2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) | 978-1-6654-8643-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SECON55815.2022.9918581

VisBLE: Vision-Enhanced BLE Device Tracking

Wenchao Jiang[1†], Feng Li[2†], Luoyu Mei[2], Ruofeng Liu[3], and Shuai Wang[2*]
[1]Singapore University of Technology and Design, Singapore, Singapore,[2]Southeast University, Nanjing, China
[3]University of Minnesota, Minnesota, United States
wenchao_jiang@sutd.edu.sg, tengtenglf@gmail.com, lymei-@seu.edu.cn, liux4189@umn.edu, shuaiwang@seu.edu.cn

***Abstract*—IoT devices have evolved from providing remote connection to being an essential component of the Metaverse. The integration of IoT and vision technologies has been incubating emerging applications such as vision-enhanced device tracking and remote education/medicine/maintenance. Despite the exciting vision, practical challenges include coordinate transformation, angle estimation, target mapping, and personal error. Instead of proposing yet-another localization approach, we propose a novel vision-enhanced device tracking system, called VisBLE. VisBLE takes advantage of the new localization capability introduced in BLE 5.1 and advances in vision technologies for high accuracy, robust, and intuitive BLE device tracking. There are two novel technical mechanisms: i) a rotation-based wireless localization mechanism that accurately and robustly locates the BLE transmitter in the camera coordinate and ii) a homography-based matching mechanism that identifies target BLE devices with high accuracy on the camera screen. We prototype VisBLE and deploy it on the smartphone (i.e., Nexus 5X) and development board (i.e., CC26X2 + BOOSTXL-AOA). Our results show that VisBLE outperforms the state of the art in both angular accuracy and position accuracy.**

I. Introduction

With the advent of the Internet of Things (IoTs) era, the number of IoT devices surges in recent years. By 2035, the number of IoT devices is estimated to reach a trillion [1]. IoT nowadays not only provides ubiquitous connection but also integrates with the Metaverse for a better user experience. One promising direction in IoT is to integrate IoT applications with AR/VR applications, such as intuitive IoT devices tracking and remote education/medicine/maintenance applications [2].

A conceptual use case is illustrated in Figure 1. There are various IoT devices in an office, such as the lamp, the fan, and the printer, embedded with BLE chips. Instead of looking for each in a long Bluetooth device list, a user turns on the camera and scans the whole room. The connectable devices will appear on the screen as clickable AR objects. The user then operates on these devices directly from the screen. Such a novel user experience is way more intuitive and immersive compared with traditional solutions. In addition to the office scenario, other application scenarios include factories, malls, restaurants, and hospitals wherever intuitive operation on IoT devices is needed.

To achieve this vision, a straight forward solution is to provide high-accuracy device tracking in both the wireless area and the computer vision area. There are extensive wireless-based and vision-based solutions respectively in the literature [3]–[6]. However, the intersection of the two lines of studies is still an open question, especially on mobile devices with constrained radio and camera module with ordinary users. First, advanced wireless solutions rely on wide-bandwidth signal and a large-size antenna array to achieve centimeter-level accuracy [7], which is not common for mobile devices. Second, cross-modality opportunities and constraints are not fully explored in such a novel user scenario. Finally, a practical application should have consistent performance for both ordinary users and experience users. VisIoT [4] is a pioneer work that achieves IoT tracking in AR by projecting an IoT device's angular information to the camera view, but it fails to explore more cross-modality opportunities to further reduce accumulating errors and output ambiguity.

Fig. 1: A conceptual scene of the VisBLE use case in the office to operate on various BLE devices from the screen.

Instead of proposing yet another wireless localization approach, in this paper, we aims to address the gaps between the wireless technology and the vision technology for them to cooperate seamlessly and supplement each other. We present VisBLE, an vision-enhanced BLE device tracking system. VisBLE is built with two key designs: (i) an angular-based wireless localization mechanism that links the wireless and vision localization technologies to accurately and robustly locate the BLE transmitter in the camera coordinate and (ii) a homography-based matching mechanism that provides the depth information of target BLE device to push the localization accuracy and resolve the non-line-of-sight (NLoS) issue. Detailed technical contributions are as follows:

†Both authors contributed equally to this research.
*Shuai Wang is the corresponding author.

- To the best of our knowledge, this work is the first to explore a vision-enhanced BLE device tracking with the new direction-finding feature. It is a fundamental building block for AR/VR applications.
- On the wireless side, we propose a novel azimuth and elevation estimation mechanism based on the Level meter (an electronic level) that advances inertial measurement unit (IMU) sensors with lower accumulating errors.
- On the vision side, we utilize the homography matrix to extract depth information from the point cloud. It helps separate objects in the foreground and the background of the camera view, and also addresses the Non-line-of-sight (NLoS) problem which troubles most vision-based technologies.
- We have prototyped and evaluated the performance of VisBLE on the smartphone camera and multiple BLE devices. Our results show that VisBLE tracks a BLE device with 3.4° angular error and $8.4cm$ position error in median which outperforms the state-of-the-art by 48% and 8% respectively. Its overall device tracking accuracy is over 90%.

II. MOTIVATION

IoT has seen rapid development in the past decade beyond ubiquitous connection and starts to be integrated into the Metaverse to empower emerging applications such as remote education/medicine/maintenance through vision technologies like AR/VR [2]. Such an integration will power Metaverse by emerging large amount of IoT data into the virtual world as well as providing IoT applications a more intuitive and immersive 3D user interface. However, the conventional operation and localization schemes on IoT devices are not designed for the new paradigm, bringing a series of technical challenges during the integration. In this work, for specification, we design an vision-enhanced BLE device tracking to demonstrate the idea. This work has the potential to enhance the performance and user experience of popular BLE tracking services such as AirTag and Tile Mate [8].

