

# Luxapose: Indoor Positioning with Mobile Phones and Visible Light

Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta  
Electrical Engineering and Computer Science Department  
University of Michigan  
Ann Arbor, MI 48109  
{samkuo,ppannuto,coolmark,prabal}@umich.edu

## ABSTRACT

We explore the indoor positioning problem with unmodified smartphones and slightly-modified commercial LED luminaires. The luminaires—modified to allow rapid, on-off keying—transmit their identifiers and/or locations encoded in human-imperceptible optical pulses. A camera-equipped smartphone, using just a single image frame capture, can detect the presence of the luminaires in the image, decode their transmitted identifiers and/or locations, and determine the smartphone’s location and orientation relative to the luminaires. Continuous image capture and processing enables continuous position updates. The key insights underlying this work are (i) the driver circuits of emerging LED lighting systems can be easily modified to transmit data through on-off keying; (ii) the rolling shutter effect of CMOS imagers can be leveraged to receive many bits of data encoded in the optical transmissions with just a single frame capture, (iii) a camera is intrinsically an angle-of-arrival sensor, so the projection of multiple nearby light sources with known positions onto a camera’s image plane can be framed as an instance of a sufficiently-constrained angle-of-arrival localization problem, and (iv) this problem can be solved with optimization techniques. We explore the feasibility of the design through an analytical model, demonstrate the viability of the design through a prototype system, discuss the challenges to a practical deployment including usability and scalability, and demonstrate decimeter-level accuracy in both carefully controlled and more realistic human mobility scenarios.

## Categories and Subject Descriptors

B.4.2 [HARDWARE]: Input/Output and Data Communications—*Input/Output Devices*; C.3 [COMPUTER-COMMUNICATION NETWORKS]: Special-Purpose and Application-Based Systems

## General Terms

Design, Experimentation, Measurement, Performance

## Keywords

Indoor localization; Mobile phones; Angle-of-arrival; Image processing

Permission to make digital or hard copies of part or all of this work is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

Copyright is held by the authors.

MobiCom’14, September 7–11, 2014, Maui, HI, USA.  
ACM 978-1-4503-2783-1/14/09.  
<http://dx.doi.org/10.1145/2639108.2639109>

## 1. INTRODUCTION

Accurate indoor positioning can enable a wide range of location-based services across many sectors. Retailers, supermarkets, and shopping malls, for example, are interested in indoor positioning because it can provide improved navigation which helps avoid unrealized sales when customers cannot find items they seek, and it increases revenues from incremental sales from targeted advertising [11]. Indeed, the desire to deploy indoor location-based services is one reason that the overall demand for mobile indoor positioning in the retail sector is projected to grow to \$5 billion by 2018 [7]. However, despite the strong demand forecast, indoor positioning remains a “grand challenge,” and no existing system offers accurate location and orientation using unmodified smartphones [13].

WiFi and other RF-based approaches deliver accuracies measured in meters and no orientation information, making them a poor fit for many applications like retail navigation and shelf-level advertising [2, 5, 31]. Visible light-based approaches have shown some promise for indoor positioning, but recent systems offer landmarks with approximate room-level semantic localization [21], depend on custom hardware and received signal strength (RSS) techniques that are difficult to calibrate, or require phone attachments and user-in-the-loop gestures [13]. These limitations make deploying indoor positioning systems in “bring-your-own-device” environments, like retail, difficult. Section 2 discusses these challenges in more detail noting, among other things, that visible light positioning (VLP) systems have demonstrated better performance than RF-based ones.

Motivated by a recent claim that “the most promising method for the new VLP systems is angle of arrival” [1], we propose a new approach to accurate indoor positioning that leverages trends in solid-state lighting, camera-enabled smartphones, and retailer-specific mobile applications. Our design consists of visible light beacons, smartphones, and a cloud/cloudlet server that work together to determine a phone’s location and orientation, and support location-based services. Each beacon consists of a programmable oscillator or microcontroller that controls one or more LEDs in a luminaire. A beacon’s identity is encoded in the modulation frequency (or Manchester-encoded data stream) and optically broadcast by the luminaire. The smartphone’s camera takes pictures periodically and these pictures are processed to determine if they contain any beacons by testing for energy in a target spectrum of the columnar FFT of the image. If beacons are present, the images are decoded to determine the beacon location and identity. Once beacon identities and coordinates are determined, an angle-of-arrival localization algorithm determines the phone’s absolute position and orientation in the local coordinate system. Section 3 presents an overview of our proposed approach, including the system components, their interactions, and the data processing pipeline that yields location and orientation from a single image of the lights and access to a lookup table.

Our angle-of-arrival positioning principle assumes that three or more beacons (ideally at least four) with known 3-D coordinates have been detected and located in an image captured by a smartphone. We assume that these landmarks are visible and distinguishable from each other. This is usually the case when the camera is in focus since unoccluded beacons that are separated in space uniquely project onto the camera imager at distinct points. Assuming that the camera geometry is known and the pixels onto which the beacons are projected is determined, we estimate the position and orientation of the smartphone with respect to the beacons' coordinate system through the geometry of similar triangles, using a variation on the well-known bearings-only robot localization and mapping problem [10]. Section 4 describes the details of estimating position and orientation, and dealing with noisy measurements.

So far, we have assumed that our positioning algorithm is given the identities and locations of beacons within an overhead scene image, but we have not discussed how are these extracted from an image of modulated LEDs. Recall that the beacons are modulated with a square wave or transmit Manchester-encoded data (at frequencies above 1 kHz to avoid direct or indirect flicker [26]). When a smartphone passes under a beacon, the beacon's transmissions are projected onto the camera. Although the beacon frequency far exceeds the camera's frame rate, the transmissions are still decodable due to the rolling shutter effect [9]. CMOS imagers that employ a rolling shutter expose one or more columns at once, and scan just one column at a time. When an OOK-modulated light source illuminates the camera, distinct light and dark bands appear in images. The width of the bands depend on the scan time, and crucially, on the frequency of the light. We employ an image processing pipeline, as described in Section 5, to determine the extent of the beacons, estimate their centroids, and extract their embedded frequencies, which yields the inputs needed for positioning.

To evaluate the viability and performance of this approach, we implement the proposed system using both custom and slightly-modified commercial LED luminaires, a Nokia Lumia 1020 smartphone, and an image processing pipeline implemented using OpenCV, as described in Section 6. We deploy our proof-of-concept system in a university lab and find that under controlled settings with the smartphone positioned under the luminaires, we achieve decimeter-level location and roughly  $3^\circ$  orientation error under lights when four or five beacons are visible. With fewer than four visible beacons, or when errors are introduced in the beacon positions, we find that localization errors increase substantially. Fortunately, in realistic usage conditions—a person carrying a smartphone beneath overhead lights—we observe decimeter position and single-digit orientation errors. Although difficult to directly compare different systems, we adopt the parameters proposed by Epsilon [13] and compare the performance of our system to the results reported in prior work in Table 1. These results, and others benchmarking the performance of the VLC channel, are presented in Section 7.

Our proposed system, while promising, also has a number of limitations. It requires a high density of overhead lights with known positions, and nearby beacons to have accurate relative positions. Adequate performance requires high-resolution cameras which have only recently become available on smartphones. We currently upload entire images to the cloud/cloudlet server for processing, which incurs a significant time and energy cost that is difficult to accurately characterize. However, we show simple smartphone-based algorithms that can filter images locally, or crop only the promising parts of an image, reducing transfer costs or even enabling local processing. Section 8 discusses these and other issues, and suggests that it may soon be possible to achieve accurate indoor positioning using unmodified smartphones in realistic retail settings.

Param	EZ	Radar	Horus	Epsilon	Luxapose
Reference	[5]	[2]	[31]	[13]	[this]
Position	2-7 m	3-5 m	$\sim 1$ m	$\sim 0.4$ m	$\sim 0.1$ m
Orientation	n/a	n/a	n/a	n/a	$3^\circ$
Method	Model	FP	FP	Model	AoA
Database	Yes	Yes	Yes	No	Yes
Overhead	Low	WD	WD	DC	DC

Table 1: Comparison with prior WiFi- and VLC-based localization systems. FP, WD, AoA, and DC are FingerPrinting, War-Driving, Angle-of-Arrival, and Device Configuration, respectively. These are the reported figures from the cited works.

## 2. RELATED WORK

There are three areas of related work: RF localization, visible light communications, and visible light positioning.

**RF-Based Localization.** The majority of indoor localization research is RF-based, including WiFi [2, 5, 15, 27], Motes [14], and FM radio [4], although some have explored magnetic fingerprinting as well [6]. All of these approaches achieve meter-level accuracy, and no orientation, often through RF received signal strength from multiple beacons, or with location fingerprinting [4, 6, 14, 29]. Some employ antenna arrays and track RF phase to achieve sub-meter accuracy, but at the cost of substantial hardware modifications [27]. In contrast, we offer decimeter-level accuracy at the 90th-percentile under typical overhead lighting conditions, provide orientation, use camera-based localization, require no hardware modifications on the phone and minor modifications to the lighting infrastructure.

**Visible Light Communications.** A decade of VLC research primarily has focused on high-speed data transfer using specialized transmitters and receivers that support OOK, QAM, or DMT/OFDM modulation [12], or the recently standardized IEEE 802.15.7 [22]. However, smartphones typically employ CdS photocells with wide dynamic range but insufficient bandwidth for typical VLC [21]. In addition, CdS photocells cannot determine angle-of-arrival, and while smartphone cameras can, they cannot support most VLC techniques due to their limited frame rates. Recent research has shown that by exploiting the rolling shutter effect of CMOS cameras, it is possible to receive OOK data at close range, from a single transmitter, with low background noise [9]. We also use the same effect but operate at 2-3 meter range from typical luminaires, support multiple concurrent transmitters, and operate with ambient lighting levels.

