Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-10166-3

1175: IOT MULTIMEDIA APPLICATIONS AND SERVICES

Image-based positioning system using LED Beacon based on IoT central management

Hyeonwoo An [1] · Nammee Moon [1]

Received: 11 April 2020 / Revised: 12 August 2020 / Accepted: 10 November 2020
Published online: 18 November 2020

Abstract
The benefits of technologies related to the Internet of Things (IoT), virtual and augmented reality (VR/AR), digital twins, and so on, can be fully realized when associated devices are positioned intuitively. However, AR systems hosted within smartphones pose challenges where auxiliary hardware and computational configurations associated with precise positioning are concerned. To this effect, we propose a deep learning-based indoor measurement system that can determine positions using images collected via beacons designed as IoT terminals. The proposed system is broadly divided into a detection unit, an extraction unit, a positioning unit, and a management server. The beacons were detected using deep learning algorithms, from which the postures were extracted using a homography matrix, and position of the imaging device was determined in reference to the beacon's position. With the unique design of our system, in that it simultaneously performs posture and positioning estimations, high immersive AR can be achieved. Moreover, scalability of the positioning space is also guaranteed as multiple beacons can be monitored at once. For the experiment, we simulated a virtual indoor space comprising pyramid beacons and the results were promising.

Keywords Positioning · IoT · Object detection · Deep learning

1 Introduction

Estimating the position of an object is crucial for technologies that coalesce reality and the virtual world, such as Internet of Things (IoT), virtual and augmented reality (VR/AR) applications, and digital twins. Positioning is an important determinant in mapping the real world onto AR systems i.e., projecting the information of real-world objects as is or based on a map of real space [6].

✉ Nammee Moon
nammee.moon@gmail.com

[1] Department of Computer Engineering, Hoseo University, Asan, Republic of Korea

Recently, positioning methods using various technologies such as image-based, sensor-based, fingerprinting, and visible light communication (VLC)-based have been studied [9, 14, 20]. However, mobile devices that are commonly used to host AR applications are neither suited to accommodate the auxiliary hardware required to implement them (like sensors) nor do they harbor appropriate navigation systems that can utilize dead reckoning, for instance, unless the design is customized for positioning. Moreover, even with the in-built positioning options available, such as the global positioning system (GPS) and Wi-Fi fingerprinting, implementing immersive AR is inherently difficult, owing to projection errors i.e., erroneous mapping of real-world objects to the virtual space, thereby rendering a heterogenous AR experience. Researchers have previously explored several approaches to improving positioning in AR systems by conceiving ensemble techniques and refining the source data and other components that likely cause classification errors. However, as these approaches mandate auxiliary sensors and extensive computational setups, they are deemed inadequate for implementing AR in a mobile environment [23].

Here, we propose a system that facilitates transmission of object information and, in turn, estimates its position using images captured via a beacon shaped as an IoT terminal, thereby eliminating the need for additional hardware and software infrastructure. The system initially detects a fixed pyramid beacon using a pre-trained detector and estimates: a) the indoor location of a user through the beacon network, and b) relative location of the device from the beacon image. The positioning is then performed independent of any external sensors other than the camera, and a stable information output is secured over the network from the corresponding beacon.

2 Related works

In this section, we review existing positioning methods and describe the deep learning-based object detection framework integrated into our prototype, including techniques for estimating the location of a captured object.

2.1 Positioning methods

Positioning methods have, in the past, been studied, combined, and utilized in a wide variety of technologies. They are broadly divided into dead reckoning using inertial sensors like acceleration and gyro sensors, Wi-Fi fingerprinting, receiving signal-based, image-based, and VLC-based [9, 14].

2.1.1 Dead reckoning

Dead reckoning is primarily used in environments where it is difficult for the GPS to receive a signal to assist with positioning. The current position is evaluated by accumulating the directions and accelerations of each cycle using an inertial navigation device, such as an acceleration, gyro or geomagnetic sensor, all of which have recently been incorporated into mobile devices, thereby negating the need for auxiliary hardware components [9]. A drawback of this method is that the longer the positioning time, the higher the accumulated error of the sensor [1]. Furthermore, considering the data is recorded as a time-series sequence in the object's active state, it is often paired with other positioning methods that provide a reference point.