A. Limitations of the SOA

In the literature, the device tracking problem is usually formulated as a 3D wireless localization problem [3], [9], [10]. These works typically rely on advanced radio technologies such as large bandwidth and antenna array which are often costly to be equipped on mobile devices if not impossible. Popular wireless radios on mobile devices, such as BLE, are intrinsically restricted in localization accuracy due to bandwidth and antenna. In addition, there is limited access to the physical layer signals open to the end-users per the specifications. On the other hand, advances in computer vision can locate the visual objects on the camera screen and mark them with bounding boxes or masks [11], [12]. Vision technologies are usually used to recognize visual landmarks in the environment in a simultaneous localisation and mapping (SLAM) task [13]. However, we argue that the visual patterns alone cannot tell whether an object is a BLE device or not since a BLE device can come in all shapes and colors. VisIoT [4] is a pioneer work that projects ZigBee angular information into the camera screen for AR applications, but it requires software-defined-radio to extract low-level phase information. In addition, though mentioned in the paper, the work does not explore the opportunities of vision technologies in such an AR application. Finally, errors due to users' operation are very relevant to the practical performance but not fully studied.

B. Opportunities and Challenges

Opportunities: Our opportunities come from the new 'direction finding' feature introduced since BLE specification 5.1 [14], where two localization elements are proposed, namely the angle of arrival (AoA) and the angle of departure (AoD). AoA and AoD measure the angle of the target device through the differences of the signal arriving multiple antenna. Compared with most existing BLE devices that are only capable of proximity measurement through RSSI, the AoA and AoD are much more accurate due to the new capability of angular measurement. In addition, the angular information is a physical quantity that can be linked to the vision technologies. In other words, instead of extracting objects' visual patterns from the picture/video footage, we are more interested in the angular relationship between the camera and the target objects in the camera screen. In such a manner, BLE devices with different shapes and colors can be tracked without pre-training.
Challenges: The technical challenges of the work are in three folds. First, though the AoA information can be read from BLE devices, it is an angle on the antenna plane, which has to be transformed to the camera coordinate. In addition, to pin an object in the 3D world, the azimuth and elevation angles have to be estimated. Finally, accumulating measurement and personal errors will be introduced during device tracking. They need to be addressed to increase the accuracy of the system.

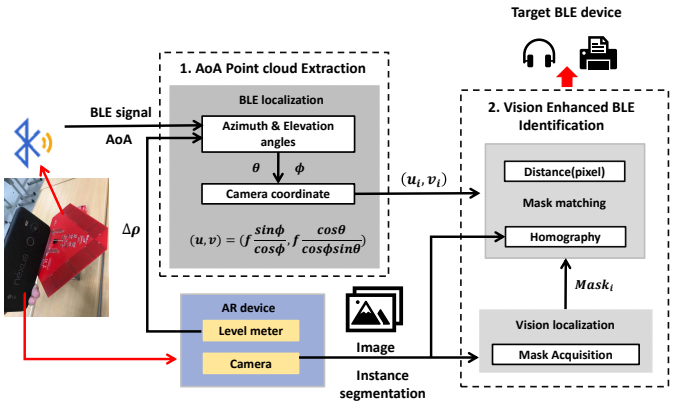


Fig. 2: VisBLE overview.

III. VISBLE OVERVIEW

Figure 2 illustrates the overall workflow of VisBLE to track a BLE device in an AR application. In a nutshell, VisBLE takes advantage of the angular information to link the wireless measure and the visual point cloud to track BLE devices

with high accuracy and robustness. It is comprised of two components, i.e., (i) the AoA point cloud extraction and (ii) the vision enhanced BLE tracking.

- **AoA point cloud extraction** 2D AoA information is extracted from BLE devices supporting the 'direction finding' feature in specification 5.1 and onward. The information will then be projected onto the camera coordinate. To further estimate the azimuth and elevation angles of the target BLE device in the 3D world, the Level meter reading and a controlled rotation are applied to the mobile device under UI assistance.
- **Vision enhanced BLE identification** Images in the camera screen are segmented into semantic mask segments. During the controlled rotation, a technology called 'Homography' is applied to analyze the depth of the mask segments for mask matching and dealing with NLoS case.

Compared to the state of the art [4], in VisBLE, the wireless measure and the vision technologies are deeply coupled during device tracking. In addition, the use of Level meter and the UI assistance during the process can help ordinary users to largely reduce accumulating errors during operation.

IV. AOA POINT CLOUD EXTRACTION

This section illustrates the VisBLE design in detail. We first give background about the AoA information obtained from BLE 5.1 devices. Then we introduce how to estimate the azimuth and elevation angles from the AoA.

A. Preliminary of BLE Device Tracking

Angle of Arrival (AoA): Direction finding is a feature introduced in BLE specification 5.1. It enables a BLE receiver to directly obtain the angle of arrival (AoA) of another BLE transmitter. Let Θ be the AoA and Ψ be the phase difference of the incident signal, we have

$$cos(\Theta) = \frac{\lambda \Psi}{2\pi d}, \tag{1}$$

where λ is the wavelength of the incoming signal and d is the interval between antennas.

However, the AoA information alone is not enough to determine the position of a BLE device in (i) the camera screen and (ii) the 3D world. To address these issues, coordinate transformation and an estimation of the azimuth and elevation angles are necessary.

Coordinate in camera screen: According to [4], the coordinate of BLE device in the camera screen (u, v) can be determined by

$$(u, v) = \left(f\frac{sin\phi}{cos\phi}, f\frac{cos\theta}{cos\phi sin\theta} \right) \tag{2}$$

where f is the focal length, ϕ is the azimuth angle, and θ is the elevation angle.

Estimating azimuth and elevation: Now the question becomes how to estimate the azimuth angle ϕ and elevation angle θ. The relationship between the two angles is formulated as

$$\Psi = \frac{2d\pi}{\lambda} cos(\phi) sin(\theta) \tag{3}$$

where Ψ is the phase difference of the signals from two antennas. In the literature, we can fix one angle (either azimuth and elevation) and estimate the other through a controlled device motion tracked by gyroscope or the inertial measurement unit (IMU) sensor using gyroscope internally [4].

