

Design of a VLC-based Beaconing Infrastructure for Indoor Localization Applications

Gergely Zachár, Gergely Vakulya, Gyula Simon

University of Pannonia

Department of Computer Science and Systems Technology

Veszprém, Hungary

Email: {zachar,vakulya,simon}@dcs.uni-pannon.hu

Abstract—In this paper the design of an LED-based beaconing infrastructure is introduced, which can be utilized in indoor localization systems. The LED beacons, which may be part of the existing lighting infrastructure, are blinking with high frequency invisible for human eyes, while the detectors are regular cameras. Since the sampling frequency of the cameras is much lower than the blinking frequency, the detection is based on a coding system, which tolerates the undersampling of the signal. The performance of the proposed system is analyzed and a possible application is introduced.

Index Terms—indoor localization, Visible Light Communication, beaconing, detection, trilateration

I. INTRODUCTION

Indoor localization, where GPS signals are not available, is a challenging and widely researched area. Dead reckoning techniques utilize only inertial sensors and possibly the map of the building, and can provide good accuracy for short distances; but the accumulating error in long distances must be handled by other means [1]. Where the accuracy demand is not too high (i.e. in the range of a few meters) most of today's systems utilize radio (WiFi or Bluetooth) signal strength measurements [2], [1]. Time of flight measurements, utilizing ultrasound or ultra-wideband radio signals, allow ranging and thus localization can be performed using trilateration [3], [4], allowing higher precision (approx. 10 cm).

Light based approaches utilize LEDs as anchors, which can be installed as a part of the lighting infrastructure. The ranging is performed either by light intensity measurements [5] or using image processing techniques [6]. In these systems visible light communication (VLC) is utilized to transmit the identifier of beacons.

Most of the camera based decoder systems utilize the rolling shutter effect: since the image sensor samples different rows of pixels at different time instants, high frequency blinking appears in the form of alternating dark and light bands in the picture, which is utilized to decode the transmitted code [6]. The main advantage of these systems is that one image frame is enough for decoding, but they require large image size for the correct operation, thus the beacon must be close to the camera.

In this paper another approach is introduced, where the detection is performed by utilizing a series of pictures, allowing very small image sizes, and thus distant beacons can also be reliably detected.

In Section II the proposed beacon and detector are introduced with design considerations. Section III provides design considerations and performance analysis. In Section IV a possible application scenario is presented. Section V concludes the paper.

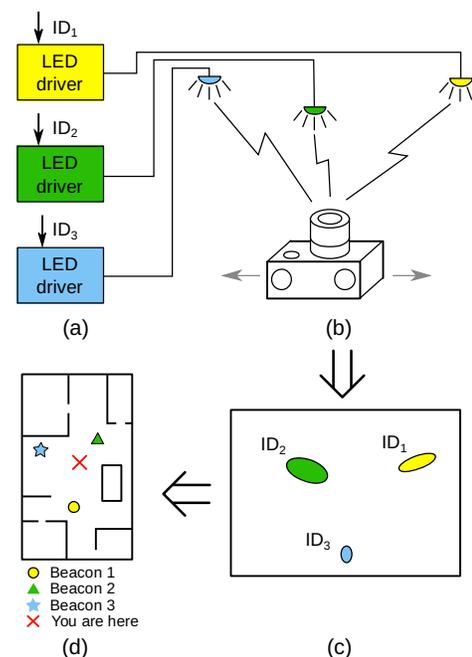


Fig. 1: The proposed system architecture. (a) Modulated LED beacons. (b) Wide angle camera. (c) Beacon detection (d) Position estimation.

II. PROPOSED SYSTEM

The block diagram of the proposed system is shown in Figure 1. The emitted light of each LED is modulated by a LED driver and a unique beacon ID is transmitted by each LED. The modulation utilizes high frequency, providing flicker-free operation. In the detector a camera is utilized, which takes pictures with approximately 30 fps. The series of images are used to detect pixels, which correspond to any of the transmitted beacon IDs. The detected pixels are then used to find blobs on the image, each blob corresponding to one beacon. The localization system then estimates the unknown

location from the detection image and the a priori information on the beacon positions. Notice that the last step (localization) is not in the scope of this paper, but a case study application will be presented to illustrate the achievable accuracy.

A. Beaconing

The beacon ID is encoded as a continuously repeated bit pattern. The codes begin with a header (11100), which is not allowed in any other position of the ID's bit pattern. In the tested system the length of the code (including header) was $N_c = 13$ bits.