2.1.2 Receiving signal-based positioning

GPS is a popular receiving signal-based system which measures the distance between signals received from three or more satellites and consequently apprehends the exact location by applying triangulation. Although this method can be implemented easily using inexpensive GPS sensors coupled with a computing device for measuring distances, it has a relatively high error rate owing to the interferences along the propagation path, clock errors of the satellites and measuring device, and defective internal circuits.

Wi-Fi fingerprinting maps the Wi-Fi signal strengths across multiple access points onto a virtual indoor space and then intercepts the location by measuring the signal strength pattern as recorded on the generated map. Similar to GPS, commercial mobile devices are supplemented with the hardware required by this method, however, due to the ensuing battery management issues in that the number of signal measurement cycles increases proportionately with time, real-time positioning is impossible; moreover, the signal strength, which can be compromised by obstructions and environmental changes between the AP(Access Point) and positioning device, directly affects the positioning. Recent studies have reported increased execution speed and accuracy using random forest and other improved matching algorithms with this method [7, 10].

2.1.3 Image pattern matching-based positioning

This method is used to extract a pattern from the properties of an image to determine the space from which it is taken. There are three ways in which the pattern can be elicited: a) marker of a preset pattern, b) comparison between inherent patterns such as the histogram properties of the image, and c) comparing an image pattern captured in a mapped virtual space. Recently, the final technique has evinced high accuracy given the advances in deep learning technologies and superior camera configuration in smartphones that support capturing high quality images. Zhang et al. [21] reported favorable results in recreating a captured image in a 3D-mapped space with high precision using various sensors.

2.1.4 VLC-based positioning

This method harnesses VLC technologies that rapidly communicate data using signals in the visible light spectrum. A light-emitting diode (LED) is initially assigned an independent frequency that acts as its ID, then the coordinates of each LED in the network are mapped, and finally, a positioning device is used to retrieve pertinent information. The relative position of the imaging device is measured by triangulating the coordinates of three or more LEDs [8]. This method has exhibited high accuracy of less than 10 cm in error averages. The object position can be computed in the following three ways: grid units, 3D coordinates (x, y, z), and measuring the altitude of a device via multiple LEDs [9, 11, 13].

2.2 Deep learning-based object detection system

In our proposed methodology, an object and its bounding box are both detected from a captured image using an artificial neural network. This technology can prove to be beneficial in several domains like assessing the traffic flow based on CCTV images [19]. Depending on the quantity and quality of learning data, the probability of false positives can be minimized to

achieve high accuracy even under unfavorable conditions as opposed to the algorithm-based object detection method.

Studies have shown significant progress in utilizing deep learning models for object detection, including region with convolutional neural networks (R-CNN) derived from CNN, which is popularly used to classify images [4]. R-CNN accomplishes object detection using either a 1-stage or 2-stage detection model to perform two distinct tasks viz. a localization task to detect the bounding box of an object and a classification task to classify the type of object detected.

2.2.1 2-stage detection model

This detection model first identifies bounding boxes via techniques like sliding windows, selective-search, and region proposal networks (RPN) and then classifies the habiting objects separately. In general, the two-step detection method obtains high accuracy but is slower as the classification is performed separately for each bounding box.

Faster R-CNN, GoogLeNet, and Region-based Fully Convolutional Networks (RFCN) are types of 2-stage detection models [3, 16, 18].

2.2.2 1-stage detection model

This type of model detects objects using grids. A backbone network helps to create the grid concept by removing a few layers from the last segment of the model trained on a CNN network that oversees the classification task. Thus, the backbone network effectively generates a feature map of low-resolution images as the final output. For instance, if the final layer is $7 \times 7 \times 512$ in size, a feature map is obtained by dividing the input image into a 7×7 grid. The positioning and classification within a grid unit is performed using CNN hence, each grid of the feature map has a value mapped to a bounding box. Although this method has the advantage of high speed, it has reportedly furnished low accuracy in past studies.