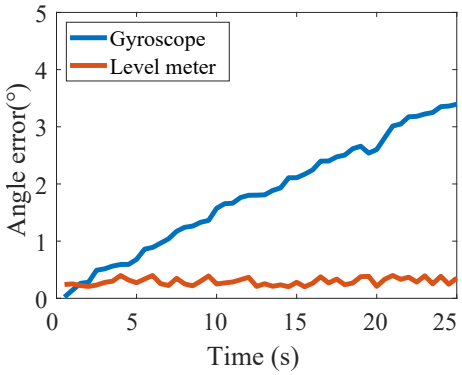


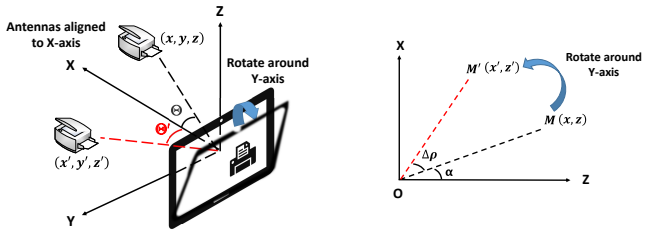
Fig. 3: Gyroscope vs Level meter

B. Level Meter Based Angle Estimation

However, one issue with gyroscope-based angle estimation is the accumulating errors, because the angle of rotation is accumulated by a real-time integrator. We argue that tracking the mobile phone gesture in real-time is not necessary. Instead, obtaining the *start and final gestures* of the mobile device is more than enough. By doing so, we can reduce the accumulating errors. In a mobile device, a Level meter can do the job. It captures the angle between the mobile device and the horizontal plane in a certain state, which is a state quantity. To verify the hypothesis, in Figure 3, we measured the accumulating error of the gyroscope vs a Level meter. We can see the gyroscope introduces 2.5° error after 18 seconds while the Level meter has almost 0° angular error.

Inspired by this observation, we propose to utilize the Level meter available in the mobile devices to estimate the azimuth and elevation angles. Specific operations are as follows: We first take the angle between the device and the horizontal plane obtained by the Level meter in the AR device ρ and the AoA Θ obtained by the Bluetooth device as a known quantity. Then we rotate the device to change the position of the antenna and get another Level meter reading. The rotation angle is calculated from the Level meter reading difference before and after the rotation. After that, we calculate the azimuth and elevation angles from the rotation angle and AoA. The azimuth and elevation angles determine the pixel coordinates of the target Bluetooth device in the camera screen following Equation 2.

We hereby derive how to calculate the azimuth angle ϕ and elevation angle θ from the rotation angle and AoA. Without loss of generality, we assume the receiving antenna is located on the X-axis and the rotation is around the Y-axis as shown in Figure 4. When the user rotates the AR device around the Y-axis, according to the relativity of the movement, the target device is rotating in the opposite direction around the Y-axis. As shown in Figure 4, the coordinates of the target Bluetooth device before rotation are $(x, y, z)^T$, and the coordinates after



(a) Rotate the device along the Y-axis. (b) Projection of the target device on the XZ plane.

Fig. 4: Spatial angle relation.

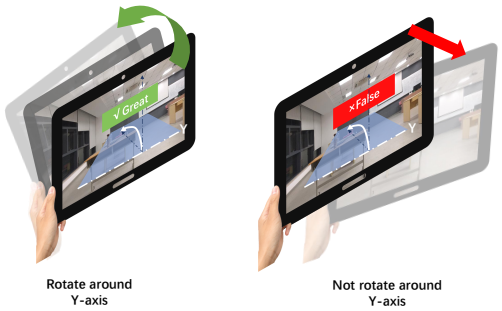


Fig. 5: Rely on prompt instead of tripods.

rotation are $(x', y', z')^T$. M is the projection of the target device on the XZ plane before rotation. Its coordinates are (x, z). Similarly, the coordinates of the projection of the rotated point M' are (x', z'). Then we have:

$$z = OMcos\alpha \qquad x = OMsin\alpha \tag{4}$$

$$z' = OM'cos(\Delta\rho + \alpha) \qquad x' = OM'sin(\Delta\rho + \alpha) \tag{5}$$

where $\Delta\rho$ represents the angle at which the user rotates the device around Y-axis and α represents the angle between M and Z-axis. Also, $OM = OM'$, since the rotation is around the Y-axis. When we rotate the device around the Y-axis, its Y-axis coordinate remains unchanged: $y = y'$. Based on this observation, the relation is as follows:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} cos\Delta\rho & 0 & sin\Delta\rho \\ 0 & 1 & 0 \\ -sin\Delta\rho & 0 & cos\Delta\rho \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6}$$

Since the horizontal angle $\Delta\rho$ and AoA Θ are known quantities, we calculate 1) elevation angle θ 2) azimuth angle ϕ

$$\theta = arccos\frac{cos(\Theta') - cos(\Delta\rho)cos(\Theta)}{sin(\Delta\rho)} \tag{7}$$

$$\phi = arccos\frac{cos(\Theta)}{sin(\theta)} \tag{8}$$

Through the calculation of the above Equations, we estimate the azimuth and elevation angles of the target Bluetooth device relative to the AR receiving device, and then estimate the position of its signal source in the camera screen. It should be noted that if the sender is outside of the camera's viewing angle, (u, v) values from Equation 2 become infeasible (i.e., either smaller than zero or larger than the number of pixels of the video).

