Avg.	S.D.
PI	1,173	301.2	N/A	N/A	N/A	N/A
LISTEN (classroom)	28.82	12.08	0.42	0.18	1.00	0.45
LISTEN (corridor)	38.73	20.03	0.62	0.50	1.31	0.77
LISTEN (webcam)	23.31	7.57	0.45	0.13	N/A	N/A

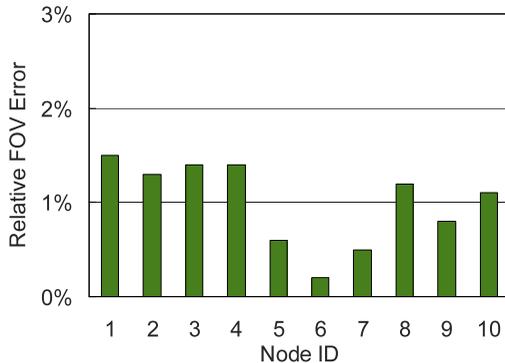


(a) Classroom experiment

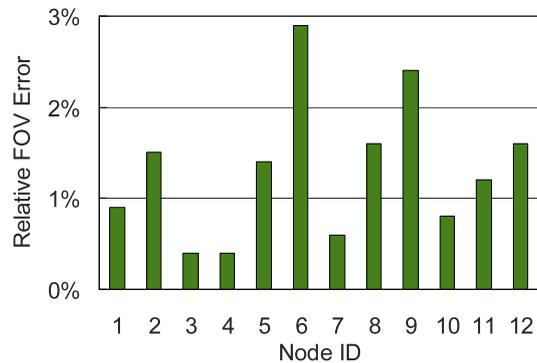


(b) Corridor experiment

Fig. 13. Location and orientation errors in two experiments.



(a) Classroom experiment



(b) Corridor experiment

Fig. 14. Relative FOV errors in two experiments.

We measure the location error, orientation error, and relative FOV error (the percentage of the real FOV that is missing in the estimated FOV) for all the nodes. Figs. 13 and 14 show the comparison results.

We can see that the results in the classroom are more accurate and consistent than those in the corridor (note the different Y-axis scales). This is mainly due to the difference in the angular distances between the beacon images, as we mentioned in the first paragraph of Section 5.2.2.

Moreover, the resulting averages of the relative FOV errors are only 1 and 1.31 percent, respectively. In other words, LISTEN is capable of supporting the location-based services of WCSNs with very accurate localization.

It is also interesting to see that nodes 5, 6, and 7 in the classroom and nodes 3 and 4 in the corridor are more

accurately located than the other nodes. We then go through and compare the images captured by all the nodes. The finding is that the beacon images captured by those five nodes are all near to the center of the image. Such beacon images correspond to lower distortions than the beacon images captured by the other nodes, hence resulting in smaller location errors. Taking (10) into account, the value of ϵ is smaller with those five nodes. So the orientation error is also smaller.

The above finding indicates that using different subset of beacon images yields localization results with different accuracies. It may be feasible to design a refining procedure with LISTEN, which intelligently selects the most appropriate subset of beacon images to achieve the best localization results. Please note that using LISTEN, the localization

TABLE 2
Sensing Cost of a Node in the Two Experiments

	Min. Sensing Cost	Max. Sensing Cost	Average	S. D.
Classroom	3	8	4.30	1.64
Corridor	3	7	4.00	1.21