Therefore, for optimized object detection, the 1-stage model is adopted for tasks that involve quick turnaround time while 2-stage model is utilized for those warranting high accuracy. Refinedet, Single Shot Detector (SSD) series, and You Only Look Once v3 (YOLOv3) are types of 1-stage detection models [5, 15, 22].

3 Methods

3.1 System overview

Proposed hereunder is a positioning system constituting a beacon that is designed to not only detect images but also centrally manage IoT. As illustrated in Fig. 1, when a user shoots the beacon using their smartphone camera, the posture and position of the device can be computed from the captured image.

The system is divided into five fundamental components viz. a pyramid beacon, management server, recognition unit, detection unit, and positioning unit. The functionality of each of these components is shown in Fig. 2.

Beacon is a basic element of positioning and comprises, as shown in Fig. 3, a wireless communication module and microcontroller for connecting to the IoT network.

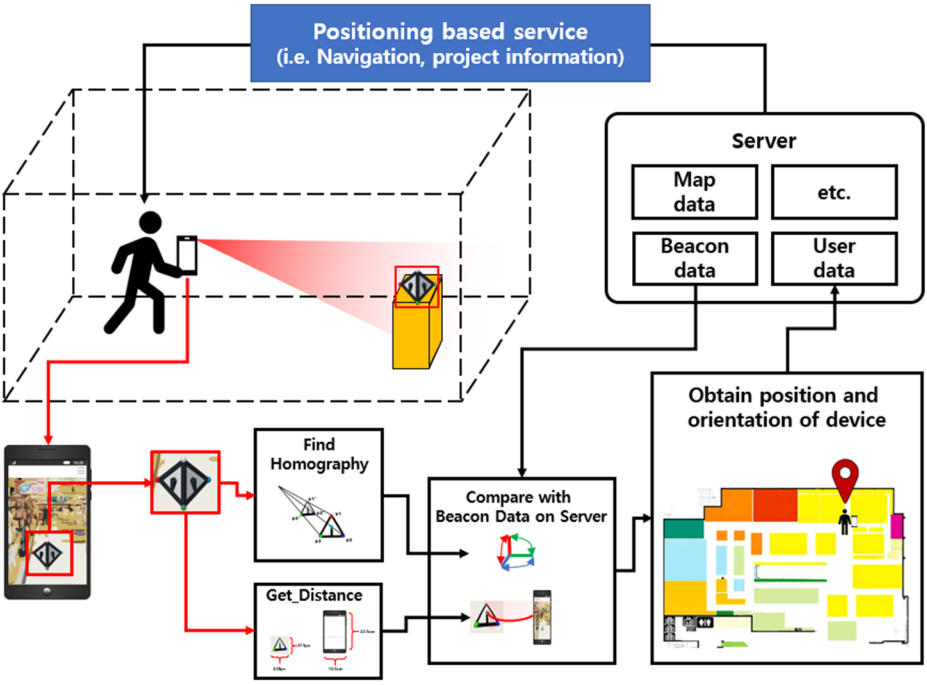

Fig. 1 Pyramid beacon-based indoor positioning system

3.2 Beacon detection

The YOLOv3 model (one-step detector) was incorporated into the detection unit for obtaining the bounding box, as it required quick turnaround time to estimate the altitude and distance of the beacon from the extraction unit. To seamlessly train the model, we also developed an image generator that churned a set of images necessary for learning.