The channel encoding is illustrated in Figure 2(a). The encoder generates the on/off sequence and power level settings for the LEDs as follows: during the *one* bits the LED is driven at full power and blinked with 50% duty cycle, while the *zero* bits are encoded as constant (not blinking) light with half brightness. The applied encoding scheme has the advantage that human eyes cannot see the modulation and flicker-free operation can be provided, if the modulation frequency is sufficiently high. The finite sampling (shutter) time of the camera somewhat limits the maximum usable frequency, but most cameras allow modulation frequencies much higher than 100 Hz. In the proposed system $f_b = 165$ Hz was utilized. The design considerations related to the modulation frequency f_b and the lengths of the pulses, coding the bits, will be discussed in Sections II/E and II/F.

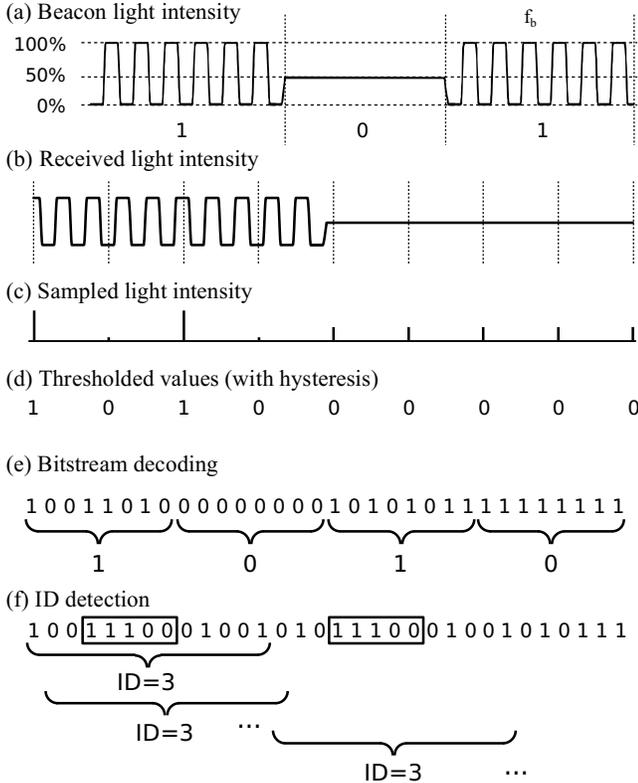


Fig. 2: Coding and decoding of the beacon IDs.

B. Bit detection

The LED beacons are observed with a camera, as shown in Figure 1. Although regular cameras cannot follow high speed light intensity changes, with proper settings suitable undersampling can be provided. With high speed shutter mode (where the shutter time $T_s \ll 1/f_b$) the camera takes short samples from each frame, as shown in Figure 2(b) and (c), thus blinking light provides frame sequences where the corresponding pixels show changes in intensity values. E.g. with $f_b = 165$ Hz a camera with 30 FPS sampling frequency provides 15 Hz alias frequency.

The detector utilizes a comparator with hysteresis to convert each pixel of the detected image frame to a binary value, as shown in Figure 2(d). A state machine is operated for each pixel: it examines whether the last K value of the pixel contains changes or not, and provides the length of the last observed blinking segment T^{D1} (which is the length of a bit 1 or a series of bits 1), or the length of the last observed static segment T^{D0} (which is the length of a bit 0 or a series of bits 0). From the output of the state machine the bit values are identified and thus the bit stream is decoded, as shown in Figure 2(e). For the detection the designed symbol length values are used, see later in Table I. To determine the number of consecutive bits, the designed signal length values (see in Table I) and the corresponding T^{D1} and T^{D0} values are used (see Figure 4)

C. ID detection

The code detection algorithm continuously monitors whether the last N_c detected bits correspond to any of the beacon IDs, where N_c is the length of the transmitted code (including the header), as shown in Figure 2(f). The search is performed in parallel, using a code table T_{CODE} with 2^{N_c} entries.

The index of a code table entry corresponds to a possible detected bit sequence, while the table contains the corresponding ID (or an invalid symbol). Thus any possible rotation of a code can be detected after the reception of N_c bits, there is no need to wait for the header before decoding.