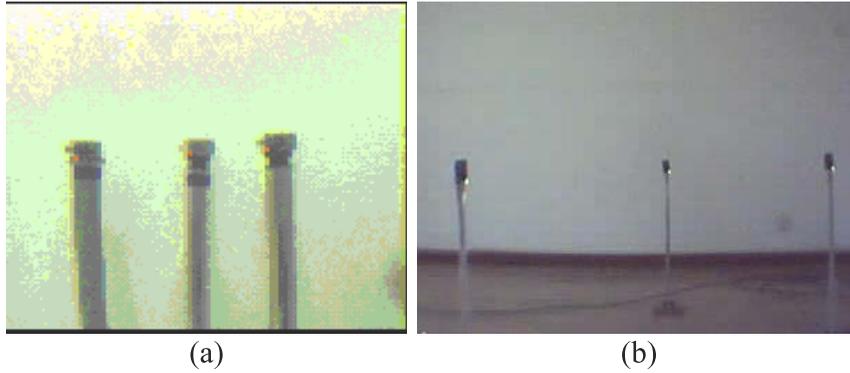
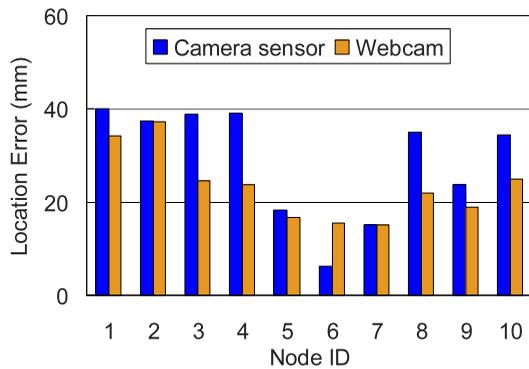
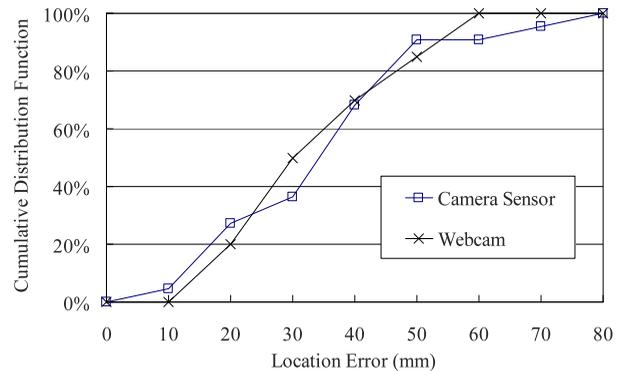


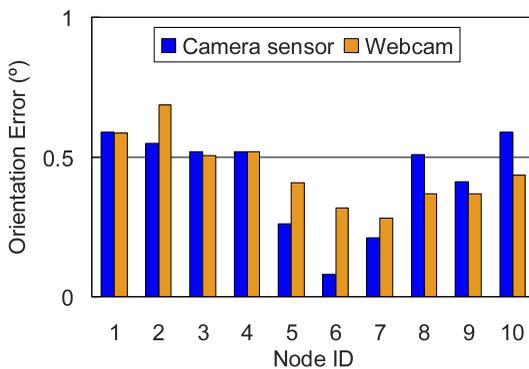
Fig. 15. Images captured by (a) a camera sensor, (b) a webcam.



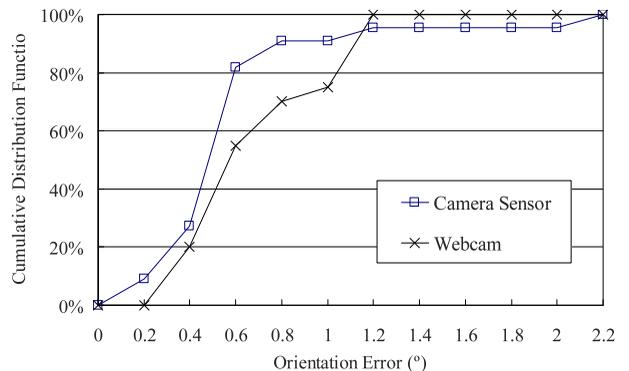
(a) Location errors in the classroom experiment



(b) Cumulative distribution of location errors



(c) Orientation errors in the classroom experiment



(d) Cumulative distribution of orientation errors

Fig. 16. Comparisons between the results on camera sensors and webcams.

accuracy of a node is solely related to factors within a node's FOV. Because the FOV is fixed, the localization result will not be affected by the deployment area. Meanwhile, it's true that in large area, beacon node at a distant position might leave an unclear image on the camera sensor node, which might introduce error in localization. In order to ensure that a camera sensor node keeps clear beacon images, we may restrict the communication range of the mobile beacon node.

Subsequently, a camera sensor node will not capture images for a distant beacon node. Setting of the communication range actually depends on the deployment environment. We will address these issues in the future work.

Table 2 shows the sensing cost of all the camera sensor nodes in two experiments. Corresponding to the analysis in Section 4.2, here the sensing cost on a node is measured by the number of image sensing the node executes before it is

located. We can see the sensing cost in both experiments is relatively low.

5.2.3 Impact of Image Quality

Fig. 15 shows two pictures. The right one taken by a 300K-pixel webcam has obviously higher quality than the left one taken by our camera sensor. It could be a major concern that the low image quality of camera sensors might degrade the performance of localization using LISTEN. To further evaluate the impact of image quality on the localization accuracy of LISTEN, we have ported it to run over the webcam pictures.