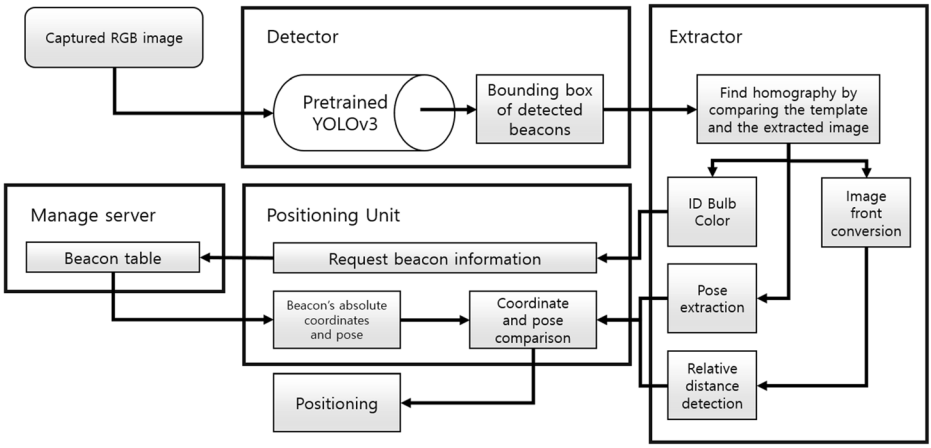

Fig. 2 Flow chart of the role of components

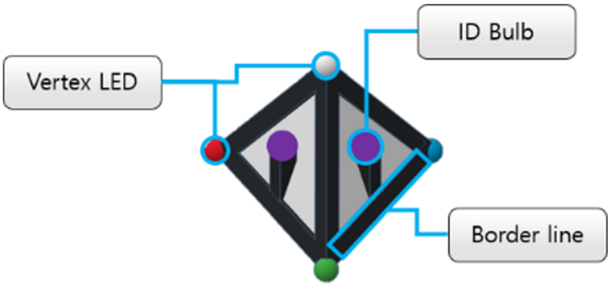

Fig. 3 Configuration of beacon

For our experiment, the model was trained to detect only one class i.e., the pyramid beacon, and each parameter set for the existing 80 classes was configured as shown in Table 1.

The image dataset for this study comprised 11,474 photos of VOC dataset for the background as well as randomly inserted the beacon images in various scales. However, the test responded poorly to changes in illuminance and rotation, hence their randomly changing applied too. Training was conducted with a total of 9179 images while 2295 images were set aside for verification. An average loss of 0.08 was accrued as a result of learning with the above parameters.

3.3 Indoor positioning mechanism

Positioning is computed using any one side of the captured beacon area. This is typically done by comparing the beacon model of a 3D object to the one captured in a 2D plane. Given that in our study the shooting angles wherein all four vertices could be captured were limited and thus it was impossible to compare invisible vertices, we performed the comparison using homography on the vertices face-to-face to elicit posture measurements [2, 5].

A homography matrix (H) contains rotation matrices and can be used to derive the Euler angles of rotation [2, 5]. It is obtained by comparing four or more feature points. Figure 4 highlights how the position of the camera is derived using a homography matrix. Plane P is initially extracted from the captured beacon. P constitutes three vertices and each vertex is then sorted in a specific order for comparison against a template using H. If the three vertices of P are assumed to be colored in either of white, green, or blue (wgb), then H is obtained by equating the face of the front face with the wgb color composition, through which the device position is then obtained.

Table 1 Learning parameters

Parameters	Values
classes	1
learning_rate	0.001
scales	.1,.1
batch	64
subdivisions	32
max_batches	6000
filters	18