**Visible Light-Based Localization.** Visible light positioning using one [19, 30, 32] or more [20, 28] image sensors has been studied in simulation. In contrast, we explore the performance of a real system using a CMOS camera present in a commercial smartphone, address many practical concerns like dimming and flicker, and employ robust decoding and localization methods that work in practice. Several visible light positioning systems have been implemented [13, 21, 24]. ALTAIR uses ceiling-mounted cameras, body-worn IR LED tags, and a server that instructs tags to beacon sequentially, captures images from the cameras, and performs triangulation to estimate position [24]. Epsilon uses LED beacons and a custom light sensor that plugs into a smartphone's audio port, and sometimes requires users to perform gestures [13]. The LEDs transmit data using BFSK and avoid persistent collisions by random channel hopping. The system offers half-meter accuracy. In contrast, we require no custom hardware on the phone, can support a high density of lights without coordination, require no special gestures, provide orientation, and typically offer better performance. Landmarks provides semantic (e.g. room-level) localization using rolling shutter-based VLC [21], but neither accurate position nor orientation, like our system does.

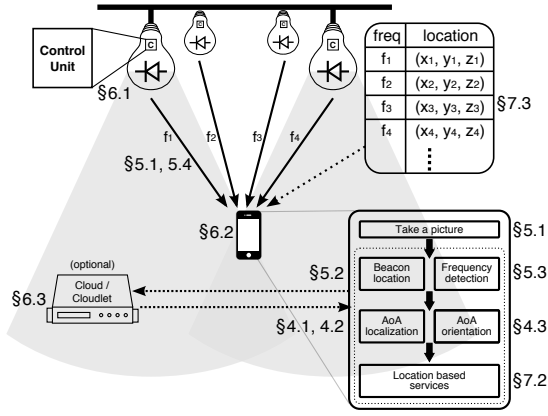


Figure 1: Luxapose indoor positioning system architecture and roadmap to the paper. The system consists of visible light beacons, mobile phones, and a cloud/cloudlet server. Beacons transmit their identities or coordinates using human-imperceptible visible light. A phone receives these transmissions using its camera and recruits a combination of local and cloud resources to determine its precise location and orientation relative to the beacons' coordinate system using an angle-of-arrival localization algorithm, thereby enabling location-based services.

### 3. SYSTEM OVERVIEW

The Luxapose indoor positioning system consists of visible light beacons, smartphones, and a cloud/cloudlet server, as Figure 1 shows. These elements work together to determine a smartphone's location and orientation, and support location-based services. Each beacon consists of a programmable oscillator or microcontroller that modulates one or more LED lights in a light fixture to broadcast the beacon's identity and/or coordinates. The front-facing camera in a hand-held smartphone takes pictures periodically. These pictures are processed to determine if they contain LED beacons by testing for the presence of certain frequencies. If beacons are likely present, the images are decoded to both determine the beacon locations in the image itself and to also extract data encoded in the beacons' modulated transmissions. A lookup table may be consulted to convert beacon identities into corresponding coordinates if these data are not transmitted. Once beacon identities and coordinates are determined, an angle-of-arrival localization algorithm determines the phone's position and orientation in the venue's coordinate system. This data can then be used for a range of location-based services. Cloud or cloudlet resources may be used to assist with image processing, coordinate lookup, database lookups, indoor navigation, dynamic advertisements, or other services that require distributed resources.

### 4. POSITIONING PRINCIPLES

Our goal is to estimate the location and orientation of a smartphone assuming that we know bearings to three or more point-sources (interchangeably called *beacons*, *landmarks*, and *transmitters*) with known 3-D coordinates. We assume the landmarks are visible and distinguishable from each other using a smartphone's built-in camera (or *receiver*). The camera is in focus so these point sources uniquely project onto the camera imager at distinct pixel locations. Assuming that the camera geometry (e.g. pixel size, focal length, etc.) is known and the pixels onto which the landmarks are projected can be determined, we seek to estimate the position and orientation of the mobile device with respect to the landmarks' coordinate system. This problem is a variation on the well-known bearings-only robot localization and mapping problem [10].

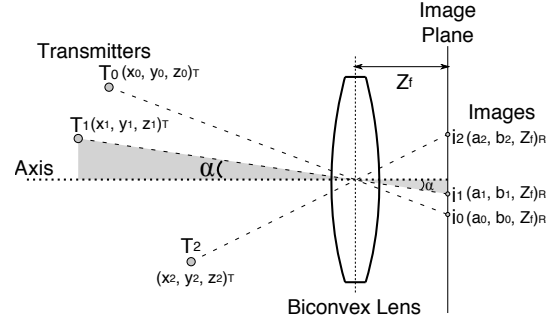


Figure 2: Optical AoA localization. When the scene is in focus, transmitters are distinctly projected onto the image plane. Knowing the transmitters' locations  $T_j(x_j, y_j, z_j)_T$  in a global reference frame, and their image  $i_j(a_j, b_j, Z_f)_R$  in the receiver's reference frame, allows us to estimate the receiver's global location and orientation.

#### 4.1 Optical Angle-of-Arrival Localization

Luxapose uses optical angle-of-arrival (AoA) localization principles based on an ideal camera with a biconvex lens. An important property of a simple biconvex lens is that a ray of light that passes through the center of the lens is not refracted, as shown in Figure 2. Thus, a transmitter, the center of the lens, and the projection of transmitter onto the camera imager plane form a straight line. Assume that transmitter  $T_0$ , with coordinates  $(x_0, y_0, z_0)_T$  in the transmitters' global frame of reference, has an image  $i_0$ , with coordinates  $(a_0, b_0, Z_f)_R$  in the receiver's frame of reference (with the origin located at the center of the lens).  $T_0$ 's position falls on the line that passes through  $(0, 0, 0)_R$  and  $(a_0, b_0, Z_f)_R$ , where  $Z_f$  is the distance from lens to imager in pixels. By the geometry of similar triangles, we can define an unknown scaling factor  $K_0$  for transmitter  $T_0$ , and describe  $T_0$ 's location  $(u_0, v_0, w_0)_R$  in the receiver's frame of reference as:

$$\begin{aligned} u_0 &= K_0 \times a_0 \\ v_0 &= K_0 \times b_0 \\ w_0 &= K_0 \times Z_f \end{aligned}$$

Our positioning algorithm assumes that transmitter locations are known. This allows us to express the pairwise distance between transmitters in both the transmitters' and receiver's frames of reference. Equating the expressions in the two different domains yields a set of quadratic equations in which the only remaining unknowns are the scaling factors  $K_0, K_1, \dots, K_n$ . For example, assume three transmitters  $T_0, T_1$ , and  $T_2$  are at locations  $(x_0, y_0, z_0)_T$ ,  $(x_1, y_1, z_1)_T$ , and  $(x_2, y_2, z_2)_T$ , respectively. The pairwise distance squared between  $T_0$  and  $T_1$ , denoted  $d_{0,1}^2$ , can be expressed in both domains, and equated as follows:

$$\begin{aligned} d_{0,1}^2 &= (u_0 - u_1)^2 + (v_0 - v_1)^2 + (w_0 - w_1)^2 \\ &= (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2 \\ &= K_0^2 |\vec{O_{i_0}}|^2 + K_1^2 |\vec{O_{i_1}}|^2 - 2K_0 K_1 (\vec{O_{i_0}} \cdot \vec{O_{i_1}}) \\ &= (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2, \end{aligned}$$

where  $\vec{O_{i_0}}$  and  $\vec{O_{i_1}}$  are the vectors from the center of the lens to image  $i_0(a_0, b_0, Z_f)$  and  $i_1(a_1, b_1, Z_f)$ , respectively. The only unknowns are  $K_0$  and  $K_1$ . Three transmitters would yield three quadratic equations in three unknown variables, allowing us to find  $K_0, K_1$ , and  $K_2$ , and compute the transmitters' locations in the receiver's frame of reference.





Figure 3: Receiver orientation. The vectors  $x'$ ,  $y'$ , and  $z'$  are defined as shown in the picture. The projection of the unit vectors  $\hat{x}'$ ,  $\hat{y}'$ , and  $\hat{z}'$  onto the  $x$ ,  $y$ , and  $z$  axes in the transmitters' frame of reference give the elements of the rotation matrix  $\mathbf{R}$ .

## 4.2 Estimating Receiver Position

In the previous section, we show how the transmitters' locations in the receiver's frame of reference can be calculated. In practice, imperfections in the optics and inaccuracies in estimating the transmitters' image locations make closed-form solutions unrealistic. To address these issues, and to leverage additional transmitters beyond the minimum needed, we frame position estimation as an optimization problem that seeks the minimum mean square error (MMSE) over a set of scaling factors, as follows:

$$\sum_{m=1}^{N-1} \sum_{n=m+1}^N \{K_m^2 |\vec{O}_{im}|^2 + K_n^2 |\vec{O}_{in}|^2 - 2K_m K_n (\vec{O}_{im} \cdot \vec{O}_{in}) - d_{mn}^2\}^2,$$

where  $N$  is the number of transmitters projected onto the image, resulting in  $\binom{N}{2}$  equations.