Compared with the state of the arts, the Level meter based angle estimation has several advantages:

- **Less accumulating errors.** We only need to know the difference between the angles before and after the rotation without tracking the angle changes during the rotation.
- **Easier operation.** We only need to rotate around a single axis once. Also, during the rotation of the device, we no longer perform threshold estimation on AoA [4], just collect the AoA and Level meter readings.
- **Low computational cost.** The proposed angle estimation algorithm causes low computing overhead and runs in real-time in mobile AR devices.

C. UI-assisted Device Rotation

Personal error is a factor that cannot be ignored in such an AR application, especially during the rotation operation. To largely alleviate this kind of error, we have designed a user interface that can assist the device rotation operation. It is inspired by the UI in the camera's 'panorama mode' which can help users to keep balance while moving the camera during the panorama shooting.

In particular, we designed a simple UI with one virtual track and an arrow moving in the track as shown in Fig. 5. This UI shows users how much they offset the right track in real time and in a intuitive way. Figure 6 shows the CDFs of the position error when the rotation mode is changed. As shown in the figure, the median positioning error dropped from 280 pixels to 246 pixels when we removed tripods and use UI-assisted. While we removed UI, the error increases. Thus, VisBLE is suitable for both tripods and non-tripods devices, and UI-assisted helps for improve the accuracy while keep the operation convenient.

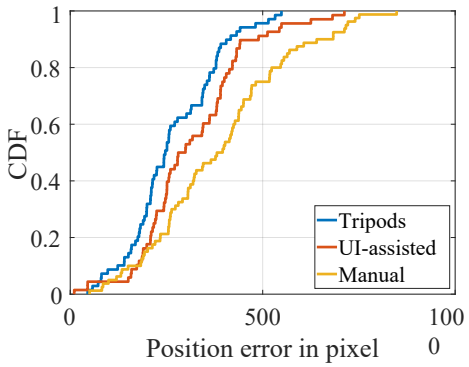


Fig. 6: Position errors: Tripods vs. UI-assisted vs. Manual.

V. VISION ENHANCED BLE TRACKING

Through the azimuth and elevation angle estimation mechanism, we estimate the position of the target device in the camera screen, but that is only halfway done. The pure wireless solution is limited in two aspects: (i) relative low accuracy due to the simple antenna and fast-changing wireless environment and (ii) scattered point cloud that cannot directly mask the target object. In this section, we introduce how to borrow ideas from computer vision technology to improve BLE device tracking accuracy.

A. Mask Acquisition

In contrast to the wireless localization approaches that generate point clouds, the computer vision algorithms segment the image into semantic objects. One state-of-the-art framework is the Mask R-CNN [12]. Mask R-CNN first realizes the instance semantic segmentation of the captured image at the pixel level. It then clusters the semantic information of each feature point to generate a Mask of the target object. Compared with edge detection-based approaches, such as SIFT [15] or SURF [16] that are based on color gradients, segmentation masks in Mask R-CNN are more rational and interpretable with pre-trained models over large labeled datasets. So it is more suitable for the device tracking tasks in this work.

B. Matching through Pixel Proximity

Although Mask R-CNN obtains accurate segmentation masks of the semantic objects in the image, the matching between the segmentation masks and the wireless point cloud is a technical challenge. One straightforward approach is matching through the pixel proximity in the camera screen. Recall that in Section IV we have transformed the AoA point cloud to the camera coordinate so that it can be naturally mapped to the closest masks.

One issue in such an approach is the sparse AoA point. Note that the number of pixels on the screen is several orders higher than the number of AoA points. For example, the camera in the experiment is with a resolution of 4032 $\times$ 3024 pixels, while the number of AoA points is only dozens. It causes unstable matching results jumping among several nearby objects. To resolve this limitation, we apply a Gaussian filter to model the point cloud distribution and further look for the most likely position of the AoA transmitter. According to the law of large numbers, we assume the spatial distribution of the point cloud approximates a two-dimensional Gaussian distribution centred at the signal transmitter.

Another issue with the pixel proximity is its inability to distinguish objects in the foreground from those in the background. In some cases, the point cloud may be scattered on two objects that are close to the camera screen but far away in the actual scene. For example, one object could be a bottle close to the user while the other is a clock hanging on the remote wall. The matching approach through pixel proximity may identify the wrong object. It is a challenging issue because the camera screen itself is a 2D plane which does not contain any depth information.

C. Matching through Homography

To distinguish objects at different depths of field, we employ the homography technology from computer vision. Recall that the camera pose changes in the azimuth and elevation estimation algorithm. The camera viewing angle also changes accordingly during the process. In the field of computer vision, and two camera screens of the same planar surface are related by homography (assuming a pinhole camera model). From the homography matrix, we derive the camera's rotation and further figure out the depth of the target object.

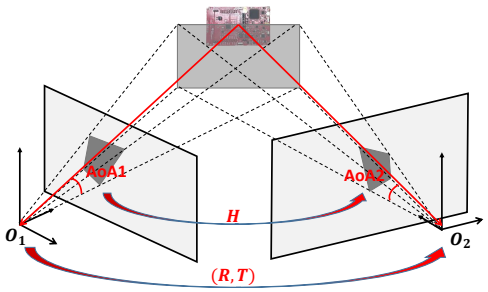


Fig. 7: Homography transformation

Homography matrix: As illustrated in Figure 7, a homography matrix describes the relation between the position projection of the feature point on the two frames of the camera on the same plane [17]:

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}, \tag{9}$$

where $(u_1, v_1, 1)^T$ represents the image point in image 1, $(u_2, v_2, 1)^T$ is the image point in image 2. Image 2 is transformed into image 1 through the Homography matrix H. During the azimuth and elevation estimation algorithm, we change the pose of the camera. Such a movement generates a unique homography matrix in the camera screen for each mask at a certain depth in the real scene, assuming each Mask approximates a plane. In other words, by calculating the homography matrix of each mask in the camera screen, we figure out the depth of the object.