The detection is performed as follows. For each valid beacon ID a detection image DIM_{id} is stored, initially each pixels being unmarked. $N_{frame} > N_c$ consecutive frames are processed, and in each frame for every pixel, using the last N_c detections Ω_{N_c} , $id = T_{CODE}(\Omega_{N_c})$ is determined: if id is a valid ID, then the corresponding pixel is marked in detection image DIM_{id} .

After N_{frame} samples the detection images are further processed to calculate the beacon positions. In the prototype system N_{frame} was set to 250 frames.

D. Beacon detection

In the detection image DIM_{id} those pixels are marked, where beacon id was detected. First connected areas (blobs) are detected, then the centroid of each blob is calculated, along with the size of the blob.

Blobs with size smaller than B_{min} are ignored, the others are used in the localization algorithm. The minimal blob size in the proposed system were set to $B_{min} = 7$, based on the analysis presented in Section III.

E. Code parameter design

Symbol length is one of the key parameters of a data transmission system. To examine the effect of different symbol sizes, a simulator was used. The simulator can virtually emit and observe an arbitrary lighting pattern, with tunable parameters (blinking frequency and camera FPS).

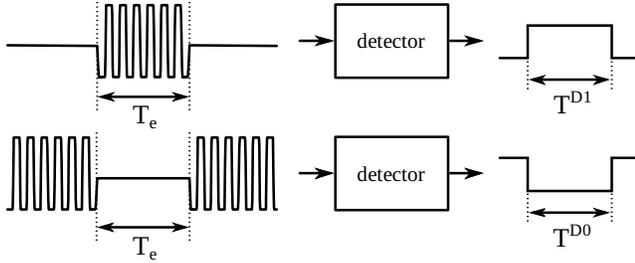


Fig. 3: Scheme of the T^{D1} vs. T_e (upper part) and T^{D0} vs. T_e (lower part) measurements.

The test setup is shown in Fig. 3. Test signal #1 contains a blinking pattern with length T_e , preceded and followed by 500 ms constant signal segments. Similarly, test signal #2 contains a constant signal segment with length T_e , preceded and followed by 500 ms blinking signal segments. Test signals are also shown in Figure 3.

In the tests T_e was varied between 30 ms and 1500 ms, and for each length, 1000 different beacon-camera time offsets and 15 different camera FPS setups (between 29 Hz and 31 Hz) were used.

Figure 4 shows the minimal (red dots) and the maximal (blue dots) observed pulse length values for test signal #1 (T^{D1} , Figure 4(a)) and for test signal #2 (T^{D0} , Figure 4(b)).

The minimal usable symbol length T_1^{Dx} is the shortest pulse, which is always detected, i.e.:

$$0 < \min(T^{D1}(T_1^1)) \quad (1)$$

$$0 < \min(T^{D0}(T_1^0)) \quad (2)$$

The shortest usable length T_2^{Dx} for two consecutive symbols must satisfy the following constraints:

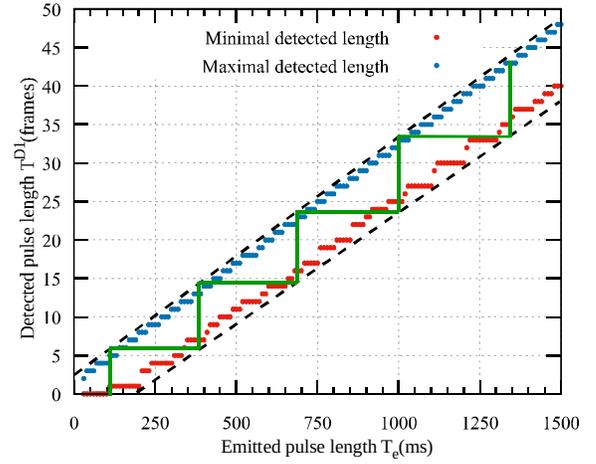
$$\max(T^{D1}(T_1^1)) < \min(T^{D1}(T_2^1)) \quad (3)$$