Once all the scaling factors are estimated, the transmitters' locations can be determined in the receiver's frame of reference, and the distances between the receiver and transmitters can be calculated. The relationship between two domains can be expressed as follows:

$$\begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \\ y_0 & y_1 & \dots & y_{N-1} \\ z_0 & y_1 & \dots & z_{N-1} \end{bmatrix} = \mathbf{R} \times \begin{bmatrix} u_0 & u_1 & \dots & u_{N-1} \\ v_0 & v_1 & \dots & v_{N-1} \\ w_0 & w_1 & \dots & w_{N-1} \end{bmatrix} + \mathbf{T},$$

where  $\mathbf{R}$  is a 3-by-3 rotation matrix and  $\mathbf{T}$  is a 3-by-1 translation matrix.

The elements of  $\mathbf{T}$  ( $T_x, T_y, T_z$ ) represent the receiver's location in the transmitters' frame of reference. We determine the translation matrix based on geometric relationships. Since scaling factors are now known, equivalent distances in both domains allow us to obtain the receiver's location in the transmitters' coordinate system:

$$(T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 = K_m^2 (a_m^2 + b_m^2 + Z_f^2),$$

where  $(x_m, y_m, z_m)$  are the coordinates of the  $m$ -th transmitter in the transmitters' frame of reference, and  $(a_m, b_m)$  is the projection of the  $m$ -th transmitter onto the image plane. Finally, we estimate the receiver's location by finding the set  $(T_x, T_y, T_z)$  that minimizes:

$$\sum_{m=1}^N \{(T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 - K_m^2 (a_m^2 + b_m^2 + Z_f^2)\}^2$$

## 4.3 Estimating Receiver Orientation

Once the translation matrix  $\mathbf{T}$  is known, we can find the rotation matrix  $\mathbf{R}$  by individually finding each element in it. The 3-by-3 rotation matrix  $\mathbf{R}$  is represented using three column vectors,  $\vec{r}_1$ ,  $\vec{r}_2$ , and  $\vec{r}_3$ , as follows:

$$\mathbf{R} = \begin{bmatrix} \vec{r}_1 & \vec{r}_2 & \vec{r}_3 \end{bmatrix},$$

where the column vectors  $\vec{r}_1$ ,  $\vec{r}_2$  and  $\vec{r}_3$  are the components of the unit vectors  $\hat{x}'$ ,  $\hat{y}'$ , and  $\hat{z}'$ , respectively, projected onto the  $x$ ,  $y$ , and  $z$  axes in the transmitters' frame of reference. Figure 3 illustrates the relationships between these various vectors. Once the orientation of the receiver is known, determining its bearing requires adjusting for portrait or landscape mode usage, and computing the projection onto the  $xy$ -plane.

## 5. CAMCOM PHOTOGRAMMETRY

Our positioning scheme requires that we identify the points in a camera image,  $(a_i, b_i, Z_f)$ , onto which each landmark  $i \in 1 \dots N$  with known coordinates,  $(x_i, y_i, z_i)$ , are projected, and map between the two domains. This requires us to: (i) identify landmarks in an image, (ii) label each landmark with an identity, and (iii) map that identity to the landmark's global coordinates. To help with this process, we modify overhead LED luminaires so that they beacon optically—by rapidly switching on and off—in a manner that is imperceptible to humans but detectable by a smartphone camera.

We label each landmark by either modulating the landmark's LED at a fixed frequency or by transmitting Manchester-encoded data in the landmark's transmissions (an approach called camera communications, or *CamCom*, that enables low data rate, unidirectional message broadcasts from LEDs to image sensors), as Section 5.1 describes. We detect the presence and estimate the centroids and extent of landmarks in an image using the image processing pipeline described in Section 5.2. Once the landmarks are found, we determine their identities by decoding data embedded in the image, which either contains an index to, or the actual value of, a landmark's coordinates, as described in Section 5.3. Finally, we estimate the capacity of the CamCom channel we employ in Section 5.4.

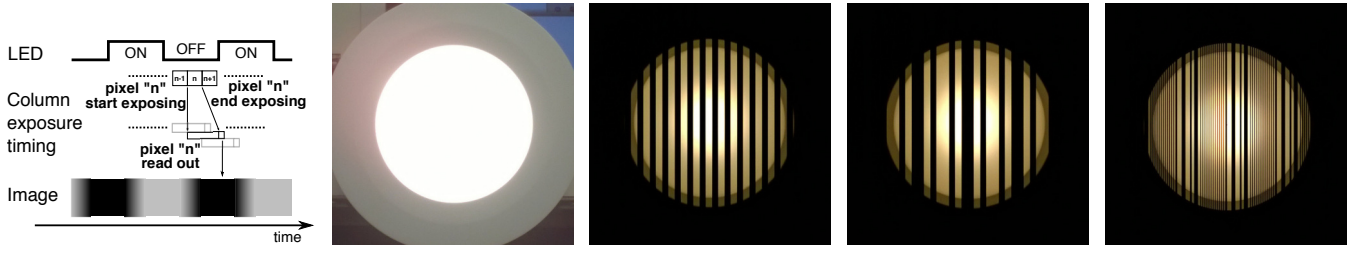
### 5.1 Encoding Data in Landmark Beacons

Our system employs a unidirectional communications channel that uses an LED as a transmitter and a smartphone camera as a receiver. We encode data by modulating signals on the LED transmitter. As our LEDs are used to illuminate the environment, it is important that our system generates neither direct nor indirect flicker (the stroboscopic effect). The Lighting Research Center found that for any duty cycle, a luminaire with a flicker rate over 1 kHz was acceptable to room occupants, who could perceive neither effect [26].

#### 5.1.1 Camera Communications Channel

When capturing an image, most CMOS imagers expose one or more columns of pixels, but read out only one column at a time, sweeping across the image at a fixed *scan rate* to create a *rolling shutter*, as shown in Figure 4a. When a rapidly modulated LED is captured with a CMOS imager, the result is a banding effect in the image in which some columns capture the LED when it is on and others when it is off. This effect is neither visible to the naked eye, nor in a photograph that uses an auto-exposure setting, as shown in Figure 4b. However, the rolling shutter effect is visible when an image is captured using a short exposure time, as seen in Figure 4c.

In the Luxapose design, each LED transmits a single frequency (from roughly 25-30 choices) as Figure 4c shows, allowing different LEDs or LED constellations to be distinctly identified. To expand the capacity of this channel, we also explore Manchester encoded data transmission, which is appealing both for its simplicity and its absence of a DC-component, which supports our data-independent brightness constraint. Figure 4d shows an image captured by an unmodified Lumia 1020 phone 1 m away from a commercial 6 inch can light. Our goal is to illustrate the basic viability of sending information over our VLC channel, but leave to future work the problem of determining the optimal channel coding.



(a) Banding pattern due to the rolling shutter effect of a CMOS camera capturing a rapidly flashing LED. Adjusting the LED frequency or duty cycle changes the width of light and dark bands in the image, allowing frequency to be detected and decoded. (b) Auto Exposure. Image of an LED modulated at 1 kHz with a 50% duty-cycle taken with the built-in camera app with a default (auto) exposure settings. The modulation is imperceptible. (c) Pure Tone. Image of an LED modulated at 1 kHz with a 50% duty-cycle taken with a short exposure setting. The modulation is clearly visible as a series of alternating light and dark bands. (d) Manchester Encoding. Image of an LED modulated at 1 kHz with a 50% duty-cycle transmitting Manchester-encoded data taken with a short exposure setting. Data repeats 0x66. (e) Hybrid Encoding. Image of an LED alternating between transmitting a 3 kHz Manchester encoded data stream and a 6 kHz pure tone. Data is 4 symbols and the preamble is 2 symbols.

Figure 4: CMOS rolling shutter principles and practice using various encoding schemes. All images are taken by a Lumia 1020 camera of a modified Commercial Electric T66 6 inch (10 cm) ceiling-mounted can LED. The camera is 1 m from the LED and pictures are taken with the back camera. The images shown are a  $600 \times 600$  pixel crop focusing on the transmitter, yielding a transmitter image with about a 450 pixel diameter. The ambient lighting conditions are held constant across all images, demonstrating the importance of exposure control for CamCom.

### 5.1.2 Camera Control

Cameras export many properties that affect how they capture images. The two most significant for the receiver in our CamCom channel are exposure time and film speed.

**Exposure Control.** Exposure time determines how long each pixel collects photons. During exposure, a pixel's charge accumulates as light strikes, until the pixel saturates. We seek to maximize the relative amplitude between the on and off bands in the captured image. Figure 5 shows the relative amplitude across a range of exposure values. We find that independent of film speed (ISO setting), the best performance is achieved with the shortest exposure time. The direct ray of light from the transmitter is strong and requires less than an on-period of the transmitted signal to saturate a pixel. For a 1 kHz signal (0.5 ms on, 0.5 ms off), an exposure time of longer than 0.5 ms ( $1/2000$  s) guarantees that each pixel will be at least partially exposed to an on period, which would reduce possible contrast and result in poorer discrimination between light and dark bands.

**Film Speed.** Film speed (ISO setting) determines the sensitivity or gain of the image sensor. Loosely, it is a measure of how many photons are required to saturate a pixel. A faster film speed (higher ISO) increases the gain of the pixel sense circuitry, causing each pixel to saturate with fewer photons. If the received signal has a low amplitude (far from the transmitter or low transmit power), a faster film speed could help enhance the image contrast and potentially enlarge the decoding area. It also introduces the possibility of amplifying unwanted reflections above the noise floor, however. As Figure 5 shows, a higher film speed increases the importance of a shorter exposure time for high contrast images. We prefer smaller ISO values due to the proximity and brightness of indoor lights.