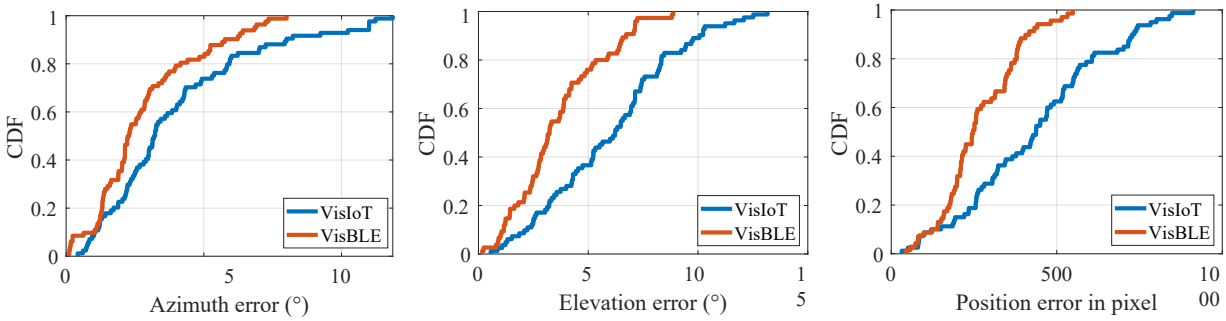
Homography matrix and depth: The formula between the homography matrix and depth is as follows [17]:

$$H = K(R + T\frac{1}{d}N^T)K^{-1}, \tag{10}$$

where K is the camera internal parameter, R is the camera rotation matrix, T is the translation matrix, d is the depth, and N is the normal vector of the plane in the frame.

From Figure 7, we observe that the change of camera pose produces two frames of images (as the black dotted line). The Homography matrix of the plane is calculated by the matching feature points in the two frames. Since each mask contains thousands of pixels, we apply edge detection approaches to extract feature points from the masks [15], [16]. From Equation 10, we find that the homography matrix H is inversely proportional to the depths d. Based on it, we leverage the homography matrix as a characteristic feature to distinguish masks of different depths to enhance the matching algorithm. In addition, Homography can naturally address the NLoS case through the *mismatch* between the directions extracted from the AoA and Homography. As illustrated in Fig. 7, when a BLE tracker of interest is behind the foreground scene, its AoA changes will be less than the angular changes calculated from pixels on the foreground scene through Homography.

A more robust matching mechanism: Note that we have projected the AoA point cloud to the camera screen. The AoA point cloud is supposed to be located on the same plane as the



(a) Azimuth estimation error. (b) Elevation estimation error (c) Position error

Fig. 8: CDFs for estimation angles and position error comparison between VisBLE and VisIoT

target device as the signal is sent from a target device. According to the principle of homography, the homography matrix of the AoA point cloud and the target device should always be similar no matter how the camera pose changes. Based on that, by calculating the similarity between each Mask's homography matrix and the AoA point cloud's Homography matrix, we distinguish Masks at the same depth as the AoA point cloud.

Dealing with multiple devices: Though our discussion so far is focused on one BLE device for simplicity, the multi-device case also has native support thanks to the 'connectionless mode' in BLE 5.1 direction-finding [14]. Under the connectionless mode, one BLE tracker actively advertises a special packet known as the continuous tone extension (CTE) with which its AoA information is obtained by a nearby locator.

To put everything together, the matching algorithm works as follows. First a Mask R-CNN algorithm is applied to processes the original picture frame/video footage and generates several Masks which represents an recognized object. Then wireless AoA point cloud projected to the camera screen are filtered by the Masks following two rules: i) the proximity rule where mask are mapped to AoA point cloud with the closest camera screen distance and ii) the consistence in angular information where the AoA information is compared with angular information calculated from the Homography matrices. Mappings with inconsistent angular information will be removed from the result set.

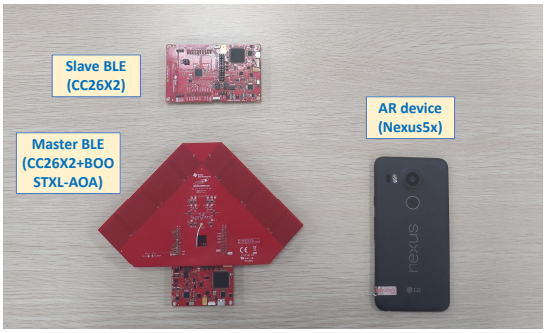


Fig. 9: Experiment setting for VisBLE.

VI. IMPLEMENTATION AND EVALUATION

In this section, we present the implementation and evaluation to quantitatively understand the system performance.

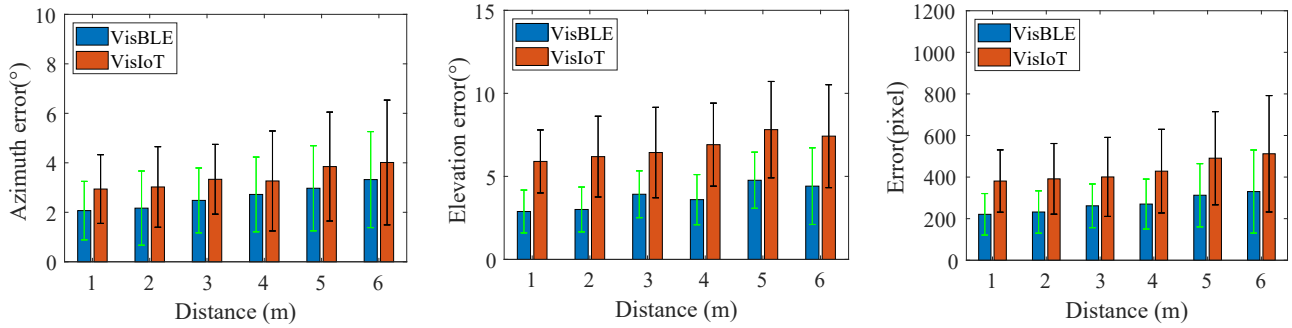
A. Implementation

We build VisBLE, which locates the position of the Bluetooth device signal source in the video and provide interactive Bluetooth device objects. Figure 9 illustrates our implementation setting platform of VisBLE. By default, connectionless Bluetooth devices broadcast and send data packets with 100ms intervals. The resolution of the smartphone screen is 4032 $\times$ 3024 pixels. Note that the use of development board is for its openness to second-development. The ideas works for other off-the-shelf BLE devices or chips that support 'direction finding', such the Tile Mate BLE tracker [8] and Nordic's nRF52810 chip [18].

