Enabling High-Precision Visible Light Localization in Today's Buildings

Shilin Zhu and Xinyu Zhang
University of Wisconsin-Madison
{szhu,xyzhang}@ece.wisc.edu

ABSTRACT

For over one decade, research in visible light positioning has focused on using modulated LEDs as location landmarks. But the need for specialized LED fixtures, and the associated retrofitting cost, has been hindering the adoption of VLP. In this paper, we forgo this approach and design iLAMP to enable reliable, high-precision VLP using conventional LEDs and fluorescent lamps inside today's buildings. Our key observation is that these lamps intrinsically possess hidden visual features, which are imperceptible to human eyes, but can be extracted by capturing and processing the lamps' images using a computational imaging framework. Simply using commodity smartphones' front cameras, our approach can identify lamps within a building with close to 100% accuracy. Furthermore, we develop a geometrical model which combines the camera image with gyroscope/accelerometer output, to estimate a smartphone's 3D location and heading direction relative to each lamp landmark. Our field tests demonstrate a mean localization (heading) precision of 3 cm (2.6°) and 90-percentile 3.5 cm (2.8°), even if a single lamp falls in the camera's field of view.

1. INTRODUCTION

Over the past decade, there has been a concerted research effort in developing an accurate, reliable, and ready-to-use indoor localization system for smartphones. Such a system can enable a multitude of location-based services. Some of the use cases include: precise navigation to rooms/items of interest in office buildings, museums, airports, and shopping centers; targeted advertisement, product recommendation and coupon delivery in retail stores; consumer analytics through aggregated foot-traffic patterns and dwell time; multi-players augmented-reality games, *etc.* To unleash these services and trigger wide adoption, the localization technology must provide the business operators or customers a compelling quality of experience, specifically in terms of: high precision, high robustness, low cost (in terms of location sensor hardware and infrastructure maintenance), and mobile friendliness (low latency and low power consumption).

Despite a wide spectrum of indoor localization technologies, there has been very limited adoption into real-world scenarios, mainly because of the challenges in simultaneously satisfying the above key metrics. Mainstream approaches, especially those based on RF signals, focused intensively on improving the location precision. In particular, recent multi-antenna based solutions have achieved a median precision of decimeters [1–3]. But they fall short of robustness in real building environment—the 90% location error often reaches 3 to 10 meters [1–3], which may jeopardize user experience. The root cause lies in the elusive nature of the wireless channel. High precision RF localization relies on phase or received signal strength (RSS), metrics that can be easily affected by ambient multipath reflections. In effect, human body blockage, reflection and even hand gestures can significantly disturb such wireless channel profiles [4, 5].

As an alternative modality, visible light positioning (VLP) holds potential to overcome the instability owing to the almost multipath-free propagation. VLP can achieve decimeter to centimeter precision, using specialized "beaconing LEDs" as location landmarks, and photodiodes [6–8] or smartphone cameras [9–11] as location sensors. However, deploying such VLP systems at building scale entails changing the fixtures/bulbs, at substantial retrofitting cost. To date, fluorescent lights (FLs) occupy 85% of the commercial buildings in the US [12]. Even the *basic LEDs* only account for 12%, and will take another 10 to 15 years to dominate the market [12], not to mention the smart beaconing LEDs. A recent solution, LiTell [13], enabled low-cost VLP by sensing FLs' inherent flickering frequencies. However, these frequency features are extremely weak. They are detectable only on FLs, under low ceilings (< 2.5 m) and with high-resolution back cameras. Moreover, such features can discriminate individual lights with only 60% accuracy, which hampers reliability.

In this paper, we propose a novel VLP system, called iLAMP, to fill the missing spot that meets the multi-faceted challenges. iLAMP uses a smartphone camera to discriminate existing FLs and LEDs, based on visual features extracted from a computational imaging framework. Furthermore, it can reliably derive the smartphone's heading direction and 3D location at centimeter precision, even when a single light landmark is visible. Leveraging the ubiquitous lighting infrastructure, iLAMP can bring highly reliable and accurate indoor localization to today's buildings, at no extra hardware cost.

The key challenge for iLAMP lies in discriminating the incumbent lights, which have no beacon-generation hardware and often come from the same model when deployed in a building. Nonetheless, iLAMP's computational imaging solution can extrapolate hidden features from images of the lights. It stores these features as unique signatures in a server database during the setup phase, and uses simple feature matching mechanism to derive a light's identity during the run-time localization phase. The *main feature* that iLAMP harnesses is the *spatial radiance pattern* (SRP), defined as the radiance intensity distribution across a light's

MobiSys '17, June 19–23, 2017, Niagara Falls, NY, USA.

 ISBN 978-1-4503-4928-4/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3081333.3081335

body. This SRP feature is resilient to the camera's viewing angle/distance, and highly diverse among lights due to inevitable manufacturing variations. In addition, iLAMP employs two sets of *assistant features*, derived from the smartphone's ambient light sensor (ALS) and camera RGB output, respectively, as coarse-grained pre-filters to curtail the computational cost in matching the main feature. Low computational load in turn translates into low response latency.

Once a light landmark is identified, a camera-based VLP system can employ the photogrammetry technique in computer vision [14] to derive a smartphone's physical location relative to the landmarks. State-of-the-art VLP solutions [6, 10] often require 3+ lights for triangulation, but smartphone cameras typically have a narrow FoV of only around 60° and can hardly capture more than one lights simultaneously in practical buildings. In contrast, iLAMP employs a *sensor-assisted photogrammetry* mechanism, which harnesses the inherent spatial heterogeneity of the radiance pattern, to estimate the phone's azimuth orientation, and subsequently its 3D location, even if a single light is visible.

Existing camera-based VLP [9,10,13] focused on full-operation mode, with camera always on, consuming substantial power (2 to 3 W [10, 13]). Observing that ceiling luminaries tend to be scattered, iLAMP turns on a camera only when it gauges that a light falls in its FoV. The key idea is to use the smartphone's ALS as a gating device, and derive the intensity correlation between the ALS and camera through simple offline calibration. With this smart camera scheduler, we can duty cycle the operation and substantially reduce the power consumption.

We have implemented iLAMP as an Android application connected to a backend database server, and evaluated its performance against the aforementioned challenging requirements. Our experiments in real-world buildings show that iLAMP achieves around 95% accuracy in identifying different ceiling lights even when using radiance pattern alone, and close to 100% when combining the two assistant features. iLAMP achieves a mean location precision of 3.2 cm and 90-percentile of 3.5 cm under a single light, and even higher when multiple lights are visible. iLAMP also estimates the phone's azimuth (heading direction) with a small error of 2.6° and 90-percentile of 2.8°. More importantly, the performance remains highly stable under practical disturbing factors, such as random variation of phone orientation and phone-to-ceiling distance. Even under extremely sparse light deployment, iLAMP can still be combined with motion-sensor based dead-reckoning mechanisms, and maintain a few decimeters of location precision. In addition, iLAMP is efficient: it has a low end-to-end latency of 400-700 ms, and total power consumption of 927 mW on a smartphone (less than 1/2 compared with LiTell [13] or Luxapose [10]).

Despite more than one decade of research [15], VLP has not been widely adopted. The main contribution of iLAMP is to fill the sweet spot between accuracy, cost and reliability, and enable a VLP system that is immediately usable in today's buildings. More specifically,

(i) We design novel computational imaging mechanisms to extrapolate intrinsic visual features from incumbent FLs and LEDs, allowing them to be distinguishable at no extra hardware cost. We further introduce simple feature compression and matching schemes to make the light identification computationally efficient and robust to image distortion.

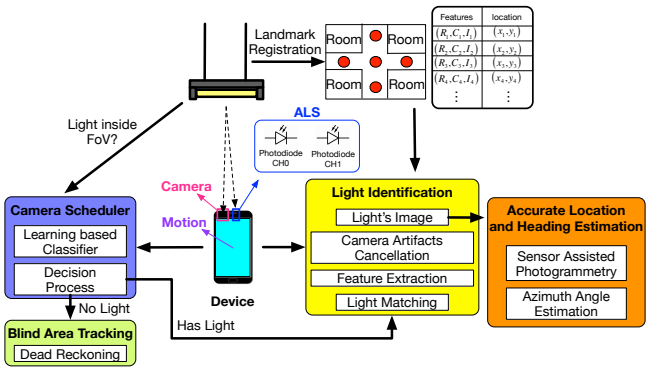

Figure 1: iLAMP system workflow.