$$\max(T^{D0}(T_1^0)) < \min(T^{D0}(T_2^0)) \quad (4)$$

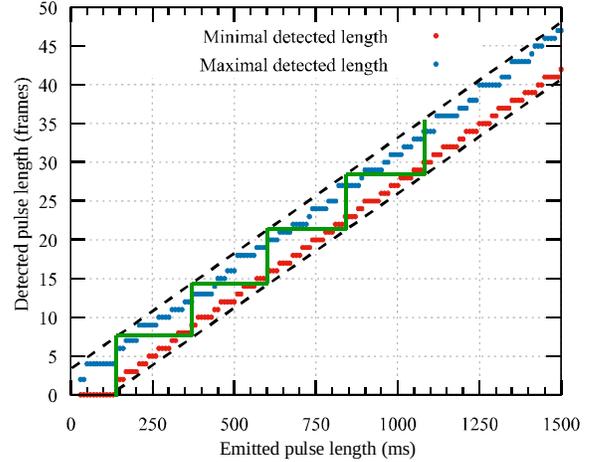
Similarly, for any consecutive k symbols the following inequality must be satisfied:

$$\max(T^{D1}) < \min(T^{D1}(T_{k+1}^1)) \quad (5)$$

$$\max(T^{D0}) < \min(T^{D0}(T_{k+1}^0)) \quad (6)$$



(a)



(b)

Fig. 4: Detected length of (a) emitted blinking signals and (b) constant intensity signals. Red dots: minimal detected length, blue dots: maximal detected length.

To simplify the design and to provide a safety margin, calculations were made using the linearized upper and lower bounds for T^{D1} and T^{D0} (Figure 4(a) and (b), dashed lines). The design procedure is symbolized with green line segments (Fig. 4(a) and (b)), implementing equations (1) – (6). The design results for symbol lengths of 1 ... 5 consecutive 0 and 1 bits are summarized in Table I.

TABLE I: Designed minimal impulse length values for 1 ... 5 consecutive 0 or 1 bits (in ms).

| k | 1 | 2 | 3 | 4 | 5 |
|---------|-----|-----|-----|------|------|
| T_k^0 | 140 | 370 | 600 | 845 | 1080 |
| T_k^1 | 110 | 380 | 690 | 1000 | 1340 |

F. Error rate vs. blinking frequency

The effect of using different blinking frequencies were demonstrated using the simulator described in Section II/E.

Test signal #1 was used, where the blinking frequency was varied between 150 Hz and 180 Hz and three different T_e values (200 ms, 400 ms and 600 ms) were used. The camera FPS was set to 30 Hz. In each test run 1000 different beacon-camera time offsets were simulated. A test run was considered to be successful if exactly one continuous blinking signal (i.e. a series of 1 bits) was observed, other cases (e.g. no detection or incorrect detections of 0 bits in between) were considered erroneous. For each blinking frequency the number of successful and erroneous detections were recorded. Figure 5 shows the error rate vs. the blinking frequency for different pulse lengths (200 ms, 400 ms and 600 ms).

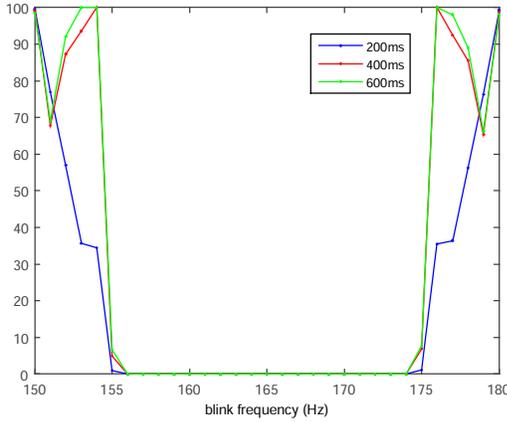


Fig. 5: Error rate of detection with 30 FPS camera vs. blinking frequency f_b , for impulse lengths of 200 ms (blue), 400 ms (red) and 600 ms (green).

As Figure 5 clearly shows, blinking frequencies between 156 Hz and 174 Hz (generating a 6-15 Hz alias frequency with the 30 FPS camera), give perfect results. The error rate dramatically increases below 155 Hz and above 175 Hz, where the alias frequency gets closer to the 0 Hz beat.

G. Implementation

The proposed design was validated by proof of concept system, shown in Figure 6. Each beacon contained a 10 Watt power LED, an LM3414 driver, providing constant current with good efficiency, and a pattern generator using an Atmega128RFA1-based sensor network node running TinyOS. The communication capability of the driver was used to implement remote control on the devices, but was not used in the beaconing process itself.

The detection algorithm was implemented as a multi-threaded C program. The images were split into slices, each processed by a separate thread. The image loading and processing subtasks were synchronized with a mutex-based barrier method.