B. Angular and Position Accuracy

Angular accuracy: We first evaluate the angular accuracy of VisBLE, say the azimuth and elevation angles. For the performance evaluation, we conducted experiments while varying the position of the target BLE device. The distance between the target device and the receiving device is between 1m and 6m. Figure 8(a) and 8(b) show the CDF of the azimuth and elevation errors with 300 experiments. The median errors of the azimuth and elevation angles of VisBLE are 2.2° and 3.1°, respectively. In comparison, VisIoT yields much higher azimuth and elevation angle estimation errors, with median errors of 3.4° and 6.4°, respectively. In other words, our azimuth and elevation estimation algorithms have error reduced by 35% and 51%, respectively. It is observed that VisBLE and VisIoT are comparable in azimuth estimation, and have a huge difference in elevation estimation. The reason is that VisBLE uses the state quantities which brings much less cumulative errors during the rotation.

Position accuracy: We then evaluate the errors between the estimated target and the actual target in the camera coordinate. Figure 8(c) shows the corresponding CDF of the positioning pixel error. The median and the 95-percentile errors of VisBLE are $8.4cm$ and $16.4cm$, respectively. In comparison, VisIoT yields higher position errors. The median error of VisIoT is $14.9cm$ and its 95-percentile error is $26.7cm$. Normalized according to the number of pixels on the diagonal, the positioning errors of VisIoT are 8.6% and 15.5% of the screen diagonal, respectively VisBLE achieves not only accurate tracking in the camera coordinate but also centimeter-level localization accuracy.

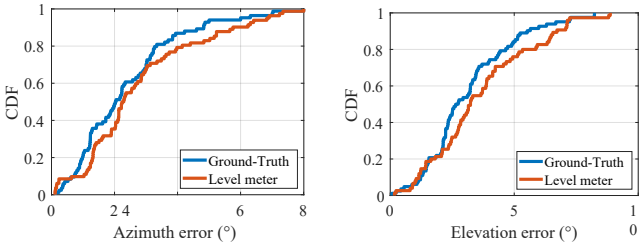


(a) Azimuth estimation error. (b) Elevation estimation error (c) Position error

Fig. 10: Evaluation in various distances

Impact of distance: Since Bluetooth is suitable for short-distance recognition and transmission, the scenario in this paper focuses on the indoor environment. We discuss the impact of distance on VisBLE positioning in an indoor environment. Figure 10(a) and Figure 10(b) show the median angles estimation errors within 95% confidence interval in various distances. And, Figure 10(c) shows the position error. We observe that VisIoT has slightly lower azimuth estimation accuracy than VisBLE, and VisBLE is much better than VisIoT in elevation estimation. The conclusion is consistent with the previous results. It once again proves that the calculation of the two-state angles by Level meter is more reliable than the calculation of rotation by Gyroscope integration.

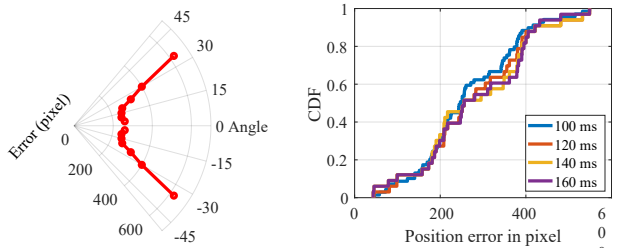
In addition, we observe that the error of VisBLE marginally grows as the distance increases from $1m$ to $6m$. However, we find that as the distance increases ($1m$ to $6m$), the median error increases slowly. The greater the distance, the greater the variance. The main reasons are as follows: i) With the same angle estimation error, larger distance leads to larger localization errors [19]. ii) In a long distance scenario, the multi-path effect becomes obvious which leads to increased errors. In all these cases, the performance of VisBLE is better than VisIoT. In addition, a horizontal comparison shows that the estimations of elevation angles are in general more accurate than those of azimuth angles.



(a) Azimuth estimation error. (b) Elevation estimation error.

Fig. 11: Angles estimation error with ground-truth rotation.

Impact of Level meter: In our azimuth and elevation estimation algorithm, the level tracks the motion of the device and records the angle of rotation. Here, we analyze how much estimation error is caused by the imperfect Level meter. We replace the rotation angle obtained based on the Level meter with the ground-truth rotation angle. Figure 11 shows the performance comparison. The results show that the Level meter is reliable, where VisBLE does not seriously suffer from the drift problem. When the ground-truth rotation angle is used, the median azimuth estimation error reduces from 2.2° to 2.0°, and the median elevation angle estimation error reduces from 3.2° to 2.6°. Regardless of the CDF estimation of azimuth or elevation, the overall trend of the curve is very similar to Ground truth.



(a) The performance of rotation angle on localization. (b) The performance of frame interval on localization.

Fig. 12: The performance of rotation angle and frame interval on localization.

Impact of rotation angle: In Section IV-B, we rotate the device to change the position of the antenna to estimate the azimuth and elevation angles. One might wonder how much the user needs to rotate the device. It depends on the angle of view of the camera and the range of Antennas receiving. Figure 12(a) show that the performance of rotation angle on localization. The result proves that the angle of rotation is no more than 20°, the positioning error is hardly affected, and the median of position error is $8.5cm$. The angle of rotation is not difficult for users to rotate that amount. In addition, we found that when the rotation angle exceeds 20°, the positioning error increases significantly. The reason for the excessive error may be due to the restricted antenna position of the development board. In brief, the results in Figure 12(a) shows that VisBLE does not require users to perform too complex rotation operations to achieve high-precision positioning.