(ii) We introduce a sensor assisted photogrammetry technique which can precisely locate a smartphone's 3D position and heading direction, even when a single light landmark is available.

(iii) We design a camera scheduling mechanism that duty cycles the power hungry camera based on its correlation with the low-power ALS. The mechanism can be generalized to all camera-based VLP systems.

2. iLAMP OVERVIEW

Main challenges and design goals. iLAMP builds its components around four major design goals: *(i)* Reliably discriminating existing ceiling lights with close to 0 confusion probability under a wide range of usage scenarios (various light models, ceiling heights, phone orientation, sunlight interference, *etc.*). *(ii)* Accurately estimating heading direction and 3D location with centimeter precision even under a single light. *(iii)* High computational efficiency, and real-time response to localization requests with sub-second end-to-end latency. *(iv)* High energy efficiency for continuous location tracking. Finally, iLAMP aims for an immediately usable localization system that is compatible with typical smartphone hardware.

System workflow. *iLAMP* comprises three main modules: light identification, phone location/heading estimation, and camera scheduling. Fig. 1 illustrates their work flow.

To bootstrap the system, we need to take a *benchmark image* for each light, extract its visual features, and register the ⟨*feature vector, location*⟩ pair in a server database. This *landmark registration* procedure only needs to be done once for each light. The light fixtures' locations are usually known at installation time; even manual survey of the locations is simple as the lights tend to be deployed regularly over space. Registering the lights' locations is also the minimal bootstrapping effort needed for all other VLP systems.

At run-time, the smartphone takes an image, preprocesses it to eliminate camera artifacts and remove background pixels (Sec. 3.1). It then extracts the *main feature* (*i.e.*, spatial radiance pattern) from pixels representing the light's body, and *compresses* the feature into a small-sized array (Sec. 3.1). Meanwhile, iLAMP computes two *assistant features*: the color pattern based on the image's RGB values, and the infrared to visible light intensity ratio (I2V ratio) based on the smartphone's ambient light sensor (ALS) (Sec. 4.1). These three features form a vector and are sent to the server. The server runs a hierarchical *light identification* algorithm: It uses the assistant features to narrow down the search space, and then looks up the database to identify the landmark whose main feature best matches the current light (Sec. 3.2).

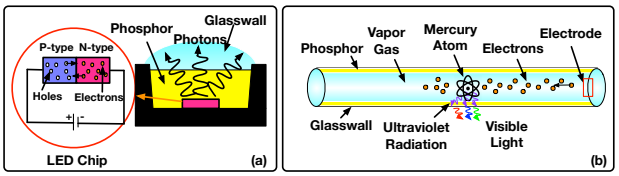

Figure 2: Physical principle of visible light emission of LED and fluorescent.

Once the light landmark is identified, iLAMP uses its *sensor assisted photogrammetry* to compute the phone's 3D location relative to the landmark, based on the camera image and the phone's gravity sensor output (Sec. 5.1). This approach also estimates the azimuth (heading direction) of the phone to substitute the notoriously inaccurate smartphone compass [16] (Sec. 5.2).

Both the light identification and location/heading estimation modules need the camera image as input. To curtail the camera's power consumption, iLAMP executes its *camera scheduler* to adaptively turn on the camera (Sec. 4.2). To ensure reliability of light identification, iLAMP only proceeds with those images containing at least one full light. When no light is visible, iLAMP uses the conventional motion-sensor based dead-reckoning method [17] to keep track of the phone's movement.

3. DISCRIMINATING LIGHTS USING HIDDEN VISUAL FEATURES

3.1 The Hidden Fingerprints of Incumbent Lights

3.1.1 Understanding the Optical Properties of LEDs and Fluorescent Lights

To understand the origin of the visual features in conventional FLs and LEDs, we first explain their working principles (Fig. 2). *Light-Emitting Diodes (LEDs)* generate light through a semiconductor chip (a *p-n* junction diode). Under a suitable voltage, the electrons within the chip will fall into a lower energy level when meeting a hole, emitting energy in the form of photons. An LED lamp typically comprises multiple LED chips, and integrates with optical lenses/glass-walls to reshape its radiation pattern. The wavelength of the light emitted, and thus its color, depends on the materials forming the LED chip. Most commercial white LEDs are formed by coating blue-color (or ultraviolet/RGB) LED chips with multiple phosphor layers of different colors [18,19]. Due to inevitable manufacturing variations, *e.g.*, phosphor thickness/composition and non-uniformity of the glass-wall, different areas of an LED lamp may manifest different optical properties (radiant flux, color temperature, *etc.*).

Fluorescent lights (FLs) use an electronic ballast to excite mercury vapor inside a lamp tube, which produces short-wave ultraviolet light that then causes a phosphor coating to radiate visible light. In FLs, characteristics of electrons and Mercury atoms traveled inside vapor gas can bring different radiance power and spatial pattern across the light tube's body. Non-uniformity of the phosphor coating and the glass wall further varies the emission characteristics, even among FLs of the same model.

Although the variations of optical properties are invisible to human eyes, they can be revealed by the computational imaging mechanisms in iLAMP, which we detail below.

3.1.2 Extracting a Lamp's Spatial Radiance Pattern

We now introduce how iLAMP extracts the hidden visual features from a light (lamp), based on the principle of camera image formation (Fig. 3).

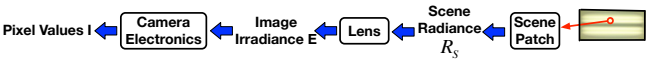

Figure 3: Principle of image formation.

Radiance characterizes the radiation property of a surface patch (either on a light source or a reflecting surface). It corresponds roughly to the brightness, and is defined as the amount of light radiated from the surface patch per solid angle per unit area (expressed in Watts per m^2 per steradian) [20]. Radiance is an intrinsic property of the light emitter. By definition, it is independent of viewing angle. It is also independent of distance, because the sampled surface area increases quadratically with the viewing distance, canceling the inverse-square path loss of optical signals [20].

On the other hand, a camera quantizes the *irradiance* E, defined as the light power per unit area (W/m^2) captured on its image sensor, and bears the following relation with radiance R_s:

$$E = \frac{\pi}{4} R_s \frac{l_0^2}{f^2} \cos\alpha^4 \quad (1)$$

where l_0 and f denote the camera lens' diameter and focal length, respectively. α is the incidental angle from the emitter to the camera. If emitter tilts away from light, the same amount of light strikes bigger sensor area which decreases E. But in practice, most cameras have a small FoV of around 60°, corresponding to $\alpha = 30^\circ$ and $\cos\alpha^4 = 0.997 \approx 1$.

Therefore, *the irradiance of a camera image only depends on the intrinsic radiance property of the scene, and is independent of the scene-to-camera distance/angle.* This should not be confused with the *light intensity*, which is known to be affected by the distance/angle factors.

Ultimately, the camera electronics post-process the image and convert the matrix of irradiance values (representing the whole scene) into a matrix of *pixel values*. This conversion follows a *camera response function* which should ideally be linear. Different camera models often use different gamma correction into their image processing pipeline which may make the response function non-linear. Nonetheless, the linearity can be restored by a standard one-time camera response calibration procedure [21], which only requires using the camera to capture the same scene under multiple exposure settings.

In iLAMP, we use the *spatial radiance pattern* (SRP) to characterize a lamp, which is defined as the distribution of radiance values across the lamp's body. Owing to the aforementioned linear relation, we can use the 2D matrix of pixel values as feature, which is independent of orientation/distance just like radiance itself. Although SRP is invisible to human eyes, it can be revealed by looking into values of all pixels of two-dimensional image. Fig. 4 plots the SRP of a set of ceiling-mounted LEDs (FLs) of the same model, which clearly shows distinguishable spatial patterns.

Since an image can contain millions of pixels, directly using the pixel-by-pixel SRP is computationally intensive. We thus employ a simple *feature compression* mechanism that abstracts the 2D SRP matrix into a small array. Consider the most commonly adopted linear light fixtures in commercial buildings. iLAMP first fits the image of such a light fixture in the Cartesian coordination. It then computes each row's and column's average radiance and standard deviation (*std.*), which compactly represents the spatial distribution

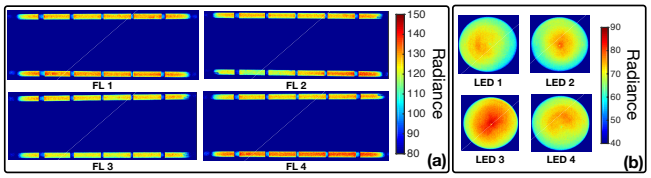

Figure 4: Example spatial radiance pattern of (a) 4 FLs and (b) 4 LEDs of the same model. The notches on the FLs are due to the cover structure.