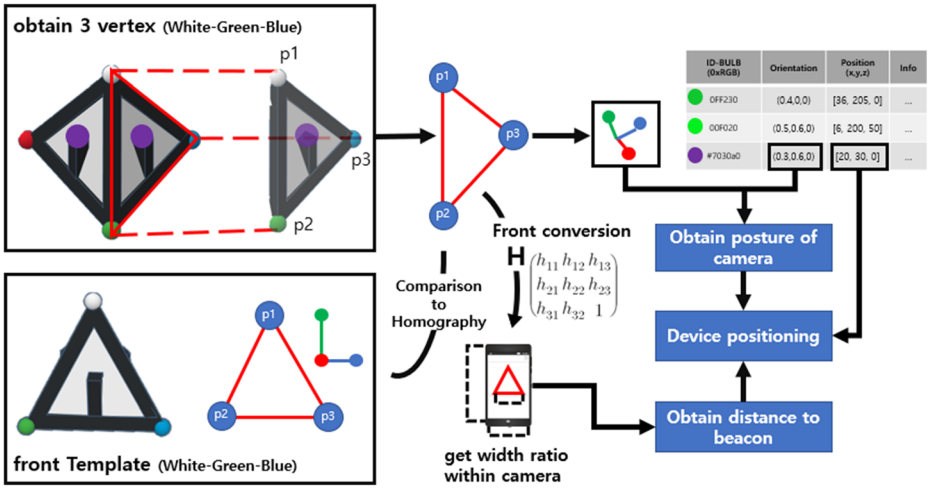

Fig. 4 Indoor positioning mechanism

The posture of an object from the detected image is determined by comparison against a reference, which in this paper, was a planar object that helped to identify rotational changes about the three axes (x, y, z). The changes along z-axis, y-axis, and x-axis, respectively, are given by:

$$R_z(\alpha) = \begin{bmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cosa & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix}, \mathrm{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \quad (1)$$

Therefore, the sum of rotational changes about all three axes, R, can be expressed as [17]:

$$\begin{aligned} R &= R_z(\alpha)\, R_y(\beta) R_x(\gamma) \\ &= \begin{bmatrix} cos\beta cos\alpha & sin\gamma sin\beta cos\alpha - cos\gamma sin\alpha & cos\gamma sin\beta cos\alpha + sin\gamma sin\alpha \\ cos\beta sin\alpha & sin\gamma sin\beta sin\alpha + cos\gamma cos\alpha & cos\gamma sin\beta sin\alpha - sin\gamma cos\alpha \\ -sin\beta & sin\gamma cos\beta & cos\gamma cos\beta \end{bmatrix} \end{aligned} \quad (2)$$

Figure 5 describes the process of deriving H and the Euler angles from both the beacon's bounding box and the detected image. Features of the detected image and corresponding LED coordinates were elicited for comparison and thereafter the homography and rotation matrices were computed from the comparison point.

H is a 3 × 3 matrix that extrapolates the transformation relationship of a 3D plane when projected into a different state. If the z-axis of the target 3D plane equals 0, then the H that maps it onto $M = (X,Y,0)^T$ for P=K[R|t], and a corresponding 2D plane, m, is given by:

$$\begin{aligned} \tilde{m} &= K\left[R^1\ R^2\ R^3\ t\right] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \\ &= \left[R^1\ R^2\ t\right] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ H &= K\left[R^1\ R^2\ t\right] \end{aligned} \quad (3)$$

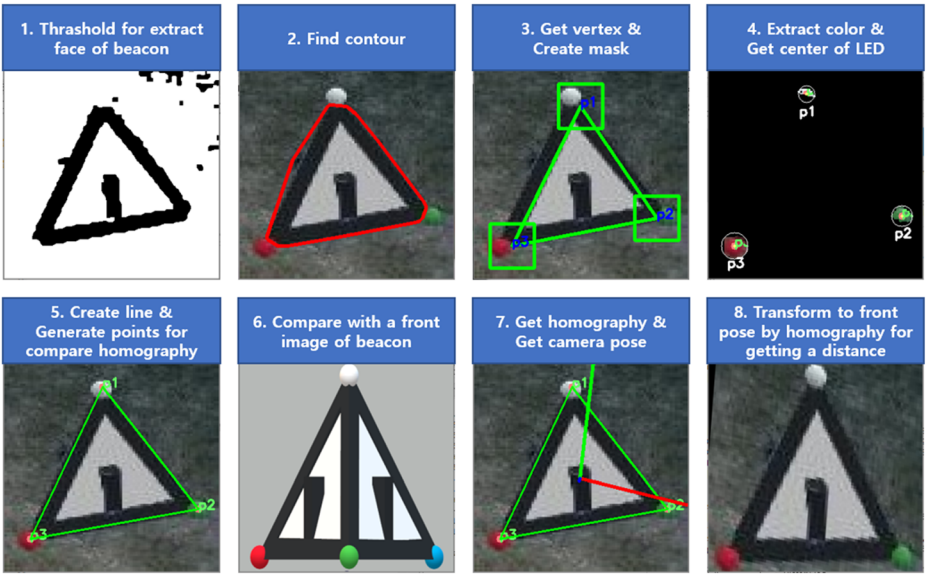

Fig. 5 H matrix acquisition and Euler angle acquisition process

where R denotes the rotation matrix, t denotes the movement matrix, and K denotes the camera matrix with a focal length, principal point, and skew coefficient.