Impact of frame interval: In the previous experiment, connectionless Bluetooth devices broadcast and send data packets at 100ms intervals by default. However, in scenarios with too many IoT devices, data packet loss and few received data packets may occur. To adapt to the needs of the actual environment, it is necessary to verify VisBLE's demand for traffic. For this reason, we evaluated the impact of VisBLE

data packet transmission interval on localization accuracy. Figure 12(b) shows the CDFs of the position error when varying the frame interval. From the collected curve, we changed the frame interval and evaluated the positioning error of VisBLE. As shown in Figure 12(b), the frame interval increased from 100ms to 160ms, and the median positioning error increased from 8.3 to 9.9cm. Our results prove that the frame interval time does not seriously affect the positioning performance of VisBLE. In addition, VisBLE of low traffic demand helps its application in large-scale IoT device scenarios.

C. Enhancement through Vision Technology

In this subsection, we evaluate the accuracy of VisBLE after Bluetooth positioning and visual positioning are coupled.

Performance in different scenarios: To confirm the reliability of VisBLE in more general, we deploy VisBLE, the above two methods, and the no-vision-based AoA point cloud method in different scenarios, including a large classroom, a small classroom, and a lab. In a large classroom, there is a large distance between objects, while in a laboratory environment, the interference object is closer to the target device. As shown in Figure 13, AoA point cloud alone obtains higher recognition accuracy in large classrooms, but in small classrooms and laboratories, it is significantly reduced. After enhancing by visual method, regional segmentation enhances the accuracy of AoA point cloud positioning objects, and the homography roughly distinguishes objects in front and back.

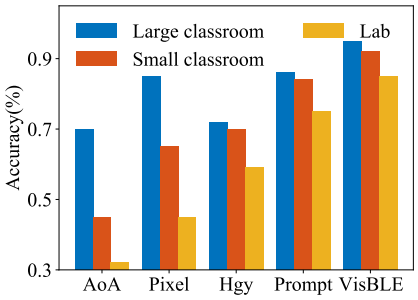


Fig. 13: Accuracy of 5 methods in different scenarios.

Figure 13 shows the accuracy of different methods to identify the correct target device. The methods applied in this experiments are as follows:

- **AoA**: relying only on the AoA information without vision enhancement.
- **Pixel**: projecting AoA information onto camera screen and use the proximity for mask matching.
- **Homography (Hgy)**: only leveraging the homography matrix to match masks.
- **Prompt**: combining pixel proximity and homography matrix for device tracking but operated by a volunteer relying on the visual UI during operation.
- **VisBLE**: ombining pixel proximity and homography matrix for device tracking but operated by an experienced operator.

We can observe from Fig. 13 that: First approaches with vision enhancement, i.e., all except 'AoA' do improves the BLE tracking performance. Second, point cloud proximity, i.e., 'Pixel', outperforms homography approach, i.e., 'Hgy', in large space but underperforms in small rooms. That is because there are more interfering objects in small scenes. Proximity-based method cannot differentiate interfering objects in different depth levels. Third, approaches combining the proximity and homography, i.e., 'Prompt' and 'VisBLE', have the best performance. Also the help of visual UI lets a volunteer performs closely to experienced operator. The overall device tracking performance can reach over 90%.

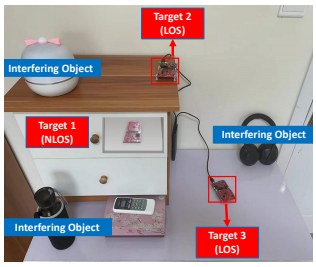


Fig. 14: Environment of tracking multiple BLE tags.

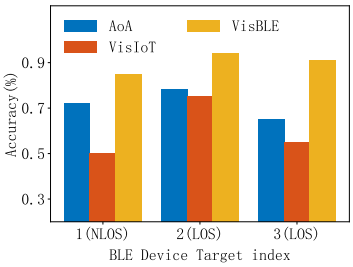


Fig. 15: Performance of tracking multiple BLE tags.

Performance with multiple BLE devices: To verify the robustness of VisBLE, we arranged multiple target devices in the experimental scene, as shown in Figure 14. There are three target devices, one of which is located in the built-in drawer, and several interference objects in the picture to simulate the actual built-in Bluetooth device. Figure 15 shows the result of the tracking accuracy of each device under different methods. Experimental results show that VisBLE's recognition accuracy in the scene of multiple devices is still stable at about 90%, and the devices of NLoS can also be identified through the directional function of Bluetooth signals. VisBLE is more robust in real-world scenarios than VisIoT.

The reasons of tracking failure: There are two main reasons for VisBLE tracking failure: First, due to the limitations of indoor positioning, we obtained the AoA with the median error of 3.4°, the resulting point cloud area with the median error of 8.4cm. Second, as described in section V-C, the homography makes a rough distinction between front and rear objects when the position changes in a small range($\leq 1m$), and cannot make a clear distinction between objects with close depth.

In summary, our extensive experiment results have shown the effectiveness of the Level meter and the vision technology in providing an accurate and robust BLE tracking. It paves the way for a lot of AR applications in the future.

VII. RELATED WORK

Vision-based localization: With the rapid development of computer vision technology, we are able to identify and locate known IoT devices in the video [20]–[23]. The concepts of AR visualization with wireless devices have been proposed. However, there is no feasible solution to achieve it at the existing visual inspection. For example, researchers deliberately construct marks in visual images, such as 2D barcodes [20], retroreflective or luminous points [21] or other patterns

[23]. To this extent, previous works primarily focused on local interactions with a device with prior knowledge of where it is located in the environment, which limits the scalability.