We deploy a webcam at exactly identical locations and orientations as those camera sensors in the classroom and corridor experiment. Fig. 16 compares the experimental results. Due to the page limit, we only present the detailed location and orientation errors in the classroom experiment, as shown in Figs. 16a and 16c. Figs. 16b and 16d compare the overall cumulative distribution. Interestingly, we find the accuracy of LISTEN is almost not affected by the image quality. The results on the webcam are only slightly better than those on the camera sensors.

Now, we briefly summarize the experiments. LISTEN realizes very accurate localization in WCSNs. Compared to the conventional RSSI-based approach, LISTEN performs apparently better and more stably under various environmental settings. Moreover, the performance of LISTEN is not significantly affected by the image quality. Thus, it is especially suitable to localization of the low-end camera sensors.

6 CONCLUSION

WCSNs present novel application fields of the WSN technology. Localization, although has been well studied in the literature of WSNs, remains a challenging issue in WCSNs. Various approaches have been proposed but are all essentially *interactive*. Those approaches thus suffer vulnerability to malicious attacks, poor applicability, and excessive overhead.

This paper proposes LISTEN, noninteractive localization for WCSNs. LISTEN is energy efficient and easy to implement in practice. By employing a mobile beacon with simple appearance to assist localization, every node to be located only needs to passively listen to the beacon signals and does not send any packet throughout the whole localization process. By calculating the angular distances between beacon positions, a node needs as few as three times of image sensing to locate itself. The implementation and experimental results demonstrate that LISTEN is easy to realize in practice and lightweight, suitable for a wide variety of WSN applications.

In our future work, we will address the issue of location refining, as we mention in Section 5.2.2. We also plan to carry out large-scale implementation of LISTEN with our own-produced camera sensors in the GreenOrbs system.

ACKNOWLEDGMENTS

This research was supported in part by the NSFC program under Grant No. 61170213, the China Postdoctoral Science

Foundation Grant No. 2011M500019, the NSFC Major Program under Grants No. 61190110, the NSFC program under Grants No. 61190113 and 60803126, Zhejiang Provincial Natural Science Foundation under Grant No. Z1080979, and the program for Zhejiang Provincial Key Innovative Research Team on Sensor Networks.

REFERENCES

- [1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks*, vol. 51, pp. 921-960, 2007.
- [2] Y.-C. Wang, Y.-F. Chen, and Y.-C. Tseng, "Using Rotatable and Directional (R&D) Sensors to Achieve Temporal Coverage of Objects and Its Surveillance Application," *IEEE Trans. Mobile Computing*, pp. 404-417, 2011.
- [3] A. Rowe, D. Goel, and R. Rajkumar, "FireFly Mosaic: A Vision-Enabled Wireless Sensor Networking System," *Proc. IEEE 28th Int'l Real-Time Systems Symp. (RTSS)*, 2007.
- [4] H. Lee, H. Dong, and H. Aghajan, "Robot-Assisted Localization Techniques for Wireless Image Sensor Networks," *Proc. IEEE Third Ann. Comm. Soc. Sensor and Ad Hoc Comm. and Networks (SECON)*, 2006.
- [5] I. Rekleitis and G. Dudek, "Automated Calibration of a Camera Sensor Network," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, 2005.
- [6] A. Barton-Sweeney, D. Lymberopoulos, and A. Savvides, "Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks," *Proc. Int'l Conf. Broadband Comm., Networks and Systems*, 2006.
- [7] H. Lee and H. Aghajan, "Collaborative Node Localization in Surveillance Networks Using Opportunistic Target Observations," *Proc. ACM Int'l Workshop Video Surveillance and Sensor Networks*, 2006.
- [8] D. Devarajan, R.J. Radke, and H. Chung, "Distributed Metric Calibration of Ad Hoc Camera Networks," *ACM Trans. Sensor Networks*, vol. 2, pp. 380-403, 2006.
- [9] H. Bidgoli, "Hacking Techniques in Wireless Networks," *The Handbook of Information Security*. John Wiley & Sons, 2006.
- [10] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. IEEE Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN)*, 2005.
- [11] GreenOrbs, <http://www.greenorbs.org>, 2012.
- [12] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X.-Y. Li, and G. Dai, "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest," *Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys)*, 2009.
- [13] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over Sensor Networks," *Proc. IEEE 27th Int'l Real-Time Systems Symp. (RTSS)*, pp. 291-302, 2006.
- [14] T.F. Abdelzaher, S. Prabh, and R. Kiran, "On Real-Time Capacity Limits of Multihop Wireless Sensor Networks," *Proc. IEEE 25th Int'l Real-Time Systems Symp. (RTSS)*, 2004.
- [15] T. Yan, D. Ganesan, and R. Manmatha, "Distributed Image Search in Sensor Networks," *Proc. Sixth ACM Conf. Embedded Network Sensor Systems (SenSys)*, 2008.
- [16] P. Kulkarni, D. Ganesan, and P. Shenoy, "The Case for Multi-Tier Camera Sensor Networks," *Proc. ACM Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pp. 141-146, 2005.
- [17] W.-C. Feng, E. Kaiser, W.C. Feng, and M.L. Baillif, "Panoptes: Scalable Low-Power Video Sensor Networking Technologies," *ACM Trans. Multimedia Computing, Comm., and Applications*, vol. 1, pp. 151-167, May 2005.
- [18] R. Holman, J. Stanley, and T. Ozkan-Haller, "Applying Video Sensor Networks to Nearshore Environment Monitoring," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 14-21, Oct.-Dec. 2003.
- [19] S. Kumar, T.H. Lai, and A. Arora, "Barrier Coverage With Wireless Sensors," *Proc. ACM MobiCom*, 2005.
- [20] Z. Guo, Y. Guo, F. Hong, X. Yang, Y. He, Y. Feng, and Y. Liu, "Perpendicular Intersection: Locating Wireless Sensors with Mobile Beacon," *Proc. IEEE Real-Time Systems Symp. (RTSS)*, 2008.
- [21] H.-I. Chang, J.-B. Tian, T.-T. Lai, H.-H. Chu, and P. Huang, "Spinning Beacons for Precise Indoor Localization," *Proc. Sixth ACM Conf. Embedded Network Sensor Systems (SenSys)*, 2008.