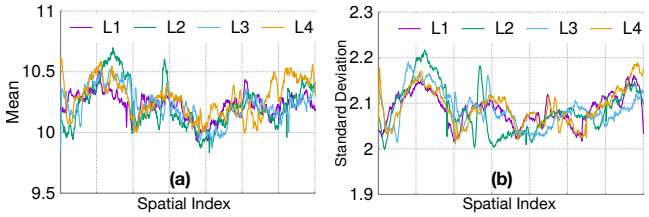

Figure 5: SRP (mean and *std*) of 4 FLs after compression.

of radiance. Therefore, we can reduce a radiance pattern of $M \times N$ pixels to a vector of size $M+N$, where M and N represents the number of rows and columns, respectively. For round shaped lamps, we use the polar coordinates, *i.e.*, radiation distribution along radial and angular directions *w.r.t.* the center of the lamp. Fig. 5 plots the compressed SRP of 4 FLs of the same model, which clearly demonstrate distinguishable patterns. Note that the small-scale fluctuation and variation are quite different.

3.1.3 Overcoming Camera Artifacts

We now describe the preprocessing needed before feeding a camera image into the aforementioned SRP extraction. Fig. 6 illustrates these operations.

Contour extraction. By default, iLAMP sets the camera's exposure time to its minimum to maximize the contrast, which renders the background pixels almost dark. Given an image, iLAMP runs a contour extraction to obtain the pixels belonging to the lamp's body, and remove the background. Direct contour extraction needs to involve millions of pixels. To reduce the computational cost, we first subsample the image, and then run the classical edge-detection based contour extraction algorithm [22] on this sub-sampled image. To reduce the dark noises within the image, we configure the ISO to the minimum value. Sub-sampling does not corrupt the contour extraction since most of the lights inside modern buildings have regular shapes. We eventually scale up the extracted contour to fit the original image and use this contour as a mask to filter out background pixels. Given the contour, we then run a shape similarity check against a benchmark image in the database to determine if a full lamp is captured.

Compensating the Color Filter Array (CFA). Cameras commonly use a two-dimensional CFA to collect photons. Each array element corresponds to a particular R, G, or B pixel sensor, interleaved across rows and columns. The heterogeneous distribution of RGB sensors may distort the SRP, because different color pixels may scale the radiance differently, depending on the color spectrum of the light emission. We thus normalize the RAW value of each pixel output, by the average value of all pixels with the same color, so as to compensate for the CFA distortion caused by existence of different colors.

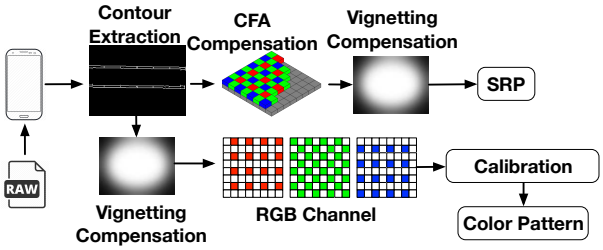

Figure 6: Image preprocessing to compensate for undesired effect of camera.

Compenstating the vignetting effect. Vignetting is a natural artifact of the imperfect camera lens, which causes lower brightness at peripheral pixels than at the center of the image. To prevent vignetting from distorting the SRP, we run a one-time camera calibration following [23], which fits each row/column of an image with a 6-th order polynomial curve. At run-time, we normalize the pixels along each row/column by its corresponding fitting curve.

Compensating the heterogeneity among camera models. Different smartphones may have different camera models with different color and intensity responses. We thus normalize the camera's average RGB and luminance values by a fixed proportion (corresponding to this specific camera) to match the standard ones stored in the database, for all the images taken by this camera. Note that different camera models may have different proportions for normalization and this one-time calibration has low overhead for each camera model. This ensures the light identification works even if the features in the server database are captured using a different camera model.

3.2 Robust SRP Matching Under Distance/ Orientation Distortion

After extracting the SRP from the current light's image, iLAMP needs to find the best matching light inside the server database. Using the Euclidean distance as a matching metric is feasible but highly vulnerable to image distortions. Depending on the phone's holding position, the run-time phone-to-ceiling distance may differ from that when the ground-truth SRPs are created for the database. Longer distances lead to fewer pixels that portrait the light, hence fewer elements in the SRP array. The phone's orientation change may also cause phase shift and deformation effect on the SRP.

DTW formulation. In iLAMP, we use the dynamic time warping (DTW) to deal with such distance/orientation distortions. DTW has been widely adopted in measuring similarity between two time series (*e.g.*, sequences of speech), owing to its robustness against signal compressing, stretching, and phase shift. Given an SRP array of length C obtained by camera and a candidate array of length D in our database, DTW first constructs an $C \times D$ matrix, with each element (c, d) being the distance between corresponding elements c and d in the two arrays. To find the best way to align these two arrays, DTW needs to retrieve a path through the matrix with the minimum cumulative distance, referred to as the warping cost or *DTW distance*. When the two arrays are exactly the same, the path simply traverses the diagonal of the matrix. In more general cases, DTW can be solved using a known dynamic programming formulation [24].

iLAMP uses the DTW distance to measure the similarity between each pair of SRP arrays. Recall that an SRP array

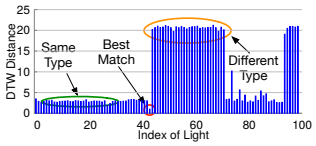

Figure 7: Uniqueness of main feature.

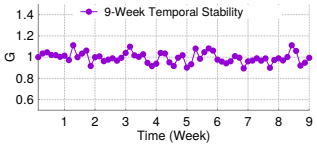

Figure 8: Stability of main feature with time.

concatenates two sequences: the mean values ($\boldsymbol{S_1}$) and *std.* values ($\boldsymbol{S_1}$), respectively, which may have different ranges. We thus compute the total DTW distance as: $DTW(\boldsymbol{S_1}) + DTW(\boldsymbol{S_2}) \cdot mean(\boldsymbol{S_1})/mean(\boldsymbol{S_2})$, where the scaling is used to make the two distances addable. Finally, the candidate with the minimum total DTW distance will be used as the best match.

Microbenchmark verification of light identification. We now use the DTW metric to benchmark the *uniqueness* and *temporal stability* of the SRP feature. Fig. 7 plots an example of DTW distances among 100 lights in our office building using one light. All DTW distances have been normalized by the minimum one (*i.e.*, DTW_{min}). We observe that the DTW distance between images of the same light is obviously the smallest, implying the SRP feature is highly unique. The DTW distance with others is much larger, due to manufacturing variations among the same light model, and physical appearances (cover, shape, *etc.*) among different models.

To simplify the evaluation of uniqueness, we introduce a metric called ***normalized tolerance gap***, defined as

$$G = \frac{DTW^c_{smin} - DTW^c_{min}}{DTW^n_{smin} - DTW^n_{min}} \quad (2)$$

where DTW^c_{min} denotes DTW distance between an image and the benchmark image of the same light (in the database). DTW^c_{smin} denotes the minimum DTW distance with all other lights in the database. Here the superscript c represents variable conditions such as time, distance, orientation *etc.* DTW^n_{smin} and DTW^n_{min} represent counterpart definitions for an image taken under similar condition as when the database benchmark (normal condition) was created.

This G factor essentially represents a microscopic metric to study the DTW matching alone. Note that a series always matches best with itself, *i.e.*, $DTW^n_{min} < DTW^n_{smin}$. Therefore, $G > 0$ *iff* $DTW^c_{smin} - DTW^c_{min} > 0$. In other words, *a light's image can be correctly matched to its database image iff the* ***G metric*** *is a positive number.* Normally, G should be close to 1. A close-to-zero G implies that the light identification may fail under minor disturbances. Occasional wrong light identification may confuse the location with a light that is far away. Such errors can be easily corrected using spatial smoothing as in [13].

Figure 8 plots the G metric of a randomly selected light across 9 weeks. Since we cannot exactly reproduce the condition when the database image was create, G varies across measurements, but it always stays around 1 and well above 0, implying that the light identification is stable over time.