In the tests an Intel i5-3210M CPU was used with 4 cores, where 3 cores were reserved for image processing tasks. The implemented detector was able to process the 30 FPS 1080p video stream in real time.



Fig. 6: The photo of the pattern generator, LED drivers (in plastic enclosure), and the power LED.

III. PERFORMANCE EVALUATION

A. Detection accuracy

The detection process may result in two types of error: a visible beacon may not be detected (false negative), or a beacon not present may be detected (false positive). To provide immunity against small-sized false positives, a minimal blob size B_{min} is defined: detections with size smaller than B_{min} are ignored.

To determine the optimal value of B_{min} , a test was performed, where 500 measurements were recorded using 4 beacons, placed at various positions, and 16 IDs (including the 4 utilized ones) were used in the detector. Figure 7 shows the frequency of false positives and false negatives (straight line and dotted line, respectively) vs. blob size threshold B_{min} . According to the measurements the optimal value of B_{min} is around 7, where the rates of both the false positive and false negative detections are around 5%.

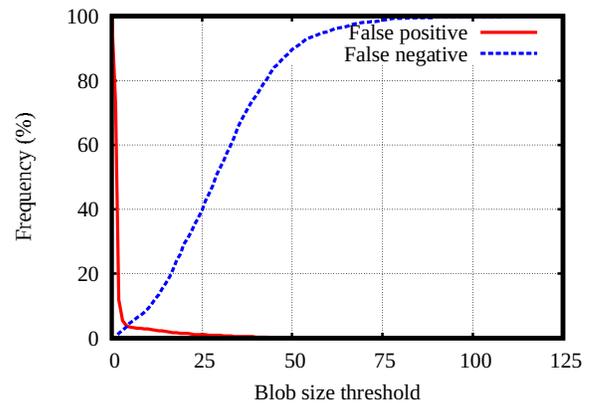


Fig. 7: Probability of false positives and false negatives.

B. Cluster size vs. distance

As results of Section III/A show, the quality of the detection highly depends on the blob size. To characterize the relationship between the detected blob size and the beacon-camera

distance, a measurement was performed. In the experiment the beacon-camera distance was varied between 5 and 50 meters and in each position 50 measurements was made. To model realistic scenarios, the camera and the LED was not facing each other, but rather each device pointed 45 degrees into the camera-beacon axis. In the tests 3 configurations were tested, with different average LED power values (5W, 2.5W, and 1W).

For each measured distance the average, minimum, and maximum detected blob sizes are shown in Figure 8. The detected blob sizes increase as the power increases. Using $B_{min} = 7$, the maximum distance for each power value can be determined from Figure 8: for 1W and 2.5W the maximum distance was around 25m, while for 5W it was more than 30m.

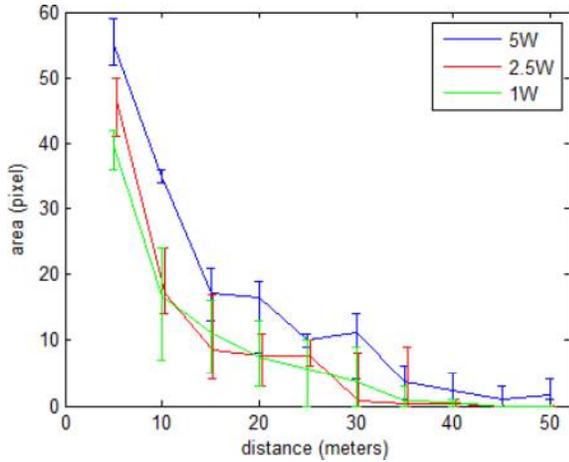


Fig. 8: Detected blob size vs. the distance between the beacon and the camera. Blue, red and green curves are recorded with average beacon powers of 5 W, 2.5 W and 1 W.

IV. APPLICATION

The proposed beacons method was applied in an indoor localization system [7]. The system utilizes beacons deployed at fixed and known positions, while the moving camera is to be localized.

The beacons can be part of the existing lighting infrastructure, which is convenient and unnoticeable for the users. Since the detection does not require special cameras, the system can be used with smartphones as well, to provide accurate indoor positioning. In the tests inexpensive wide angle (fisheye) web-cameras were utilized.

The system can utilize arbitrary number of beacons, which are detected using the proposed solution. From the detections and the known beacon positions the actual camera position can be calculated using trilateration or other fusion techniques [7].