## 5.2 Finding Landmarks in an Image

Independent of any modulated data, the first step is to find the centroid and size of each transmitter on the captured image. We present one method in Figures 6a to 6e for identifying disjoint, circular transmitters (e.g. individual light fixtures). We convert the image to grayscale, blur it, and pass it through a binary OTSU filter [18]. We find contours for each blob [25] and then find the minimum enclosing circle (or other shape) for each contour. After finding each of the transmitters, we examine each subregion of the image independently to decode data from each light. We discuss approaches for processing other fixture types, such as Figure 18, in Section 8.

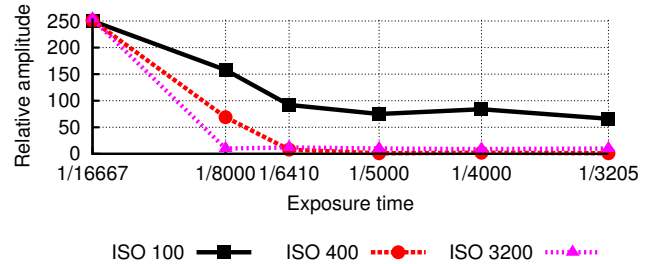


Figure 5: Maximizing SNR, or the ratio between the brightest and darkest pixels in an image. The longer the exposure, the higher the probability that a pixel accumulates charge while another saturates, reducing the resulting contrast between the light and dark bands. As film speed (ISO) increases, a fewer number of photons are required to saturate each pixel. Hence, we minimize the exposure time and film speed to maximize the contrast ratio, improving SNR.

## 5.3 Decoding Data in Images

Once the centroid and extent of any landmarks are found in an image, the next step is to extract the data encoded in beacon transmissions in these regions using one of four methods.

**Decoding Pure Tones – Method One.** Our first method of frequency decoding samples the center row of pixels across an image subregion and takes an FFT of that vector. While this approach decodes accurately, we find that it is not very precise, requiring roughly 200 Hz of separation between adjacent frequencies to reliably decode. We find in our evaluation, however, that this approach decodes more quickly and over longer distances than method two, creating a tradeoff space, and potential optimization opportunities.

**Decoding Pure Tones – Method Two.** Figures 6g to 6j show our second method, an image processing approach. We first apply a vertical blur to the subregion and then use an OTSU filter to get threshold values to pass into the Canny edge detection algorithm [3]. Note the extreme pixelation seen on the edges drawn in Figure 6i; these edges are only 1 pixel wide. The transmitter captured in this subregion has a radius of only 35 pixels. To manage this quantization, we exploit the noisy nature of the detected vertical edge and compute the weighted average of the edge location estimate across each row, yielding a subpixel estimation of the column containing the edge.

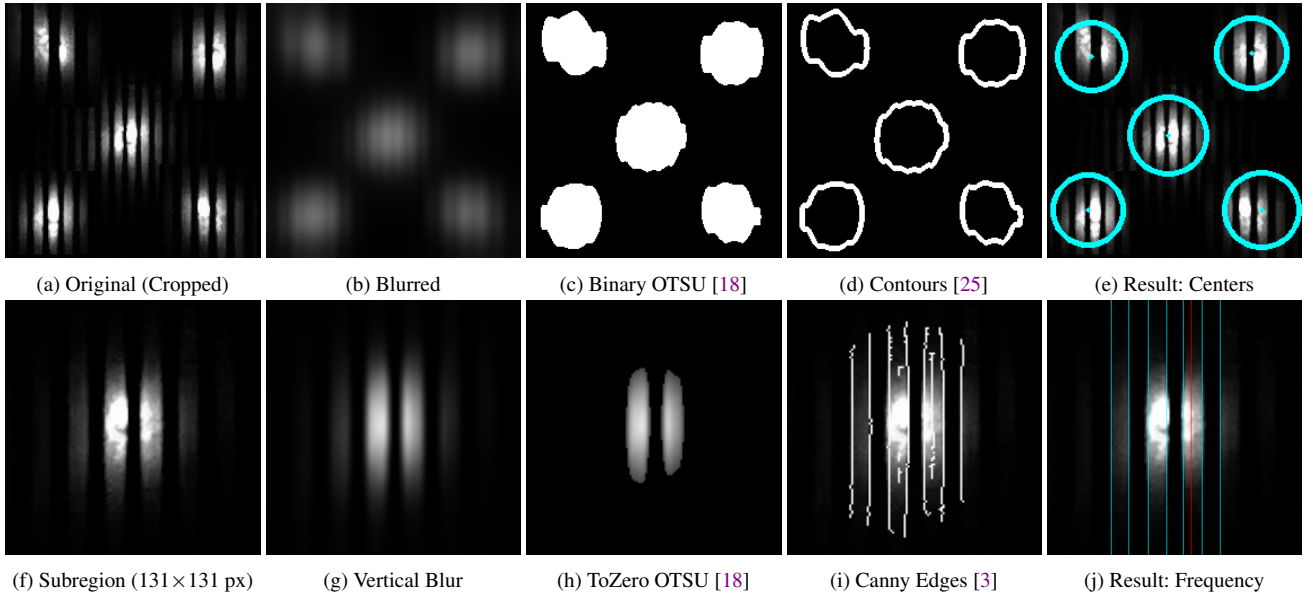


Figure 6: Image processing pipeline. The top row of images illustrate our landmark detection algorithm. The bottom row of images illustrate our image processing pipeline for frequency recovery. These images are edited to move the transmitters closer together for presentation.

Near the transmitter center, erroneous edges are sometimes identified if the intensity of an *on* band changes too quickly. We majority vote across three rows of the subregion (the three rows equally partition the subregion) to decide if each interval is light or dark. If an edge creates two successive light intervals, it is considered an error and removed. Using these precise edge estimates and the known scan rate, we convert the interval distance in pixels to the transmitted frequency with a precision of about 50 Hz., offering roughly 120 channels (6 kHz/50 Hz). In addition to the extra edge detection and removal we also attempt to detect and insert missing edges. We compute the interval values between each pair of edges and look for intervals that are statistical outliers. If the projected frequency from the non-outlying edges divides cleanly into the outlier interval, then we have likely identified a missing edge, and so we add it.

**Decoding Manchester Data.** To decode Manchester data, we use a more signal processing-oriented approach. Like the FFT for tone, we operate on only the center row of pixels from the subregion. We use a matched filter with a known pattern (a preamble) at different frequencies and search for the maximum correlation. When found, the maximum correlation also reveals the preamble location. The frequency of the matched filter is determined by the number of pixels per symbol. It can be calculated as  $\frac{F_s}{2 \times n}$ , where  $F_s$  is the sampling rate of the camera and  $n$  is an integer. As the frequency increases,  $n$  decreases, and the quantization effect grows. For example,  $F_s$  on the Lumia 1020 is 47.54 kHz, so an  $n$  value of 5 matches a 4.75 kHz signal. Using the best discrete matched filter, we search for the highest correlation value anywhere along the real pixel x-axis, allowing for a subpixel estimation of symbol location, repeating this process for each symbol.

**Decoding Hybrid Transmissions.** To balance the reliability of detecting pure tones with the advantages of Manchester encoded data, we explore a hybrid approach, alternating the transmission of a pure tone and Manchester encoded data, as Figure 4e shows. By combining frequency and data transmission, we decouple localization from communication. When a receiver is near a transmitter, it can take advantage of the available data channel, but it can also decode the frequency information of lights that are far away, increasing the probability of a successful localization.

## 5.4 Estimating Channel Capacity

The size of the transmitter and its distance from the receiver dictate the area that the transmitter projects onto the imager plane. The bandwidth for a specific transmitter is determined by its image length (in pixels) along the CMOS scan direction. Assuming a circular transmitter with diameter  $A$  m, its length on the image sensor is  $A \times f/h$  pixels, where  $f$  is the focal length of the camera and  $h$  is the height from the transmitter to the receiver. The field of view (FoV) of a camera can be expressed as  $\alpha = 2 \times \arctan(\frac{X}{2 \times f})$ , where  $X$  is the length of the image sensor along the direction of the FoV. Combining these, the length of the projected transmitter can be expressed as  $\frac{A \times X}{h \times 2 \times \tan(\text{FoV}/2)}$ .

As an example, in a typical retail setting,  $A$  is 0.3–0.5 m and  $h$  is 3–5 m. The Glass camera ( $X = 2528$  px,  $14.7^\circ$  FoV) has a “bandwidth” of 588–1633 px. The higher-resolution Lumia 1020 camera ( $X = 5360$  px,  $37.4^\circ$  FoV) bandwidth is actually lower, 475–1320 px, as the wider FoV maps a much larger scene area to the fixed-size imager as the distance increases. This result shows that increasing resolution alone may not increase effective channel capacity without paying attention to other camera properties.

## 6. IMPLEMENTATION DETAILS

To evaluate the viability and performance of the Luxapose design, we implement a prototype system using a variety of LED luminaires, an unmodified smartphone, and a Python-based cloudlet (all available at <https://github.com/lab11/vlc-localization/>).