4. EFFICIENT FEATURE MATCHING

4.1 Improving Computational Efficiency

In this section, we introduce how the two assistant features help reducing the computational cost in light identification.

Color pattern. An FL or LED lamp's color temperature rating reflects the power spectrum distribution of the optical frequencies that it emits, which manifests through the "softness" of its white color. The exact color temperature pattern is determined by the dominant wavelength of the LED chip (or FL vapor), as well as the composition/thickness of the phosphor layers with different colors.

Color temperature can be characterized by *chromaticity* and *luminance* [18], representing color quality and brightness. Both metrics are almost unaffected by distance/orientation, but manufacturing variations inevitably deviate them from their nominal ratings. To capture such variations, we first compute $\overline{P}_R, \overline{P}_G, \overline{P}_B$, *i.e.*, the average value of the R, G, and B channel, respectively, across all pixels inside the image of the lamp. We then use the value ratio $\overline{P}_R/\overline{P}_G$ and $\overline{P}_G/\overline{P}_B$ to represent the lamp's chromaticity. On the other hand, the luminance Y follows a linear relation with the RGB values [25] and it is directly related to radiance:

$$Y = \mathcal{A}[\overline{P}_R, \overline{P}_G, \overline{P}_B]^T \quad (3)$$

where $\mathcal{A}$ is a vector of Color Space Transform (CST) Matrix [26]. CST is camera-specific (which maps from camera color space to CIE XYZ) and this is why we need to compensate the heterogeneity among camera models after RGB and luminance extraction as mentioned in Sec. 3.1.3.

We note that the RGB values are interleaved in a RAW image due to the use of CFA (Sec. 3.1.3). Thus, assume CFA outputs a matrix of $m \times n$ pixel values, JPEG's interpolation effect will fill each pixel with all 3 RGB values, expanding the matrix to $m \times n \times 3$. Different models of smartphone cameras may have different color configurations, and hence different RGB readings even when capturing the same scene. We thus need a one-time calibration of a camera, so that its RGB readings become consistent with the camera that was used to create the feature database. Since the main feature (SRP) alone has high confidence in identifying a light (albeit at high computational cost), iLAMP runs the SRP matching once, to identify one light that the user's camera captured. Suppose the mean power of the captured R channel is $\overline{P}_R$, and that of the database is $\overline{P}_{R0}$. Then, iLAMP scales the user camera's subsequent R channel measurements by $\overline{P}_R/\overline{P}_{R0}$, when computing the chromaticity and brightness. The same scaling process is used on the other two channels. To prevent unnecessary twisting of the colors, we set the camera's white balancing to a fixed mode (*e.g.*, daylight).

Infrared to visible intensity ratio (I2V ratio). The visible light wavelength ranges from 400 nm to 700 nm, but an FL and LED's emission spectrum can go up to 1000 nm [27] and 800 nm [19]. Optical signal leakage beyond 700 nm falls in the infrared spectrum, and the signal intensity depends on the heat generated inside the light, subject to manufacturing variation. Smartphone cameras have built-in infrared filters and cannot directly estimate the infrared intensity. Fortunately, we can repurpose the smartphone ALS as an infrared intensity sensor. Mainstream smartphones' ALS comprises two photodiodes: CH0, used primarily for sensing ambient light intensity (for adjusting screen brightness); and CH1, an infrared sensor originally used to detect proximity between the phone screen and user's cheek. One can directly measure infrared intensity using CH1, yet the measurement will vary wildly as the phone-to-light distance/orientation changes. To make the infrared intensity a stable feature, iLAMP normalize the CH1 reading by the CH0. Since the two photodiodes' frequencies are close and their FoVs are designed to cover similar range, the impact of

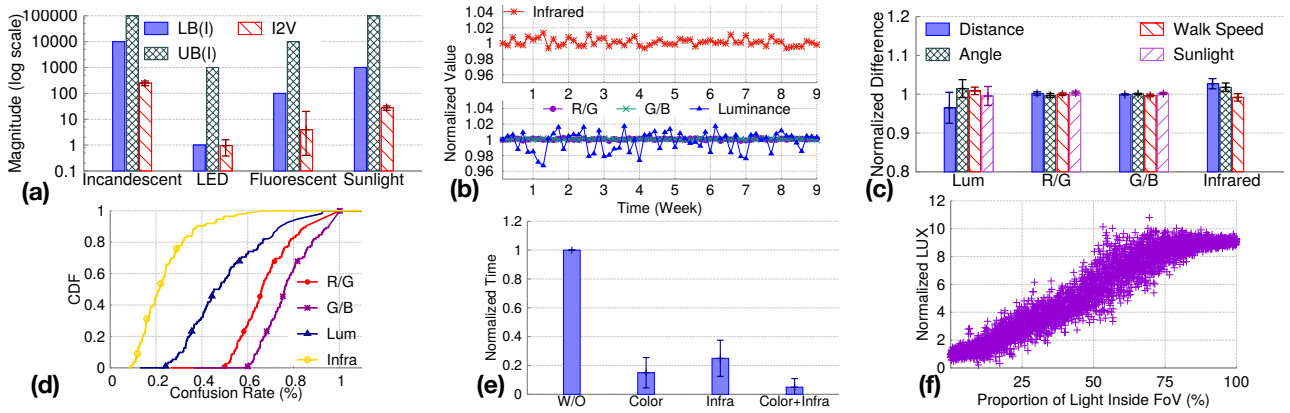

Figure 9: (a) Infrared emission of different source of lights; (b) Stability of assistant features over time; (c) The robustness of color features to overcome different distortions; (d) Confusion rate of assistant features among different lights; (e) Improving computational efficiency via hierarchical feature matching; (f) Correlation between ALS and fraction of lamp inside camera FoV.

distance/orientation on them is similar and can be canceled out after normalization.

It is also worth noting that sunlight has a wide optical spectrum and may interfere the I2V ratio. But such interference can be easily detected because sunlight has a much stronger Infrared emission. Fig. 9(a) shows the infrared intensity ("LB/UB" means lower/upper bound) and I2V on incandescent, LED, fluorescent and sunlight that we measured under a variety of conditions. For artificial lights, LB and UB are determined by distance (from 2 meters to close to 0 meter) and for natural sunlight, they are determined throughout an entire sunny day from early morning to late night. We see that the absolute values of infrared intensity is at least an order of magnitude higher than that of FLs and LEDs, even in a partially shaded region with indirect sunlight (the "LB" case), and sunlight's I2V is an order of magnitude lower than that of incandescent. Therefore, whenever iLAMP detects an ultra-strong infrared value and small I2V, it degrades to a fail-safe mode, disabling I2V ratio computation and instead uses the color pattern alone as assistant feature.

Hierarchical feature matching using assistant features. Computing the DTW distance between two main feature arrays involves quadratic complexity *w.r.t.* the array size which equals $(M + N)$. This translates into tens of ms computation time for a million-pixel image, but the computation cost increases linearly with the number of lights. To curtail the cost, iLAMP uses the assistant features to pre-filter the candidate lights before running the DTW over the main feature. Note that the assistant features may be affected slightly by random factors, *e.g.*, distance/orientation and ambient interference. We thus empirically set an upper-bound drift for each of the assistant features to guarantee we do not filter out the correct light. For the pre-filtering, we rule out the lights whose feature values deviate beyond the bound, which only requires a linear comparison across all lights in the database.

To get an intuitive understanding of the feature stability, Fig. 9(b) plots the color pattern and I2V ratio across 9 weeks, measured on a randomly selected FL in our office building, and normalized *w.r.t.* the first-day measurement. We can see that *the assistant features are highly stable over time*, with a maximum deviation of ±0.26% for the chromaticity (RGB ratio), ±3.2% for the luminance, ±1.3% for the I2V ratio, and the error is unbiased. We further evaluate the stability over usage behaviors/scenarios, by varying the distance (±1m from holding position), angle (±45°), sunlight intensity (direct sunlight from glass window on a side wall) and walking speed (0 to 2 m/s). From the results (Fig. 9)(c), we observe that the features are almost unaffected, with maximum deviation of ±7%, ±0.5% and ±4% (shown in error bars), for luminance, RGB ratio and I2V, respectively.

We thus set a conservative threshold of ±14%, ±1% and ±8% as the upper-bound drift for these three sets of features, respectively. For different light models, these thresholds can be calibrated *a priori*; but even without calibration, a conservative threshold can be used at the expense of lower discrimination (and less saving in computation).