A test scenario is shown in Figure 9. In this measurement 4 beacons were, placed at the corners of a gymnasium of size 17 m x 30 m. The altitude of the sensors varied between 3 and 4 meters.

The image taken by the fisheye camera is shown in Figure 9(a). The beacon positions are marked and the beacons are color-coded for better visibility.

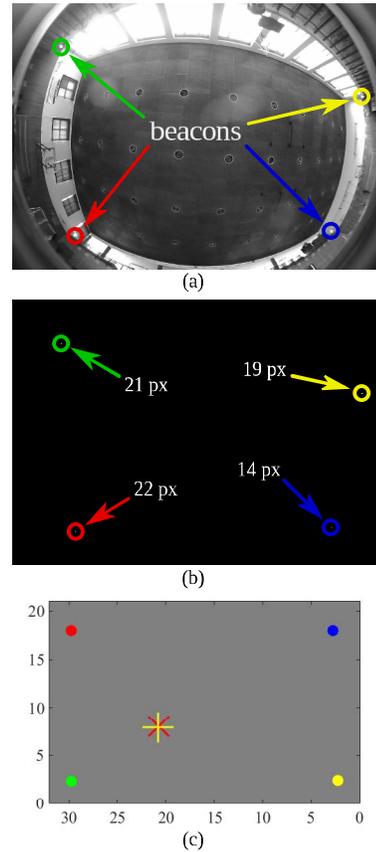


Fig. 9: Localization of the camera in a gymnasium. (a) Camera image. (b) Detected beacons (ID is color coded) with blob sizes. (c) Localization. Dots: fixed beacons, yellow cross: estimated camera position, red cross: true position.

Figure 9(b) shows the detection images $DIM_1 \dots DIM_4$, plotted in one figure, using the same color coding to identify beacons. The sizes of the blobs, corresponding to the detected beacons, are also shown.

In Figure 9(c) the localization result is shown. The figure shows a map, where the known beacon positions are shown with color dots. The true and estimated positions are denoted by red and yellow crosses, respectively. In the experiment the true camera position was (20.8 m, 7.91 m), while the detected position was (20.73 m, 8.03 m), the error being 0.14 m.

Multiple tests were performed to characterize the accuracy of the localization system. Using 4 beacons and 49 test positions, scattered in the test area shown in Figure 9, the mean absolute error was 0.17 m.

V. CONCLUSION

A beacons and a corresponding beacon detection scheme was proposed using LED lights and inexpensive cameras. The beacons use VLC to transmit their IDs, using a coding scheme which enables flicker-free user experience. The high frequency blinking of the beacons is detected by ordinary cameras, utilizing proper undersampling of the signal. A method to design the system parameters was also presented.

According to measurement results, the proposed detector requires only a small number of image pixels (7) to provide robust beacon detection and identification. In the presented proof of concept system, using average power of 5 W for the LEDs, the beacons were reliably detected from distances larger than 30 m. The proposed beaconing scheme was illustrated in a localization case study, where the unknown camera location was determined using 4 fixed beacons, with accuracy in the decimeter range.

REFERENCES

- [1] Z. Chen, Q. Zhu, H. Jiang, and Y. C. Soh, "Indoor localization using smartphone sensors and iBeacons," in *Proceedings of IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 1723–1728.
- [2] H. Lim, L. C. Kung, J. C. Hou, and H. Luo, "Zero-Configuration, Robust Indoor Localization: Theory and Experimentation," in *Proceedings IEEE INFOCOM 2006*, 2006, pp. 1–12.
- [3] P. Lazik, N. Rajagopal, B. Sinopoli, and A. Rowe, "Ultrasonic time synchronization and ranging on smartphones," in *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS*, 2015, pp. 108–117.
- [4] S. Huseth, B. Dewberry, and R. McCrosky, "Pulsed-RF Ultrawideband Ranging for the GLANSER GPS-Denied Emergency Responder Navigation System," in *Proceedings of 2011 International Technical Meeting of The Institute of Navigation*, 2011, pp. 389–396.
- [5] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'14. Berkeley, CA, USA: USENIX Association, 2014, pp. 331–343.
- [6] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 447–458.
- [7] "Microsoft Indoor Localization Competition – IPSN 2015." [Online]. Available: <https://www.microsoft.com/en-us/research/event/microsoft-indoorlocalization-competition-ipn-2015/>