### 6.1 LED Landmarks

We construct landmarks by modifying commercial LED luminaires, including can, tube, and task lamps, as shown in Figure 7a, but full-custom designs are also possible. Figure 7b shows the modifications, which include cutting (×) and intercepting a wire, and wiring in a control unit that includes a voltage regulator (VR) and a microcontroller (MCU) or programmable oscillator (OSC) controlling a single FET switch. We implement two control units, as shown in Figure 7c, for low- and high-voltage LED driver circuits, using a voltage-controlled oscillator with 16 frequency settings.



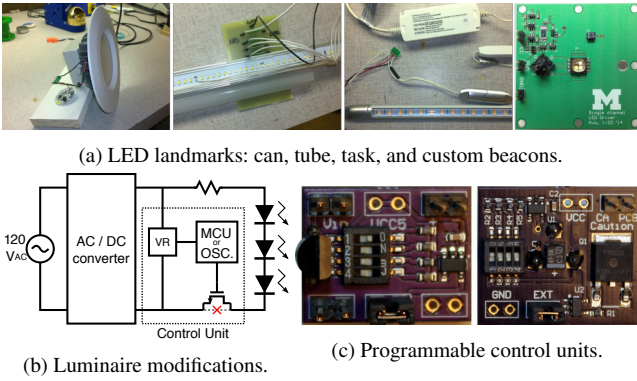


Figure 7: LED landmarks. (a) Commercial and custom LED beacons. (b) A commercial luminaire is modified by inserting a control unit. (c) Two custom control units with 16 programmable tones. The units draw 5 mA and cost ~\$3 each in quantities of 1,000, suggesting they could be integrated into commercial luminaires.

## 6.2 Smartphone Receiver

We use the Nokia Lumia 1020 to implement the Luxapose receiver design. The Lumia’s resolution— $7712 \times 5360$  pixels—is the highest among many popular phones, allowing us the greatest experimental flexibility. The deciding factor, however, is not the hardware capability of the smartphone, but rather its OS support and camera API that expose control of resolution, exposure time, and film speed. Neither Apple’s iOS nor Google’s Android currently provide the needed camera control, but we believe they are forthcoming. Only Windows Phone 8, which runs on the Lumia, currently provides a rich enough API to perform advanced photography [16].

We modify the Nokia Camera Explorer [17] to build our application. We augment the app to expose the full range of resolution and exposure settings, and we add a streaming picture mode that continuously takes images as fast as the hardware will allow. Finally, we add cloud integration, transferring captured images to our local cloudlet for processing, storage, and visualization without employing any smartphone-based optimizations that would filter images.

We emphasize that the platform support is not a hardware issue but a software issue. Exposure and ISO settings are controlled by OS-managed feedback loops. We are able to coerce these feedback loops by shining a bright light into imagers and removing it at the last moment before capturing an image of our transmitters. Using this technique, we are able to capture images with  $1/7519$  s exposure on ISO 68 film using Google Glass and  $1/55556$  s exposure and ISO 50 on an iPhone 5; we are able to recover the location information from these coerced-exposure images successfully, but evaluating using this approach is impractical, so we focus our efforts on the Lumia.

Photogrammetry—the discipline of making measurements from photographs—requires camera characterization and calibration. We use the Nokia Pro Camera application included with the Lumia, which allows the user to specify exposure and ISO settings, to capture images for this purpose. Using known phone locations, beacon locations, and beacon frequencies, we measure the distance between the lens and imager,  $Z_f$  (1039 pixels, 5620 pixels), and scan rate (30,880 columns/s, 47,540 columns/s), for the front and back cameras, respectively. To estimate the impact of manufacturing tolerances, we measure these parameters across several Lumia 1020s and find only a 0.15% deviation, suggesting that per-unit calibration is not required.

Camera optics can distort a captured image, but most smartphone cameras digitally correct distortions in the camera firmware [23]. To verify the presence and quality of distortion correction in the Lumia,



Figure 8: Indoor positioning testbed. Five LED beacons are mounted 246 cm above the ground for experiments. Ground truth is provided by a pegboard on the floor with 2.54 cm location resolution.

we move an object from the center to the edge of the camera’s frame, and find that the Lumia’s images show very little distortion, deviating at most 3 pixels from the expected location.

The distance,  $Z_f$ , between the center of lens and the imager is a very important parameter in AoA localization algorithms. Unfortunately, this parameter is not fixed on the Lumia 1020, which uses a motor to adjust the lens for sharper images. This raises the question of how this impacts localization accuracy. In a simple biconvex lens model, the relationship between  $s_1$  (distance from object to lens),  $s_2$  (from lens to image), and  $f$  (focal length) is:

$$\frac{1}{s_1} + \frac{1}{s_2} = \frac{1}{f}$$

where  $s_2$  and  $Z_f$  are the same parameter but  $s_2$  is measured in meters whereas  $Z_f$  is measured in pixels.  $s_2$  can be rewritten as  $\frac{s_1 \times f}{s_1 - f}$ . For the Lumia 1020,  $f = 7.2$  mm. In the general use case,  $s_1$  is on the order of meters which leads to  $s_2$  values between 7.25 mm ( $s_1 = 1$  m) and 7.2 mm ( $s_1 = \infty$ ). This suggests that  $Z_f$  should deviate only 0.7% from a 1 m focus to infinity. As lighting fixtures are most likely 2~5 m above ground, the practical deviation is even smaller, thus we elect to use a fixed  $Z_f$  value for localization. We measure  $Z_f$  while the camera focuses at 2.45 m across 3 Lumia phones. All  $Z_f$  values fall within 0.15% of the average: 5,620 pixels.

While the front camera is more likely to face lights in day-to-day use, we use the back camera for our experiments since it offers higher resolution. Both cameras support the same exposure and ISO ranges, but have different resolutions and scan rates. Scan rate places an upper bound on transmit frequency, but the limited exposure range places a more restrictive bound, making this difference moot. Resolution imposes an actual limit by causing quantization effects to occur at lower frequencies; the maximum frequency decodable by the front camera using edge detection is ~5 kHz, while the back camera can reach ~7 kHz. Given Hendy’s Law—the annual doubling of pixels per dollar—we focus our evaluation on the higher-resolution imager, without loss of generality.

## 6.3 Cloudlet Server

A cloudlet server implements the full image processing pipeline shown in Figure 6 using OpenCV 2.4.8 with Python bindings. On an unburdened MacBook Pro with a 2.7 GHz Core i7, the median processing time for the full 33 MP images captured by the Lumia is about 9 s (taking picture: 4.46 s, upload: 3.41 s, image processing: 0.3 s, estimate location: 0.87 s) without any optimizations. The cloudlet application contains a mapping from transmitter frequency to absolute transmitter position in space. Using this mapping and the information from the image processing, we implement the techniques described in Section 4 using the `leastsq` implementation from SciPy. Our complete cloudlet application is 722 Python SLOC.