Wireless Signal-based localization:

i) *Direction of arrival*: In the latest wireless signal localization research [3], [10], researchers have achieved centimeter-level 3D localization accuracy. They both track the device in 3D space by estimating the distance from 3 or more anchor points, and finding the intersections of the circles whose radii are the estimated distances. Although they provide a reliable location estimation method, it cannot meet the requirement of visualizing the target device to the AR device because it cannot estimate the direction of arrival of the received signal.

ii) *Additional hardware requirements*: In order to separate the azimuth and elevation angles from the AoA measurement, the researchers adopt a dedicated antenna array, such as a uniform circular array [24], L-shaped array [25], parallel linear array [16]. To our knowledge, these methods need the support of additional hardware. Scenariot [26] utilizes a combination of UWB distance localization and visual SLAM, which perceives smart devices in the surrounding environment and spatially register them on SLAM-based AR devices.

iii) *Complicated operation*: Park, Y et al. [4] focused on the visualization of IoT devices and conducted design experiments based entirely on wireless communication technology. The proposed azimuth and elevation angles estimation algorithms utilize the phase difference of the received signals from the two antennas, combined with the motion of the AR device tracked by inertial measurement unit (IMU) sensors.

VIII. CONCLUSION

This work presents VisBLE, which provides users with a novel way to interact with Bluetooth devices. VisBLE is the first work to achieve visual tracking with Bluetooth devices under the Bluetooth 5.1 protocol. In contrast with previous wireless signal localization visualization works, VisBLE proposes a novel azimuth and elevation angle estimation mechanism to simplify the user's operation and combines it with computer vision technology to improve the reliability of the system. Our results are divided into the following two aspects: i) Angular and position accuracy: VisBLE positions a BLE device with the median angular error of 3.4° and position error of 8.4cm. ii) Accuracy of tracking: VisBLE identifies a BLE device with an overall success rate of over 90%.

ACKNOWLEDGMENT

This work was supported by China National Key R&D Program 2018YFB2100302, National Natural Science Foundation of China under Grant No. 61902066 and Natural Science Foundation of Jiangsu Province under Grant No. BK20190336 and the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20221-0017), National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme (FCP-SUTD-RG-2021-009).

REFERENCES

[1] H. Chao, "Internet of things and cloud computing for future internet," 2011.
[2] D. D. Ramlowat and B. K. Pattanayak, "Exploring the internet of things (iot) in education: a review," *Information systems design and intelligent applications*, pp. 245–255, 2019.
[3] W. Mao, H. Jian, H. Zheng, Z. Zhang, and L. Qiu, "High-precision acoustic motion tracking: demo." in *International Conference on Mobile Computing & Networking*, 2016.
[4] Y. Park, S. Yun, and K. H. Kim, "When iot met augmented reality: Visualizing the source of the wireless signal in ar view," in *the 17th Annual International Conference*, 2019.
[5] B. Guo, W. Zuo, S. Wang, W. Lyu, Z. Hong, Y. Ding, T. He, and D. Zhang, "Wepos: Weak-supervised indoor positioning with unlabeled wifi for on-demand delivery," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 2, 2022.
[6] D. Cao, R. Liu, H. Li, S. Wang, W. Jiang, and C. X. Lu, "Cross vision-rf gait re-identification with low-cost rgb-d cameras and mmwave radars," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2022.
[7] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.
[8] J. Martin, *The Best Bluetooth Trackers in 2022*, 2022, https://www.techadvisor.com/test-centre/gadget/best-bluetooth-trackers-3674869/.
[9] G. White, C. Cabrera, and A. Palade, "Augmented reality in iot," in *The 8th International Workshop on Context-Aware and IoT Services*, 2018.
[10] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," *Proceedings of SenSys*, 2018.
[11] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.
[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017.
[13] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE International Conference on Robotics and Automation*, 2020.
[14] M. Woolley, "Bluetooth direction finding," *A technical Overview*, 2019.
[15] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, 2004.
[16] L. Gan, J. F. Gu, and P. Wei, "Estimation of 2-d doa for noncircular sources using simultaneous svd technique," *IEEE Antennas and Wireless Propagation Letters*, vol. 7, pp. 385–388, 2008.
[17] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques," *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.
[18] N. Semiconductor, *Bluetooth 5.3 SoC supporting Bluetooth Low Energy*, https://www.nordicsemi.com/products/nrf52810/getstarted.
[19] X. Zhang, W. Wang, X. Xiao, H. Yang, X. Zhang, and T. Jiang, "Peer-to-peer localization for single-antenna devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–25, 2020.
[20] F. Romero-Ramirez, R. Muoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, 2018.
[21] K. Dorfmuller, "Robust tracking for augmented reality using retroreflective markers," *Computers & Graphics*, vol. 23, no. 6, 1999.
[22] Z. Dong, Y. Lu, G. Tong, Y. Shu, S. Wang, and W. Shi, "Watchdog: Real-time vehicle tracking on geo-distributed edge nodes," *ACM Transactions on Internet of Things*, 2022.
[23] S. Wang, J. Lu, B. Guo, and Z. Dong, "Rt-ved: Real-time voi detection on edge nodes with an adaptive model selection framework," *Proceedings of the 28th ACM SIGKDD*, 2022.
[24] Y. Wu and H. C. So, "Simple and accurate two-dimensional angle estimation for a single source with uniform circular array," *IEEE Antennas & Wireless Propagation Letters*, vol. 7, pp. 78–80, 2008.
[25] N. Xi and L. Li, "A computationally efficient subspace algorithm for 2-d doa estimation with l-shaped array," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 971–974, 2014.
[26] H. Ke, Y. Cao, H. Y. Sang, Z. Xu, and K. Ramani, "Scenariot: Spatially mapping smart things within augmented reality scenes," in *Proceedings of the CHI Conference*, 2018.