Furthermore, for a given value of K, we can decompose H to determine R as [12]:

$$\begin{aligned} R^1 &= l_1 * K^{-1} H^1 \\ R^2 &= l_2 * K^{-1} H^2 \\ R^3 &= R^1 \oplus R^2 \\ t &= l_3 * K^{-1} H^3 \end{aligned} \tag{4}$$

where (l_1, l_2, l_3) are the scaling factors given by [12]:

$$\begin{aligned} l_1 &= \frac{1}{norm\left(K^{-1}H^1\right)} \\ l_2 &= \frac{1}{norm\left(K^{-1}H^2\right)} \\ l_3 &= \frac{l_1 + l_2}{2} \end{aligned} \tag{5}$$

The Euler angles about the posture of the device can then be calculated from R by applying a conversion function.

To evaluate the relative relationship between the beacon and device, the indoor position and posture information of the beacon need to be referenced. We obtained this by first extracting the color of the beacon from the detected area using its bulb ID and then sending it as a query in a predetermined format to the beacon server. Considering all beacons in the system were molded in a uniform shape i.e., tetrahedron, and size, we can assume that each face is always configured in the same posture.

Let the posture of plane P be P_pose, the beacon posture in the indoor coordinate system be B_pose, then the posture of the beacon for a given reference plane in the indoor coordinate system of P, In_pose, is given by (P_pose + B_pose). Furthermore, to measure the position of the camera, P2C_pose, as it aligns with P using the relative distance, we extract the altitude of the camera facing P, C_pose, then compute it as (In_pose – C_pose).

Finally, the indoor coordinates of the camera device are obtained using the relative distance and P2C_pose values. Therefore, for a given set of indoor coordinates of x', y', and z 'of a beacon, the position is determined solely by deputizing the pitch and yaw values (excluding the roll value) in the below formula:

$$\begin{aligned} x &= x' + distance*\sin(yaw)*\cos(pitch) \\ y &= y' + distance*sin(pitch) \\ z &= z' + distance*\cos(yaw)*\cos(pitch) \end{aligned} \tag{6}$$

3.4 Simulation

An experiment was conducted to: a) validate the performance of the model, b) determine the shooting range, and c) evaluate the accuracy of the detected posture and final position. A virtual environment was simulated in Unity with a beacon installed horizontally on a 100 cm high cradle. Virtual environments are advantageous in that the locations and postures of objects can be meticulously designed and manipulated as opposed to the real world and they are immune to changes in illumination and other ambient environmental settings.

To understand how changes in the distance and shooting angles affected the system performance, we conducted tests with 540 instances of images taken at different distances and angles. Additionally, errors occurring in the detection unit were excluded as they adversely affected the positioning.

For the simulation, a rear wide-angle camera built into Galaxy S10 was administered; the development environment was OpenCV 3.4.7, which is a digital image processing framework, and Python was used for data analysis.