Note that the RGB ratio is much more stable compared with luminance, as it only depends on the physical properties of the lights, such as phosphor thickness/composition.

Microbenchmark verification of computational cost. To verify the effectiveness of hierarchical feature matching, we first examine how distinguishable the assistant features are among different lights, under the same experimental setup as in Sec. 3.2. We use *confusion rate* as a metric, defined for each light $\mathcal{L}$, as the fraction of candidate lights whose feature are indistinguishable from $\mathcal{L}$, *i.e.*, the feature difference is smaller than the aforementioned upperbound threshold.

The results in Fig. 9(d) show that, although these assistant features are not as unique as the SRP, they can filter out majority of the lights. In particular, the $\overline{P}_R/\overline{P}_G$, $\overline{P}_G/\overline{P}_B$, luminance Y and I2V features have a mean confusion rate of 65%, 78% , 48%, 23%, respectively. This means on average, these assistant features can narrow down the search space to a small fraction $(0.65 \times 0.78 \times 0.48 \times 0.23 = 6\%)$ of the light candidates, assuming the features are independent across lights.

The effectiveness of discrimination in turn translates into a smaller search space for the SRP, and lower computational cost. To verify this, we use a server machine (i7-4770, 3.9 GHz) to run the light matching over 550 lights in our office building. We found that, to identify one light, a brute-force DTW matching over the million-pixel images of all light candidates takes almost 2.5 hours. Our SRP compression reduces the computation time to about 5 seconds. Fig. 9(e) shows how the hierarchical feature matching further reduces

this value. We see that the color pattern or I2V ratio alone reduces the latency to below 30%, and together cut it down to around 10.6% (0.53 second), which enables real-time localization response.

4.2 Camera Scheduling

To save power, iLAMP duty-cycles the camera and turns it on only if it is likely to capture a full lamp. Note that the smartphone's camera and light sensor have comparable FoVs and the same orientation, *i.e.*, the same norm vector with respect to phone's screen surface. Therefore, the ALS ambient light intensity should be proportional to the fraction of lamp body captured by the camera. To test this hypothesis, we walk below a light many times and intentionally vary the phone's holding position/orientation to its extremes. The scatter plot in Fig. 9(f) shows the normalized ALS ambient light intensity (*w.r.t.* the case with no light) vs. lamp fraction across 300 samples, which clearly shows a quasi-linear relation. We thus use a least square method to approximate the statistical relation, so that we can predict the availability of a full lamp by just reading the low power ALS.

More specifically, let $C_p \in [0, 1]$ be the fraction of the lamp captured by camera at a random position/orientation p. Let I_p be the corresponding ALS reading. Then,

$$C_p = I_p \cdot \phi_p + \epsilon_p \tag{4}$$

where the unknown parameter ϕ_p is a linear coefficient and ϵ_p represents errors which are assumed to be zero-mean Gaussian. To materialize this linear relation, we need to train the camera scheduler by randomly changing the phone orientation/distance, just as in real use cases. This will generate a random set of samples, corresponding to measured values $I = [I_p, I_{p-1}, \cdots, I_0]'$ and $C = [C_p, C_{p-1}, \cdots, C_0]'$. This dataset is created within the regarded light area. We then use the least square method [28] to solve for the parameter ϕ_p:

$$\phi_p = \left(I'I\right)^{-1} I'C \tag{5}$$

At run-time, given an ALS measurement I_p, we estimate $C_p = \phi_p I_p$ and turn on the camera if $C_p > 0.5$. Here we set a conservative threshold of below 1, because the camera has a setup latency of around 0.2 s, and needs to be triggered slightly ahead of time. Note that strong sunlight may mislead the camera scheduling. Thus, we switch to a fail-safe mode and keep the camera on when sunlight is detected based on ALS's infrared reading (Sec. 4.1).

The parameter training procedure can be done by each user at run-time, as iLAMP collects more and more ground-truth samples. Inside one building, different types of lamps may be deployed, corresponding to different ϕ_p. Yet iLAMP can choose the parameter that is likely to fit the nearby lights. In the worst case, iLAMP can choose the parameter so as to turn on the camera aggressively — this may reduce the power saving from camera scheduler but does not compromise the localization accuracy of iLAMP.

5. SENSOR ASSISTED PHOTOGRAMMETRY

Once a light landmark is identified and its location obtained from the database, iLAMP pinpoints the phone's location relative to that light landmark, which also provides the phone's global location within the building map. iLAMP further computes the phone's horizontal orientation, *i.e.*, heading direction, based on a geometrical model.

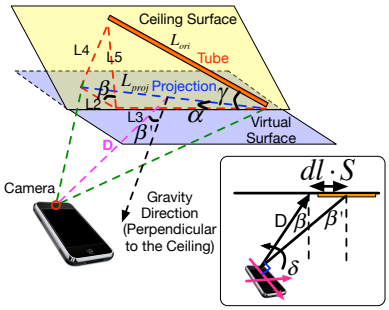

Figure 10: Geometrical model to locate a phone under a single light.

5.1 Estimating 3D Location

Our geometrical model analyzes the phone-to-light distance/orientation based on how the light's body is projected into the phone camera. The model also takes as input the standard gravity sensor output [29] from the phone's accelerometer/gyroscope. The gravity reading, unlike compass, is known to be accurate and unaffected by ferromagnetic interferences [16]. Our model abstracts a tube light as a line segment with known physical length L_{ori}. Later we will generalize it to arbitrary light shapes (Sec. 5.2).

Fig. 5.1 illustrates the geometrical model. Here L_3 is the intersection line between the ceiling plane, and the phone's *virtual surface plane* (a plane parallel to the phone surface but intersects with the lowest points on the lamp). Without loss of generality, we assume the ceiling to be perpendicular to the gravity. L_4 is perpendicular to the virtual surface plane. L_5 is a line segment within the ceiling plane and $L_5 \perp L_3$. L_{proj} is the projection of the light tube onto the virtual surface plane. α, β, β' and γ are various angles between the line segments.

Following these definitions, the following geometrical relations are straightforward: $L_{ori} = \sqrt{L_3^2 + L_5^2}$, $L_{proj} = \sqrt{L_2^2 + L_3^2}$, $L_3 = L_{proj} \cos\alpha$, $L_2 = L_5 \cos\beta = L_{proj} \sin\alpha$. Here β represents the angle between the ceiling surface and virtual plane, which can be obtained from gravity sensor ($\cos\beta = \frac{G_z}{\sqrt{G_x^2+G_y^2+G_z^2}}$ where G_* represents the earth gravity projected to the *-axis of phone). α is the phone's azimuth angle relative to the light which will be computed following Sec. 5.2. We can then obtain a *shrink factor* F between the original length of light tube L_{ori} and the length of its projection L_{proj}:

$$F = \frac{L_{ori}}{L_{proj}} = \sqrt{\cos^2\alpha + \left(\frac{\sin\alpha}{\cos\beta}\right)^2} \tag{6}$$

Further, we can compute the angle γ as:

$$\gamma = \cos^{-1}\frac{L_3}{L_{ori}} = \cos^{-1}\left(\frac{cos\alpha}{F}\right) \tag{7}$$

Suppose the camera has a focal length f and size of a single pixel L_{pix}, and the light's long side has N_{pix} pixels on the image. Following the camera imaging principle (*i.e.*, the pinhole model) [20], we have:

$$\frac{L_{proj}}{D} = \frac{N_{pix} \cdot L_{pix}}{f} \tag{8}$$

From which we obtain the phone-to-light distance D, *i.e.*, distance between the camera and the light tube's projection on the virtual surface plane. We further obtain the distance projection on each of the 3D axes as:

$$D_x = D \cdot \sin\beta' \cdot \cos\gamma, D_y = D \cdot \sin\beta' \cdot \sin\gamma, D_z = D \cdot \cos\beta' \tag{9}$$

Here the angle $\beta' = \beta$ if the center of the light aligns with the center of the image (because the gravity and D are perpendicular to L_5 and L_2, respectively). But this no longer holds when the user does not perfectly point the camera to the light. In such general cases, the deviation angle can be approximated as $\delta \approx (dl \cdot S)/D$ (Fig. 5.1), where S is the scaling factor between the physical length of the lamp with that measured on the image. dl is the horizontal or vertical (on image's coordination) pixel distance from the light's center to image center, which is scaled to a physical distance by S. The angular approximation holds since typically $D \gg dl \cdot S$ given the narrow FoV of camera. We then compensate the deviation angle as: $\beta' = \beta - \delta$ if δ is clockwise and $\beta' = \beta + \delta$ if δ is counterclockwise (Fig. 5.1). This approximation and compensation of δ is done twice, for pitch and roll angle, separately.