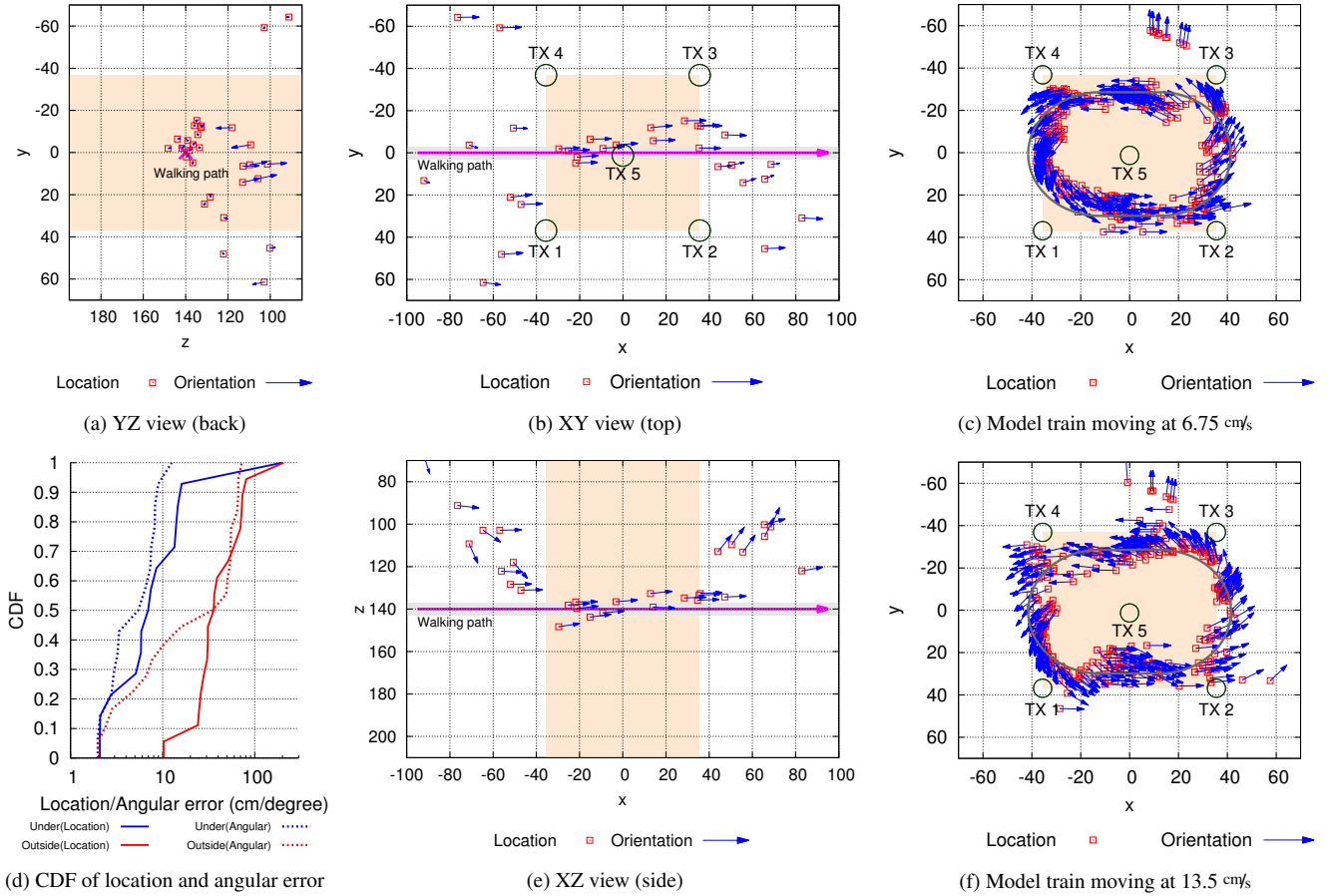


Figure 9: Key location and orientation results under realistic usage conditions on our indoor positioning testbed. The shaded areas are directly under the lights. (a), (b), and (e) show Luxapose’s estimated location and orientation of a person walking from the back, top, and side views, respectively, while using the system. A subject carrying a phone walks underneath the testbed repeatedly, trying to remain approximately under the center ( $x = -100 \dots 100$ ,  $y = 0$ ,  $z = 140$ ). We measure the walking speed at  $\sim 1$  m/s. (d) suggests location estimates (solid line) and orientation (dotted line) under the lights (blue), have lower error than outside the lights (red). (c) and (f) show the effect of motion blur. To estimate the impact of motion while capturing images, we place the smartphone on a model train running in an oval at two speeds. While the exact ground truth for each point is unknown, we find the majority of the estimates fall close to the track and point as expected.

## 7. EVALUATION

In this section, we evaluate position and orientation accuracy in both typical usage conditions and in carefully controlled settings. We also evaluate the visible light communications channel for pure tones, Manchester-encoded data, and a hybrid of the two. Our experiments are carried out on a custom indoor positioning testbed.

### 7.1 Experimental Methodology

We integrate five LED landmarks, a smartphone, and a cloudlet server into an indoor positioning testbed, as Figure 8 shows. The LED landmarks are mounted on a height-adjustable pegboard and they form a  $71.1 \times 73.7$  cm rectangle with a center point. A complementary pegboard is affixed to floor and aligned using a laser sight and verified with a plumb-bob, creating a 3D grid with 2.54 cm resolution of known locations for our experimental evaluation. To isolate localization from communications performance, we set the transmitters to emit pure tones in the range of 2 kHz to 4 kHz, with 500 Hz separation, which ensures reliable communications (we also test communications performance separately). Using this testbed, we evaluate indoor positioning accuracy—both location and orientation—for a person, model train, and statically.

### 7.2 Realistic Positioning Performance

To evaluate the positioning accuracy of the Luxapose system under realistic usage conditions, we perform an experiment in which a person repeatedly walks under the indoor positioning testbed, from left to right at  $1$  m/s, as shown from the top view of the testbed in Figure 9b and side view in Figure 9e. The CDF of estimated location and orientation errors when the subject is under the landmarks (shaded) or outside the landmarks (unshaded) is shown in Figure 9d. When under the landmarks, our results show a median location error of 7 cm and orientation error of  $6^\circ$ , substantially better than when outside the landmarks, which exhibit substantially higher magnitude (and somewhat symmetric) location and orientation errors.

To evaluate the effect of controlled turning while under the landmarks, we place a phone on a model train running at  $6.75$  cm/s in an oval, as shown in Figure 9c. Most of the location samples fall on or within 10 cm of the track with the notable exception of when the phone is collinear with three of the transmitters, where the error increases to about 30 cm, though this is an artifact of the localization methodology and not the motion. When the speed of the train is doubled—to  $13.5$  cm/s—we find a visible increase in location and orientation errors, as shown in Figure 9f.



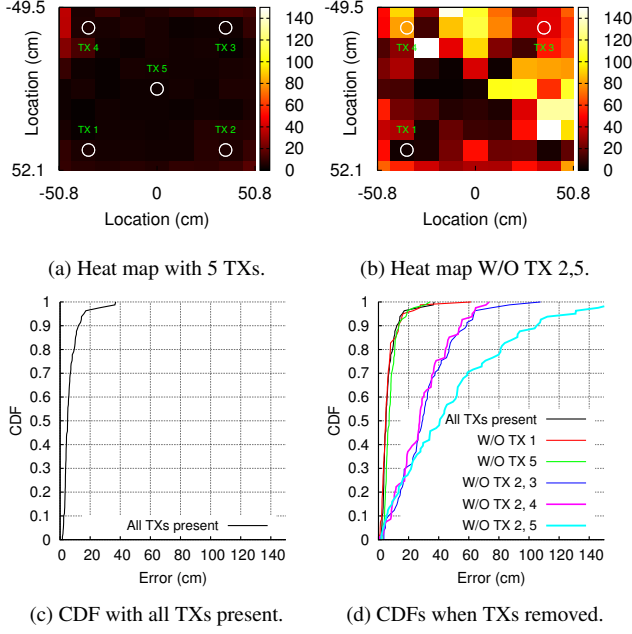


Figure 10: Localization accuracy at a fixed height (246 cm). (a) shows a heat map of error when all 5 transmitters are present in the image, and (c) shows a CDF of the error. (d) explores how the system degrades as transmitters are removed. Removing any one transmitter (corner or center) has minimal impact on location error, still remaining within 10 cm for ~90% of locations. Removing two transmitters (leaving only the minimum number of transmitters) raises error to 20–60 cm when corners are lost and as high as 120 cm when the center and a corner are lost. As shown in the heat map in (b), removing the center and corner generates the greatest errors as it creates sample points with both the largest minimum distance to any transmitter and the largest mean distance to all transmitters.

### 7.3 Controlled Positioning Accuracy

To evaluate the limits of positioning accuracy under controlled, static conditions, we take 81 pictures in a grid pattern across  $100 \times 100$  cm area 246 cm below the transmitters and perform localization. When all five transmitters are active, the average position error across all 81 locations is 7 cm, as shown in Figure 10a and Figure 10c. Removing any one transmitter, corner or center, yields very similar results to the five-transmitter case, as seen in the CDF in Figure 10d.

Removing two transmitters can be done in three ways: (i) removing two opposite corners, (ii) removing two transmitters from the same side, and (iii) removing one corner and the center. Performing localization requires three transmitters that form a triangle on the image plane, so (i) is not a viable option. Scenario (iii) introduces the largest error, captured in the heatmap in Figure 10b, with an average error as high as 50 cm in the corner underneath the missing transmitter. In the case of a missing side (ii), the area underneath the missing transmitters has an average error of only 29 cm. Figure 10d summarizes the results of the removing various transmitter subsets.

In our worst case results, on an unmodified smartphone we are able to achieve parity (~50 cm accuracy) with the results of systems such as Epsilon [13] that require dedicated receiver hardware in addition to the infrastructure costs of a localization system. However, with only one additional transmitter in sight, we are able to achieve an order of magnitude improvement in location accuracy.

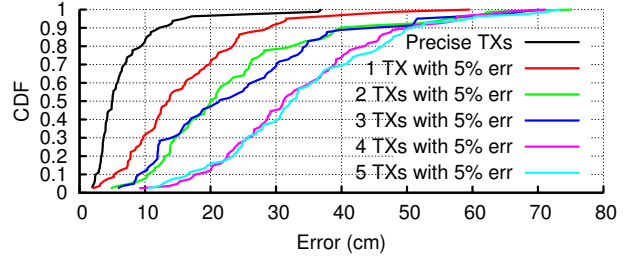


Figure 11: CDF of location error from a 5% error in absolute transmitter location under the same conditions as Figure 10a. This experiment simulates the effect of installation errors.

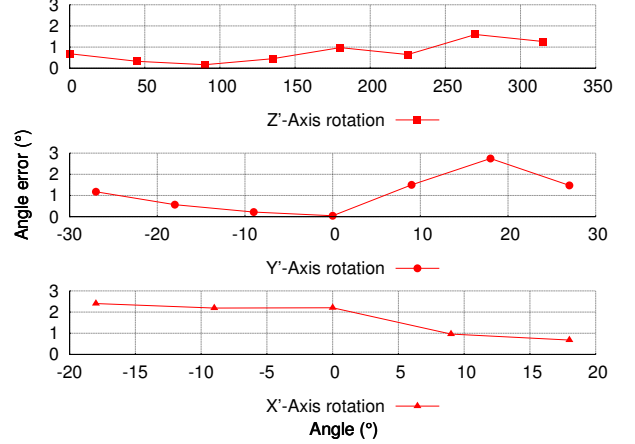


Figure 12: We rotate the mobile phone along axes parallel to the  $z'$ -,  $y'$ -, and  $x'$ -axis. Along the  $z'$ -axis, the mobile phone rotates  $45^\circ$  at a time and covers a full circle. Because of FoV constraints, the  $y'$ -axis rotation is limited to  $-27^\circ$  to  $27^\circ$  and the  $x'$ -axis is limited to  $-18^\circ$  to  $18^\circ$  with  $9^\circ$  increments. The experiments are conducted at a height of 240 cm. The angle error for all measurements falls within  $3^\circ$ .