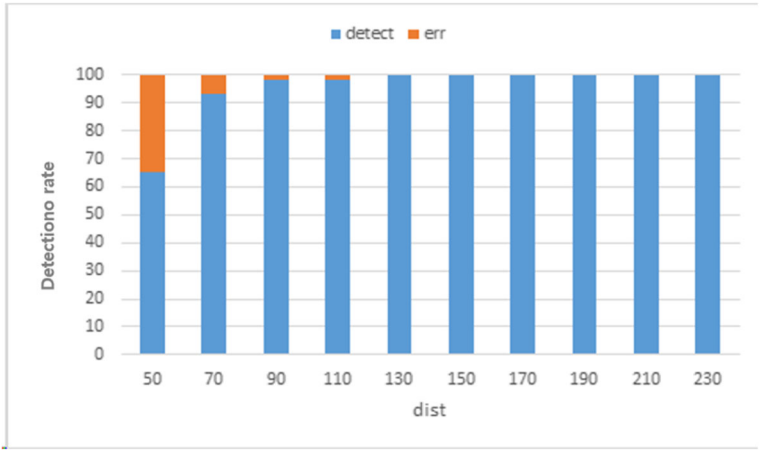

Fig. 6 Change of undetected rate according to the shooting distance

Table 2 Relative distance estimation accuracy test result table

	DISTANCE (Unit: cm)									
	50	**70**	**90**	**110**	**130**	**150**	**170**	**190**	**210**	**230**
samples	34	56	43	59	60	48	48	48	48	48
err	4	8	2	3	3	1	0	0	0	1
Acc(%)	79.94	83.56	81.09	81.57	82.88	80.44	77.64	77.07	79.85	79.88
MAE	10.03	11.51	17.02	20.28	22.25	29.35	38.02	43.57	42.32	46.27

4 Results and discussion

Results revealed that the sensitivity of the beacon decreased with distance. As highlighted in Figs. 6, 21 beacons at 50 cm, 4 beacons at 70 cm, and 1 beacon at 90 and 110 cm were not detected.

Furthermore, the confidence score, which is used to delineate the bounding box, remained consistently above 90% beyond a certain range.

Table 2 enumerates the results of the relative distances obtained in our experiement. For a given camera angle, the relative distance was computed for a given number of samples (denoted by "samples") i.e., object and pertinent image data passed to the detection unit, which was then compared with the actual values to derive the mean average error (MAE). Accuracies ("acc") of the derived distances were also recorded for performance validation.

Table 3 below highlights the error of estimated position of the camera about the three axes for a given shooting range and LED color composition. It was observed that the accuracy decreased proportionally with distance.

Lastly, Table 4 presents a comparative study of the proposed system against the positioning techniques explored in section 2. For the existing methods, we referenced studies that reported superior results to elicit more accurate comparisons.

Table 3 Final positioning error test result table

	Mean Error (Unit:cm)			samples
	x_err	**y_err**	**z_err**	
color composition				
wrb	69.33	25.29	5.63	167
wgb	44.06	24.42	68.44	183
wrg	30.77	24.62	50.02	120
distance				
50 cm	15.4	12.01	11.92	30
70 cm	23.7	13.03	20.86	48
90 cm	37.53	16.65	26.02	41
110 cm	36.82	20.72	28.07	56
130 cm	42.41	27.1	35.86	57
150 cm	53.87	26.45	40.67	47
170 cm	59.82	27.95	50.32	48
190 cm	68.77	30.68	54.86	48
210 cm	75.09	31.73	60.34	48
230 cm	72.49	36.04	75.92	47

Table 4 Comparison table of proposed system and related studies

	Proposed Method	Dead Reckoning [1]	Signal based [10]	VLC based [11]
Application technology	Deep learning (YOLO v3), Image processing	step length estimation algorithm(using Neural network)	Deep learning(Random Forest), WiFi Finger Printing	VLC, Image processing, PnP model
Measure dimension	3D position in space	Specific reference distance	2D Grid in space (2 m interval)	3D position in space
Required specifications	Camera device, LED Beacon	Devices with inertial system	WiFi Sensor device, 5 or more APs	Camera device, 4 or more LEDs
Limitations and disadvantages	Obstacle, recognition distance, LED Beacon production required.	Accumulation of errors, positioning impossible for position and direction	Obstacles, early radio map construction, reduced practicality due to smartphone WiFi policy changes	Obstructions, limited shooting distance (4 or more LEDs must be taken at once)
Experiment environment	Check at 20 cm intervals in a circular space of approximately 5 m in diameter	Fixed track of 1 km	total of 10 square shaped cells with 2 m gap. Measured for about 90 s per cell	1x1x0.8 m room, 10 s recording at each point
Experiment result	80.53% accuracy for each point	10 m error when walking 1 km	97.5% accuracy of cell	MAE 4.81 cm

5 Conclusions

We proposed a novel indoor positioning system with a smartphone camera and pyramid beacon as its main components. YOLOv3, a one-step R-CNN-based detector, was utilized for object detection that helped to overcome limitations like erroneous detection for varied shooting ranges, rotation, and illuminance of the existing image pattern-based and marker-based positioning techniques.