5.2 Estimating Heading Direction

In iLAMP, the server database stores not only the location, but also azimuth direction of each light landmark relative to the north. We then compute the phone's azimuth angle relative to the landmark's direction.

We observe that the SRP is typically distributed non-uniformly across a lamp's body (see, *e.g.*, Fig. 4). Therefore, we can define the azimuth direction even for symmetrically shaped lamps. When a smartphone changes its azimuth orientation, the SRP it measures should be rotated accordingly. So the run-time SRP of a light is a rotated version of the database version (*w.r.t.* the x-axis in the Cartesian coordinates or the 0-degree vector in the polar coordinates). The rotation angle is a direct output of the DTW matching procedure.

The above heuristic works straightforwardly when the phone is held flat, *i.e.*, the phone's azimuth plane is parallel to the ceiling and hence the pitch/roll angle equals 0 (this is also the way when the database image was taken). Under arbitrary roll/pitch/yaw angles, the light's contour may be distorted slightly, and the ratio between its width/length may deviate from the database version. But we can still fit the contour to the most similar shape in the database. Without loss of generality, consider the most commonly used rectangular lamp. Suppose P and Q are midpoints on the edges of the best-fit rectangle (Fig. 11). Then the relative azimuth between the phone and the light is:

$$\theta = \arctan \frac{y_P - y_Q}{x_P - x_Q} \tag{10}$$

where the coordinates of P and Q on the image are available after contour extraction (Sec. 3.1.3). Note that fitting the original contour to a rectangle may inject some errors but we will show the resulting heading estimation error is still quite small. For non-rectangular shaped lights, we can define a virtual rectangle (Fig. 11) corresponding to the north, and execute the same model. After obtaining heading estimation θ, we first subtract the yaw angle (*i.e.*, rotation angle with z-axis) and then obtain α in Sec. 5.1 to realize 3D localization as we mentioned previously.

5.3 Blind Area Tracking

In the "blind" area with no light coverage, alternative location tracking strategies can be employed to complement iLAMP and fill in the gap. We choose the classical motion-sensor based dead-reckoning as it is ready to use on most mobile devices. Specifically, we implement dead-reckoning following FootPath [17], which counts steps based on sharp

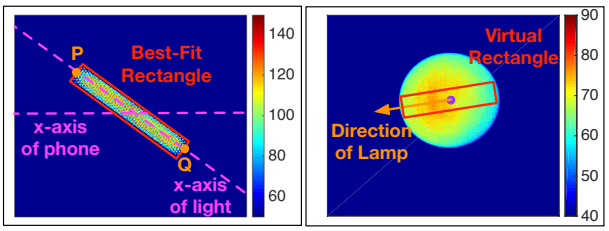

Figure 11: Heading estimation of rectangle light tube and circular bulb.

drops of phone acceleration, and uses the compass azimuth reading as heading direction. FootPath assumes a fixed stride length. In iLAMP, we estimate a user's average stride length based on the ground-truth distance between two lamps. This estimation is run between consecutive lamps, and used subsequently to translate step counts into walking distance.

Dead-reckoning is known to suffer from drift, due to the inaccurate compass and noise accumulation of the accelerometer over time [30]. Fortunately, whenever the user moves to a new light from blind area, iLAMP automatically runs the light matching and localization to correct the drift. Most indoor environment has densely deployed luminaries with a few meters of separation. Hence the blind area tends to be small and the dead-reckoning error can be well confined.

6. IMPLEMENTATION

We implement iLAMP based on a simple client-server architecture. The client side is an Android application that captures and preprocesses the images. More computationally intensive tasks are offloaded to the server which also hosts the light landmark's location database. Given an image input, the client executes the contour extraction (Sec. 3.1.3), extracts and compresses the SRP (Sec. 3.1.2). Meanwhile, it also computes the assistant features, *i.e.* RGB color pattern and I2V ratio, from the image and light sensors, respectively. The light intensity is measured by the smartphone's ALS based on an open-source driver [31], which streams the sensor readings to the user-space through SYSFS interfaces. Both the SRP and assistant features are subsequently sent to the server for matching.

The server executes the light matching mechanisms (Sec. 3.2 and Sec. 4.1) and returns the matching light landmark's global location on the floor map. Meanwhile, the client computes its 3D location and azimuth relative to the light landmark (Sec. 5), and converts the result into its own global location once it gets the server feedback. To prompt the landmark database, we have implemented a graphical user interface, which takes the building floor plan as input, and allows a user to mark a landmark's position within and associate it with a sample image of the light. We use a mobile laser ranger to measure the landmark location *w.r.t.* the floor map. We emphasize that such landmark registration procedure is needed for all infrastructure based localization schemes. For VLP systems, registering each light only takes tens of seconds, given that most of the lights in a building have similar shapes and have regular geometrical separations.

7. EXPERIMENTAL EVALUATION

7.1 Effectiveness of Light Identification

We first evaluate the accuracy and robustness of iLAMP in identifying light landmarks.

Figure 12: Field test in large buildings.

Accuracy. To represent typical use cases of indoor localization/navigation, we choose 4 different environments (Fig. 12): office 1 (588 FLs, 2.5 m ceiling), office 2 (a mix of 190 LEDs and 129 FLs, 3 m ceiling), semi-open parking ramp (232 FLs, 2.5 m ceiling) and retail store (330 FLs, 6 m ceiling). Except in the parking ramp, all the ceiling lights have plastic covers/decorators and multiple lights may be co-located inside the same house. By default, we use a Nexus 5X phone, held comfortably at around 1.2 m above the floor, capturing RAW image from its front camera. Robustness of iLAMP across different configurations will be tested subsequently.

Fig. 13(a) plots the fraction of lights that are correctly identified without any confusion with any other one inside the same building. We observe that the *main feature alone* can achieve more than 96% identification accuracy for typical buildings with up to 3 m ceilings, and more than 82% accuracy even for a 6 m ceiling. When combining the main and assistant features together through hierarchical feature matching (Sec. 4.1), the accuracy is boosted to above 95% for all the buildings. Therefore, the assistant features not only reduce computational cost, but also bring the light matching accuracy close to 100%.

The results also show that JPEG images have relatively lower discrimination accuracy when JPEG images are captured by user as well as stored in database. This is because JPEG compression processes the RAW pixels through non-linear operations, which may distort the SRP. However, the assistant features are unaffected and can still bring the accuracy above 90%. The minor residual error can be easily eliminated by combining two consecutive lights' features as the user moves as in [13]. Therefore, even for those phone models that do not support RAW output, iLAMP can still achieve much higher accuracy in light identification than the most advanced VLP system LiTell [13].

Robustness. Multiple factors in practical usage scenarios may disturb the light identification. In the following micro-benchmarks, we test the sensitivity of the SRP features to such factors, using the G metric defined in Sec. 3.2. We randomly pick one light and find its best DTW match in the database, under various disturbing factors[1]. Here we vary each following variable while keeping others fixed to their typical values.

(i) Phone orientation and height variation. The images in iLAMP's database are captured when holding the phone flat at a certain height. But the run-time images may differ as users' height and holding position varies. We first test the impact of height deviation inside Office 1, by deviating the phone from 0 m to 1.2 m relative to the height when creating the database image. We adjust the height using a tripod.

[1]We use office building 1 and Nexus 5X as the representative testbed and device for the rest of the experiments, unless otherwise stated.

As shown in Fig. 13(b), the DTW tolerance gap G decreases with height deviation, since distance affects the number of pixels of a light's image. Fortunately, the main features are diverse enough, and DTW can tolerate the missing/addition pixels (*i.e.*, contraction and stretching) due to image distortion. Consequently, G is still well above 0 even if the height deviates from the database benchmark by 1.2 m (typically from the holding position to the ground). We further rotate the phone around the axial direction of the tube, so that the relative orientation (radiation angle from the lamp and incidental angle into the phone) changes by up to 45° (maximum deviation to ensure the entire light can still be captured). Fig. 13(c) shows that the G metric deviates slightly, but remains well above 0 even with 45° angular deviation. These two experiments verify that *iLAMP can maintain high light identification accuracy even if the run-time capturing height/orientation deviates from the database benchmark by a practical offset.* Note that severe distortion may happen under certain cases, *e.g.*, when the phone rotates its pitch angle which results in perspective changes. Such effects can be compensated through classical computer vision techniques, but are beyond the scope of the present work.