Thus far, we have assumed the precise location of each transmitter is known. Figure 11 explores the effect of transmitter installation error on positioning by introducing a 5% error in 1–5 transmitter positions and re-running the experiment from Figure 10a. With 5% error in the origin of all five transmitters, our system has only a 30 cm 50th percentile error, which suggests some tolerance to installation-time measurement and calibration errors.

To evaluate the orientation error from localization, we rotate the phone along the  $x'$ ,  $y'$ , and  $z'$  axes. We compute the estimated rotation using our localization system and compare it to ground truth when the phone is placed 240 cm below the 5 transmitters. Figure 12 shows the orientation accuracy across all 3 rotation axes. The rotation errors fall within  $3^\circ$  in all measurements.

### 7.4 Frequency-Based Identification

We evaluate two frequency decoders and find that the FFT is more robust, but edge-detection gives better results when it succeeds.

**Rx Frequency Error vs Tx Frequency.** Figure 13 sweeps the transmit frequency from 1 to 10 kHz in 500 Hz steps and evaluates the ability of both the FFT and edge detector to correctly identify the transmitted frequency. The edge detector with  $1/16667$  s exposure performs best until 7 kHz when the edges can no longer be detected and it fails completely. The FFT detector cannot detect the frequency as precisely, but can decode a wider range of frequencies.

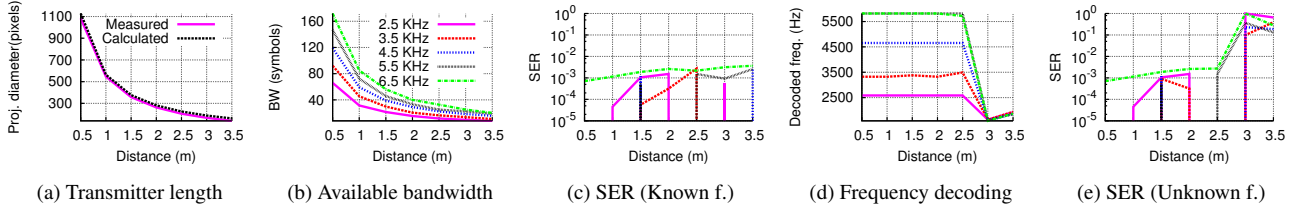


Figure 15: Examining the decodability of Manchester data across various transmit frequencies and distances. Figures (b) through (e) share the same legends. The transmitted frequencies are 2.5 KHz, 3.5 KHz, 4.5 KHz, 5.5 KHz and 6.5 KHz.

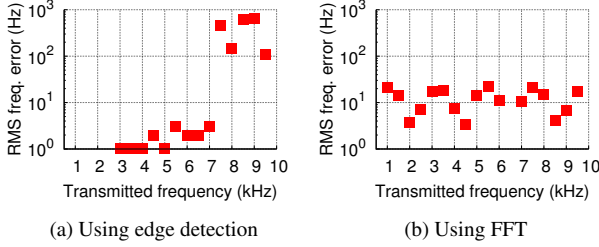


Figure 13: Frequency recovery at 0.2 m,  $1/16667$  s, ISO 100. The edge detector performs better until  $\sim 7$  kHz when quantization causes it to fail completely. The FFT method has lower resolution but can decode a wider frequency range.

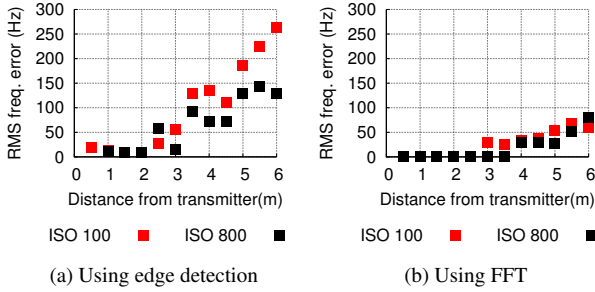


Figure 14: As distance grows, the light intensity and area fall super-linearly. Using a higher ISO amplifies what little light is captured, enhancing frequency recoverability. We transmit a 1 kHz frequency on a commercial LED and find that the decoded frequency error remains under 100 Hz for distances up to 6 m from the transmitter.

**Rx Frequency Error vs Tx Distance.** As the distance between the transmitter and phone increases, the received energy at each pixel drops due to line of sight path loss [8]. The area of the transmitter projected onto the imager plane also decreases. These factors reduce the ability to decode information. In Figure 14 we use a 10 cm diameter 14 W Commercial Electric can light to explore the impact of distance on our ability to recover frequency, and the effect of varying the ISO to attempt to compensate for the lower received power. As intensity fades, the edge detection cannot reliably detect edges and it fails. The FFT method is more robust to this failure, as it is able to better take advantage of pixels with medium intensity.

**The Importance of Frequency Channels.** Human constraints and optics constraints limit our bandwidth to 1~7 kHz. With an effective resolution of 200 Hz, the FFT decoder can only identify about 30 channels, and thus can only label 30 unique transmitters. The finer 50 Hz resolution of the edge detector allows for about 120 channels. A typical warehouse-style store, however, can easily have over 1,000 lights. We explore techniques for more efficiently using this limited set of frequency channels in Section 8.

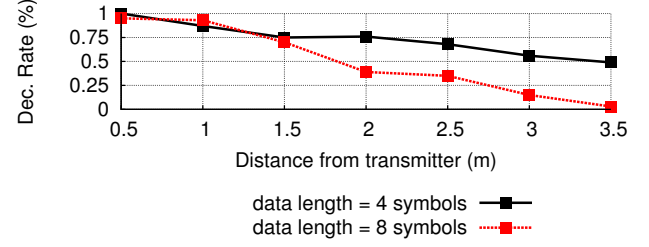


Figure 16: Hybrid decoding is able to better tolerate the frequency quantization ambiguity than pure Manchester. Shorter data has a higher probability of being correctly decoded at long distances.

## 7.5 Decoding Manchester Data

The relative size of a transmitter in a captured image dominates data decodability. If the physical width of a transmitter is  $A$  and the distance from the imager is  $D$ , the width of the transmitter on the image is  $A/D \times Z_f$  pixels. Figure 15a shows measured width in pixels and theoretical values at different distances. Figure 15b shows the effect on the maximum theoretical bandwidth when using Manchester encoding for various frequencies. Figure 15c finds that if the transmitter frequency is known, the symbol error rate (SER) is  $\sim 10^{-3}$ . Our sweeping match filter is able to detect frequency until a quantization cutoff, as Figure 15d shows. When the frequency is not known a priori, Figure 15e shows that the SER correlates strongly with the ability to decode frequency.

## 7.6 Decoding Hybrid Data

Hybrid decoding first decodes the pure tone frequency and then is able to use the known frequency to improve its ability to decode the data. As distance increases, the probability of capturing the data segment in the limited transmitter area falls, thus Figure 16 finds that shorter messages are more robust to large distances.

## 8. DISCUSSION

In this section, we discuss some limitations of our current system and potential directions for future work.

**Deployment Considerations.** In real settings, all LED locations must be known, although only the relative distances between closely located LEDs must be known with high accuracy. Although not trivial, it does not seem difficult to ensure that this condition holds. We have deployed a grid of sixteen luminaires in our lab, and we analyze the effect of location errors on localization accuracy in Section 7.3. We note that almost any localization system must know the anchor locations. In a practical setting, this would be done, presumably, with the aid of blueprints and a laser rangefinder.

**Usability.** Our system targets an active user, so that the front-facing camera naturally observes the ceiling during use. Passive localization (e.g. while the phone is in a pocket) is out of scope.

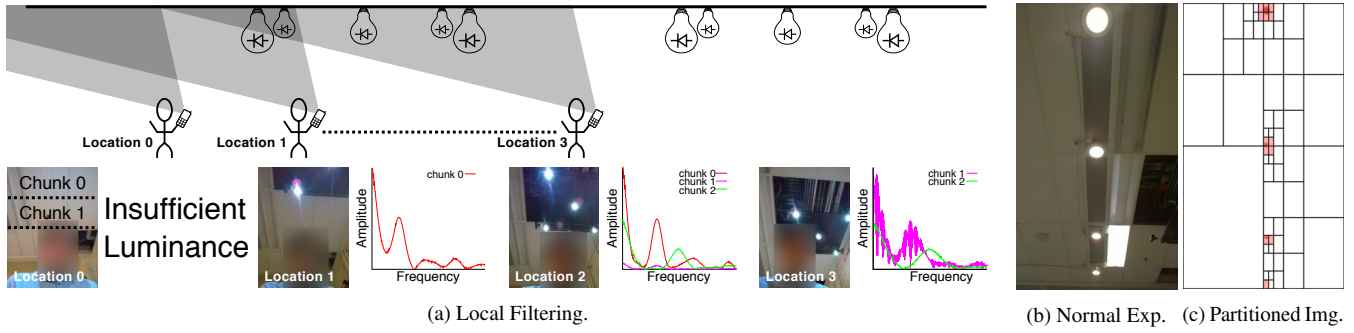


Figure 17: (a) Local filtering. In this experiment, we walk under our testbed, capturing images at about 1 fps. We divide each frame into 8 “chunks” and run an FFT along the center row of pixels for each chunk. The FFTs of non-negligible chunks are presented next to each image. At each location, we also capture an image taken with traditional exposure and film speed settings to help visualize the experiment; the FFTs are performed on images captured with  $\frac{1}{16667}$  s exposure on ISO 100 film. (b)-(c), Recursive Searching. The image is partitioned and each segment is quickly scanned by taking an FFT of the column sum of each segment. Segments with no peaks are discarded and segments with interesting peaks are recursed into until the minimum decodable transmitter size ( $\sim 60$  pixels) is found.