- [22] Z. Zhong and T. He, "Achieving Range-Free Localization Beyond Connectivity," *Proc. Seventh ACM Conf. Embedded Network Sensor Systems (SenSys)*, 2009.
- [23] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices," *Proc. Fifth Int'l Conf. Embedded Networked Sensor Systems (Sensys)*, 2007.
- [24] O. Faugeras, F. Lustman, and G. Toscani, "Motion and Structure from Motion from Point and Line Matches," *Proc. IEEE First Int'l Conf. Computer Vision (ICCV)*, 1987.
- [25] X. Liu, P. Kulkarni, P. Shenoy, and D. Ganesan, "Snapshot: A Self-Calibration Protocol for Camera Sensor Networks," *Proc. Int'l Conf. Broadband Comm., Networks and Systems*, 2006.
- [26] R. Hartley, "In Defense of the Eight-Point Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580-593, June 1997.



Yuan He received the BE degree from the University of Science and Technology of China in 2003, the ME degree from the Institute of Software, Chinese Academy of Sciences, in 2006, and the PhD degree from Hong Kong University of Science and Technology. He is a member of the Tsinghua National Lab for Information Science and Technology. He now works as a postdoctoral fellow with Professor Yunhao Liu. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE, the IEEE Computer Society, and the ACM.



Yunhao Liu received the BS degree in automation from Tsinghua University, China, in 1995, and the MS and PhD degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is a professor at Tsinghua University and a faculty member at the Hong Kong University of Science and Technology. His research interests include peer-to-peer computing, pervasive computing, and sensor networks. He is a senior member of the IEEE and the IEEE Computer Society.



Xingfa Shen received the BE degree in electrical engineering and the PhD degree in control science and engineering from Zhejiang University, China, in 2000 and 2007, respectively. He is currently an associate professor at Hangzhou Dianzi University. His research interests include wireless sensor networks and embedded systems. He is a member of the IEEE.



Lufeng Mo received the BE degree from Xi'an Jiaotong University and the ME degree from Pecking University. He is currently working toward the PhD degree in the Department of Computer Science and Technology at Xi'an Jiaotong University. His research interests include wireless sensor networks and pervasive computing. He is the author/coauthor of more than 10 peer-reviewed journal and conference publications and holds two patents.



Guojun Dai received the BE and ME degrees from Zhejiang University in 1988 and 1991, respectively, and the PhD degree from the College of Electrical Engineering, Zhejiang University, in 1998. He is currently a professor and the deputy dean of the College of Computer Science, Hangzhou Dianzi University, China. He is the author or coauthor of more than 20 papers and books in recent years, and holds more than 10 patents. His research interests include biomedical signal processing, computer vision, embedded systems design, and wireless sensor networks. He is a member of the IEEE.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**