(ii) Image resolution and ceiling height. To further test iLAMP's robustness against JPEG compression, we adjust the JPEG's resolution of Nexus 5X from its default 5MP to lower than 1.3MP, while using the same 5 MP images in the database for light matching. From Fig. 13(d), we can see that a more aggressive JPEG compression reduces the diversity of visual features among lights. But even at a low resolution of 1.56 MP, the G metric remains above 0. When the image resolution degrades to below 1.3MP, G becomes close to 0, implying that confusion among lights occurs. Nonetheless, today's mainstream front-cameras mostly have higher than 2 MP resolution, which ensures iLAMP's robustness. Note that the RAW image quality is unaffected by the JPEG resolution. It only depends on the size of the image sensor within the camera. On the other hand, increasing the ceiling height has the same effect as reducing the image resolution. Fig. 13(e) quantifies the effects by proportionally sub-sampling the image, which shows that the accuracy remains above 70% even under the extreme case with 10 m height ceiling. In contrast, alternative solutions that leverage frequency features [13] can only correctly discriminate individual lights with 40% accuracy even at a low ceiling height of 2.5 m.

(iii) Partial light distortion. Although iLAMP invokes light matching only if a full light is captured, it can also run in an aggressive mode and responds even if a partial light is visible. Fig. 13(f) plots the light matching accuracy, where we intentionally vary the fraction of a light inside the camera FoV (an entire light fixture includes 2 to 3 tubes, occupying 1.2 m × 0.6 m in Office Building 1). The resulting G *degrades only slightly even when only* 1/2 *of the light is visible in common office buildings.* Besides the SRP itself, the resilience is also attributed to the assistant features, which capture the average color pattern or I2V ratio and remain stable even with a partial light image.

(iv) Ambient sunlight interference. Certain buildings may install sidewall windows through which sunlight can peek in and interfere the camera imaging. We verify the impact by placing the smartphone 1.5 m under an FL immediate to a window in Office 1. Fig. 13(g) plots the measured G metric

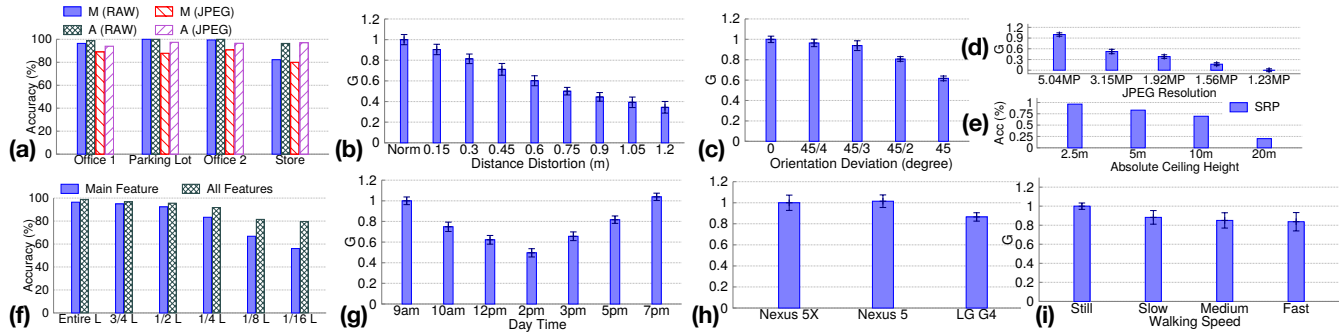

Figure 13: (a) Accuracy of light identification with Main (M) feature or All (A) features; Impact of (b) distance variations (relative to normal holding position); (c) specific orientation variations; (d) different camera resolutions and (e) ceiling heights; (f) partial light; (g) ambient sunlight interferences; (h) device heterogeneity; (i) walking speed on main SRP feature.

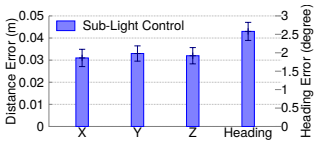

Figure 14: 3D location and heading estimation even under a single light landmark.

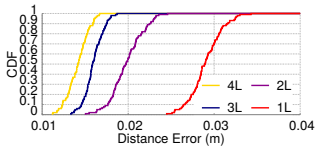

Figure 15: Localization precision improves as more light landmarks fall in the camera's FoV.

across a sunny day. At 2pm, the sunlight has the smallest incidental angle of approximately 30°, which results in the lowest G, but still well above 0. Therefore, *iLAMP's light discrimination mechanism is robust against normal indoor sunlight interference.* Although the sunlight may slightly reduce the image contrast, the scaling effect can be easily counteracted by the DTW matching. In case when there is strong direct sunlight with incidental angle close to 0 (*i.e.*, coming from the same angle as the lamp), the camera tends to be saturated and the lamp's image is no longer viable for feature extraction. But such cases rarely occur in practice.

(v) Device heterogeneity. To test how iLAMP works across different camera hardware, we create the database using Nexus 5X (5 MP, 1.4μm sensor), and then test the light discrimination accuracy using *calibrated* LG G4 (8 MP, 1.2 μm), Nexus 5 (8 MP, 1.4μm). The Nexus 5 front-camera cannot output RAW image, so we use its rear camera instead. We use these phones to capture the same light, and then compute the corresponding G metric. Fig. 13(h) shows that the G metric varies negligibly and well above 0 across phone models, implying that *iLAMP's light identification accuracy is almost unaffected even if the run-time images are captured using different phones than the database images.*

(vi) Walking speed. The user's walking speed may affect the image quality because most front-cameras do not have an optical image stabilizer. To measure the impact, we walk and hold the phone in stable at slow (≈0.5 m/s), medium (≈1 m/s) and fast (≈ 2 m/s) speed across one light. Fig. 13(i) shows that the G metric remains around 1, implying *iLAMP's light discrimination mechanism is robust against the walking patterns.*

7.2 Precision of Location and Heading Estimation

To verify the sensor-assisted photogrammetry, we place the phone under a tube FL at 25 random spots, with horizontal displacement up to 1.6 m, and vertical up to 2 m. At each spot, we randomly rotate the phone at 3 different roll/yaw/pitch angles. Fig. 14 plots the mean and 90-percentile (error bars) accuracy. We observe that *iLAMP achieves a mean localization precision of around 3.2 cm and 90-percentile of 3.5 cm across all axis, and mean heading estimation error of* 2.6° *and 90-percentile of* 2.8°. Further, we repeat the experiment by mounting multiple lamps closely on the ceiling, and take a simple average of the location estimation *w.r.t.* each lamp. Fig. 15 further shows that, as the number of lights increases to 4, the 3D localization error (90-percentile) drops to 1.7 cm, implying that *iLAMP's accuracy further improves under densely deployed light fixtures.*

We further conduct *field tests* and use iLAMP to navigate across two environments (Fig. 16): *(i)* a $9 \times 9m^2$ research lab with *densely deployed* ceiling FLs which ensures 1 or 2 lights are always visible to the phone and thus no blind area tracking is needed); *(ii)* a $90 \times 70m^2$ office building corridor with *sparse* light deployment (3 m separation). Thus both room and corridor are included. In both scenarios, ground-truth is created by placing markers with different colors along a predefined track. We use Google Tango [32], which is known to have centimeter precision. We stick it with the smartphone so that it runs spatially in sync with iLAMP. Here we follow the best-practice guideline of Google Tango to ensure a highly accurate ground truth trace assisted by these unique markers on the floor [32]. A user walks across the track while naturally holding the phone with Tango and sends a localization request through iLAMP when passing each marker position.

Fig. 16(a) shows that the location trace measured by iLAMP is highly consistent with the ground truth. For clarity, Fig. 17 further plots the error vector on the horizontal plane across all the sampled spots. iLAMP demonstrates a 90-percentile precision of around 2.7 cm, which is consistent with the previous controlled test.