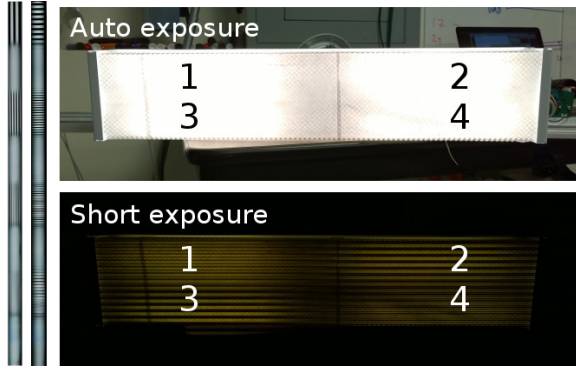


Figure 18: (left) The same LED tube imaged twice at  $90^\circ$  rotations shows how multiple beacons can be supported in a single lamp. (right) A single fixture can support multiple LED drivers (four here). An image capturing only this fixture could be used to localize.

**Distance.** Distance is the major limitation for our system. Received signal and projected image size are strongly affected by distance. We find that a 60 pixel projection is roughly the lower bound for reliable frequency decoding. However, as camera resolutions increase, our usable distance will improve.

**Local Filtering.** Not all images capture enough transmitters to successfully localize. It would be desirable to perform some local filtering to discard images that would not be useful for positioning, thus avoiding the cost of transferring undecodable images to the cloud. We explore one such possibility in Figure 17a. The phone selects a sampling of image rows and performs an FFT, searching for the presence of high frequency components. This fast and simple algorithm rejects many images that would not have decoded.

**Alternative Image Processing.** Building on the local filtering concept, another possible approach for locating transmitters in the captured image, like Figure 17b, may be a divide and conquer technique, as shown in Figure 17c. As this algorithm already partitions the image into bins with FFTs, it is also well suited to solve the problem of separating non-disjoint transmitters. If only the filtered chunks are processed, the processing load is substantially reduced—from 33 MP to 0.42 MP ( $13 \text{ chunks} \times (33/1024) \text{ MP/chunk}$ ), dramatically reducing image transfer time to the cloudlet, and the processing time on the cloudlet. This approach may even allow positioning to occur entirely on the smartphone.

**Fixture Flexibility.** Our system requires that at least three transmitters are captured and decoded. Many LED fixtures, such as office fluorescent T8 tube replacements, are actually multiple LED transmitters in a single fixture. Figure 18 shows how a single LED tube can transmit multiple beacons (left) and how a fixture with multiple tubes could support the non-collinear transmitter requirement (right). Localizing with this fixture would require improving our image processing, which currently assumes disjoint, circular transmitters.

**Interference.** Since only the direct line-of-sight path is captured by our short exposure time, there is little danger from interference regardless of transmitter density (for two transmitters’ projections to alias, the pixel quantization must be so poor that they are only mapping to a few pixels and are undecodable anyway).

**Limited Frequency Channels.** Our system has a limited set (up to 120) of frequencies with which to label each transmitter. One method to increase the number of labels would be to have each transmitter alternate between two frequencies ( $\binom{120}{2} = 7140$ ). Reliably and accurately estimating inter-frame motion (e.g. using the accelerometer and gyroscope), however, could prove difficult, making it difficult to match transmitter projections across frames.

A simpler approach that still requires only a single image is to simply re-use labels and leverage transmitter adjacency relationships. As our system captures contiguous images and requires at least three landmarks to localize, the adjacency relationships between lights form another constraint that can uniquely identify transmitters. Actually identifying transmitters with this system is surprisingly simple. For each frequency observed, consider all possible transmitter locations and compute the total inter-transmitter distance. The set of transmitters that minimizes this distance are the actual transmitters. This transmitter labeling technique is the same minimization procedure already used by the processing for AoA estimation.

**Dimmable LEDs.** Dimming is a requirement in 802.15.7. LEDs can be dimmed by either reducing their current or using PWM. As PWM dimming may affect our transmitted signal, we briefly explore its impact by PWM dimming an LED using a frequency higher than the phone’s scan rate (we use 1 MHz, 10% duty cycle). We find that it does not affect our ability to decode data.

**Privacy.** Our design does not require interaction with the local environment. Luminaires are unidirectional beacons and image capture emits no signals. If needed, the lookup table can be acquired once out of band, and processing could be done either on the phone or a user’s private cloud. A user can thus acquire location estimates without sharing any location information with any other entity.



## 9. CONCLUSIONS

Accurate indoor positioning has been called a “grand challenge” for computing. In this paper, we take a small step toward addressing this challenge by showing how unmodified smartphones and slightly-modified LED lighting can support accurate indoor positioning with higher accuracy than prior work. Our results show that it is possible to achieve decimeter location error and  $3^\circ$  orientation error by simply walking under an overhead LED light while using one’s smartphone. When used in typical retail settings with overhead lighting, this allows a user to be accurately localized every few meters, perhaps with dead reckoning filling in the gaps. Although our current approach has many drawbacks, none appear to be fundamental. Having demonstrated the viability of the basic approach, future work could explore the rolling shutter channel, improve channel capacity, increase image processing performance, and reduce positioning error.

## 10. ACKNOWLEDGMENTS

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA. This research was conducted with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. This material is based upon work partially supported by the National Science Foundation under grants CNS-0964120, CNS-1111541, and CNS-1350967, and generous gifts from Intel, Qualcomm, and Texas Instruments.

## 11. REFERENCES

- [1] J. Armstrong, Y. A. Sekercioglu, and A. Neild. Visible light positioning: A roadmap for international standardization. *IEEE Communications Magazine*, 51(12), 2013.
- [2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. 19th Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM '00)*, volume 2, 2000.
- [3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [4] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. FM-based indoor localization. In *Proc. of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, 2012.
- [5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proc. of the 16th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '10)*, 2010.
- [6] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In *Proc. of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, 2011.
- [7] P. Connolly and D. Boone. Indoor location in retail: Where is the money? ABI Research Report, 2013.
- [8] K. Cui, G. Chen, Z. Xu, and R. D. Roberts. Line-of-sight visible light communication system design and demonstration. *7th IEEE IET International Symposium on Communication Systems Networks and Digital Signal Processing*, 2010.
- [9] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas. Using a CMOS camera sensor for visible light communication. In *IEEE Globecom Workshops*, 2012.
- [10] M. C. Dean. *Bearings-Only Localization and Mapping*. PhD thesis, Carnegie Mellon University, 2005.
- [11] A. Jovicic, J. Li, and T. Richardson. Visible light communication: Opportunities, challenges and the path to market. *IEEE Communications Magazine*, 51(12), 2013.
- [12] T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using LED lights. *IEEE Transactions on Consumer Electronics*, 50(1), 2004.
- [13] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *Proc. of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI '14)*, 2014.
- [14] K. Lorincz and M. Welsh. Motetrack: A robust, decentralized approach to RF-based location tracking. *Personal Ubiquitous Computing*, 11(6), Aug. 2007.
- [15] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy. Precise indoor localization using smart phones. In *Proc. of the ACM International Conference on Multimedia*, 2010.
- [16] Microsoft. PhotoCaptureDevice class. <http://msdn.microsoft.com/en-us/library/windowsphone/develop/windows.phone.media.capture.photocapturedevice>.
- [17] Nokia. Camera explorer. <https://github.com/nokia-developer/camera-explorer>, 2013.
- [18] N. Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1), 1979.
- [19] G. B. Prince and T. D. Little. A two phase hybrid RSS/AoA algorithm for indoor device localization using visible light. In *IEEE Global Communication Conference (GLOBECOM '12)*, 2012.
- [20] M. S. Rahman, M. M. Haque, and K.-D. Kim. Indoor positioning by led visible light communication and image sensors. *International Journal of Electrical and Computer Engineering (IJECE)*, 1(2), 2011.
- [21] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proc. of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '14)*, 2014.
- [22] S. Rajagopal, R. D. Roberts, and S.-K. Lim. IEEE 802.15.7 visible light communication: Modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3), 2012.
- [23] A. Richardson, J. Strom, and E. Olson. AprilCal: Assisted and repeatable camera calibration. In *Proc. of International Conference on Intelligent Robots and Systems (IROS '13)*, 2013.
- [24] M. Sakata, Y. Yasumuro, M. Imura, Y. Manabe, and K. Chihara. Location system for indoor wearable PC users. In *Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, 2003.
- [25] S. Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1), 1985.
- [26] J. Tan and N. Narendran. A driving scheme to reduce AC LED flicker. *Optical Engineering*, 2013.
- [27] J. Xiong and K. Jamieson. ArrayTrack: A fine-grained indoor location system. In *Proc. of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13)*, 2013.
- [28] S.-H. Yang, E.-M. Jeong, D.-R. Kim, H.-S. Kim, Y.-H. Son, and S.-K. Han. Indoor three-dimensional location estimation based on LED visible light communication. *Electronics Letters*, 49(1), January 2013.
- [29] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proc. of the 18th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, 2012.
- [30] M. Yoshino, S. Haruyama, and M. Nakagawa. High-accuracy positioning system using visible LED lights and image sensor. In *IEEE Radio and Wireless Symposium (RWS '08)*, 2008.
- [31] M. A. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proc. of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, 2005.
- [32] Z. Zhou, M. Kavehrad, and P. Deng. Indoor positioning algorithm using light-emitting diode visible light communications. *Optical Engineering*, 51(8), 2012.