While the system exhibited satisfactory results for positioning (80.53% accuracy on an average), it had a few drawbacks. Firstly, if the path between the beacon and camera is obstructed in any manner, then the extraction cannot be performed. Secondly, in rarer occasions, it was observed that the pitch value was inverted and, in turn, the output, during posture extraction. These issues can occur in systems that make point-to-point comparisons to estimate the object position– by performing exception processing using the characteristics of the grounded beacon, the error rate can be reduced.

Acknowledgments This work is supported by the National Research Foundation of Korea (NRF) and the grant was funded by the Korean Government (MSIT, No. NRF-2017R1A2B4008886).

We would like to thank Editage (www.editage.co.kr) for English language editing.

References

1. Beauregard S, Haas H (2006) Pedestrian dead reckoning: a basis for personal positioning. In: proceedings of the 3rd workshop on positioning, navigation and communication. Pp 27–35
2. Chum O, Pajdla T, Sturm P (2005) The geometric error for homographies. Comput Vis Image Underst 97: 86–102
3. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: Lee DD, Sugiyama M, Luxburg UV, et al (eds) Advances in neural information processing systems 29. Curran Associates, Inc., pp. 379–387
4. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587
5. Ha H, Rameau F, Kweon IS (2016) 6-DOF direct Homography tracking with extended Kalman filter. In: Image and Video Technology. Springer International Publishing, pp. 447–460
6. Jo D, Kim GJ (2019) IoT+ AR: pervasive and augmented environments for "Digi-log" shopping experience. Human-centric Computing and Information Sciences 9:1–17
7. Khongkraphan K (2019) An efficient fingerprint matching by multiple reference points. Journal of Information Processing Systems 15:
8. Komine T, Nakagawa M (2004) Fundamental analysis for visible-light communication system using LED lights. IEEE Trans Consum Electron 50:100–107
9. Lee SW, Kim SW (2015) Indoor positioning technology trends and outlook. The Journal of the korean institute of communication sciences 32:81–88
10. Lee S, Kim J, Moon N (2019) Random forest and WiFi fingerprint-based indoor location recognition system using smart watch. Human-centric Computing and Information Sciences 9:6
11. Li Y, Ghassemlooy Z, Tang X, Lin B, Zhang Y (2018) A VLC smartphone camera based indoor positioning system. IEEE Photon Technol Lett 30:1171–1174
12. Lu C, Hager GD, Mjolsness E (2000) Fast and globally convergent pose estimation from video images. IEEE Trans Pattern Anal Mach Intell 22:610–622
13. Luo P, Zhang M, Zhang X, et al (2013) An indoor visible light communication positioning system using dual-tone multi-frequency technique. In: 2013 2nd international workshop on optical wireless communications (IWOW). Pp 25–29
14. Mautz R, Tilch S (2011) Survey of optical indoor positioning systems. In: 2011 international conference on indoor positioning and indoor navigation. Pp 1–7

15. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv [cs.CV]
16. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39:1137–1149
17. Slabaugh GG (1999) Computing Euler angles from a rotation matrix. Retrieved on August
18. Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9
19. Wang X-X, Zhao X-M, Shen Y (2019) A Video Traffic Flow Detection System Based on Machine Vision Journal of Information Processing Systems 15:
20. Werner M, Kessel M, Marouane C (2011) Indoor positioning using smartphone camera. In: 2011 international conference on indoor positioning and indoor navigation. Pp 1–6
21. Zhang X, Rad AB, Wong Y-K (2012) Sensor fusion of monocular cameras and laser rangefinders for line-based simultaneous localization and mapping (SLAM) tasks in autonomous mobile robots. Sensors 12:429–452
22. Zhang S, Wen L, Bian X, et al (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4203–4212
23. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30:3212–3232