Fig. 16(b) and Fig. 18 plot the localization traces and error vectors inside the large office environment. Since lights are visible intermittently, iLAMP invokes the blind area tracking (Sec. 5.3) occasionally. When the dead-reckoning (DR) is used alone, the mean error is around 3 m, consistent with state-of-the-art evaluation [17]. iLAMP can intermittently correct the DR drift, reducing the mean error to 0.18 m and 90-percentile to 0.44 m. Therefore, *iLAMP not only provides absolute position fixes to DR, but also enhances its precision by an order of magnitude in environment with sparse light installation.*

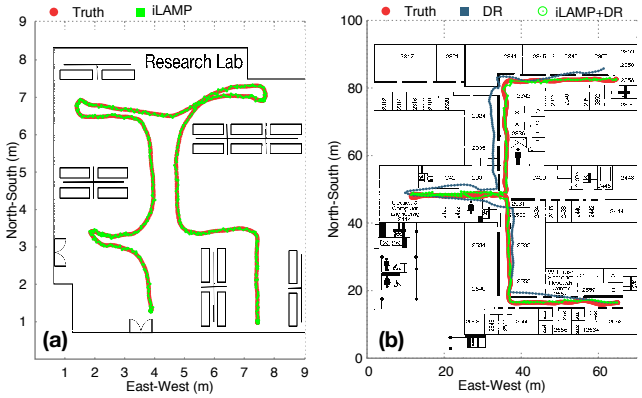

Figure 16: Field tests of real-time navigation.

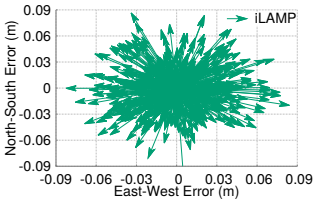

Figure 17: Error vector in environment with densely deployed lights.

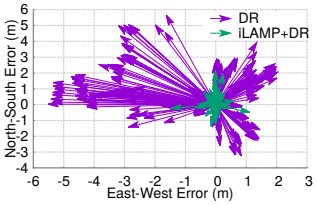

Figure 18: Error vectors under sparse light deployment (with blind area tracking).

7.3 System Efficiency

Latency. iLAMP's end-to-end operations can break down into 3 steps: local processing (feature extraction) on the phone, phone-to-server data transmission, and light matching on the server. We time-stamp these operations and plot the latency in Fig. 19. The measurement is done on a Nexus 5X client and an Intel i7-4770 3.9 GHz server. We observe the end-to-end latency takes 0.37 to 0.7 s per localization operation, and is roughly proportional to the number of lights inside a building. The local processing and transmission takes only around 160 ms on the smartphone, and is invariant across environment. Remarkably, the light matching procedure on the server dominates the computational cost, taking almost 0.6 s inside the large Office 1 with 588 light fixtures. This latency can be reduced substantially through several measures: (i) An optimized C-based DTW implementation which replaces our current Matlab implementation on the server. (ii) A multi-thread implementation that harnesses the server's multiple CPU cores. In addition, iLAMP can easily scale to a large number of clients, because the clients' DTW computation and database lookups are independent and can be easily distributed across many servers.

Energy efficiency. We use the Monsoon power monitor [33] to measure the smartphone's power consumption when running iLAMP on Nexus 5X inside the Office 1 (Fig. 12). Fig. 20 plots the real-time power consumption as the user walks across 7 lights, which demonstrates that the camera scheduler can judiciously turn off the camera in the blind region. We further run the test across 100 lights and examine the average power consumption. Fig. 21 further plots the fraction of camera-on time, verifying that setting the C_p threshold to an intermediate value makes a balanced tradeoff: it is more responsive compared with an aggressive strategy (*i.e.*, camera on if $C_p \approx 1$), yet saves more power compared with a conservative strategy (*i.e.*, camera on if $C_p > 0$).

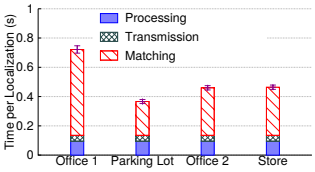

Figure 19: Breakdown of localization latency (using a single CPU core).

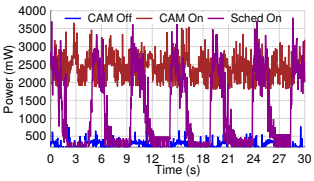

Figure 20: A snapshot of power consumption.

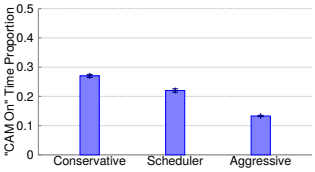

Figure 21: Robustness of camera scheduler.

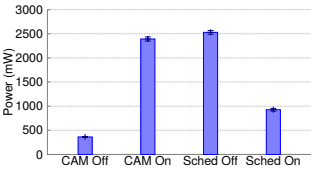

Figure 22: Breakdown of average power consumption.

Fig. 22 provides a breakdown analysis of the power consumption. Even without running iLAMP, the camera alone escalates the system power consumption to 2.5 W ("CAM on"). Turning on iLAMP's processing only adds less than 100 mW ("Sched off"). With the camera scheduler activated ("Sched on"), *iLAMP can effectively reduce the power consumption to 0.93 W — a 62% reduction.* For buildings with dense light deployment, the power saving will be smaller, but iLAMP can still duty cycle the camera based on how frequently the localization is needed.

8. DISCUSSION

Bootstrapping overhead. The landmark survey is a critical bootstrapping step in all infrastructure based localization systems. Specific to iLAMP, it requires capturing the features of each lamp and marks the lamp on a floor plan. This procedure only needs to be done once. It involves much less overhead compared with traditional WiFi fingerprinting, which requires surveying each location spot instead of landmark. iLAMP's sensor-assisted photogrammetry technique needs to know the physical size (edge width and length) of each light fixture, but the measurement needs to be done mostly only for a few representative lights because most of the lights in a building come from the same hardware model. As a contrast, the WiFi APs' locations can be much harder to identify because many of the APs are hidden from users and are accessible only to building managers.

Robustness under various lamp shapes and image distortions. Most of the large commercial buildings we observed and experimented with embed their light fixtures inside the ceiling. Or they house the light bulbs/tubes with a cover that reshapes majority of the light beam towards the floor. Therefore, a 2D camera image mainly captures the azimuth cross-section of the light fixtures. Changing the imaging perspective, or cutting/distorting part of the image (or equivalently adding certain non-cross-section parts) does not affect the light identification in a noticeable way (Sec. 7.1). However, certain pendant lamps or chandeliers may largely expose the side fractions of their bodies. To deal with such cases, iLAMP can fall back to a conservative mode, and trigger light identification only if the image shape matches cross-section of the light taken in the database.

Our current implementation of iLAMP captures the ceiling light fixtures' tubes as well as covers to extrapolate unique

features. Whereas ambient physical accessories may enrich a light's feature, they may also make the light identification sensitive to phone orientation or perspective changes. For example, the partition structures in certain FL fixtures may occupy different fraction of the image depending on the phone's horizontal position relative to the light. As mentioned in Sec. 7.1, such artifacts can be reduced by using proper structural analysis and computer vision techniques, but the solution is beyond the scope of this work.

Integration with alternative localization modalities. RF localization can be an alternative to dead-reckoning in iLAMP's blind area tracking. But to achieve high accuracy, RF localization schemes often require dense AP deployment, known AP locations, and CSI readings [1–3], which are not readily available for most of today's buildings and smartphones. Yet simple AP identities can inform iLAMP of the building or section it is in, which can become assistant features to help iLAMP narrow down its search space. iLAMP works best when the phone is held with camera facing up. RF localization may complement the cases when the phone is not exposed to LoS lights (*e.g.*, in user's pocket). Existing system's light features such as flickering frequency [13] can also be introduced as one assistant feature of iLAMP.

Privacy issues. iLAMP sets the camera exposure time to a very small value to make the ceiling light stand out of the background. In fact, the background is rendered black in all the indoor environment we have tested. Moreover, iLAMP compresses the image features into a single row and column, so no visible information will be leaked to the server. Overall, iLAMP easily preserves user privacy, unlike other visual localization approaches such as SLAM [34] which needs to capture physical scenes.

9. RELATED WORKS

Over the past two decades of research in indoor localization, RF based approaches garnered the most attention due to the wide adoption of WiFi. RF localization falls in two general categories: *fingerprinting* and *model-driven*. The fingerprinting method associates each location with the RSS [35] or channel state information (CSI) [36, 37] measured *w.r.t.* multiple access points (APs). Such RF metrics are known to be unreliable due to the small-scale fading effects, caused by multipath reflections, device movement, and human activities. Also, the fingerprinting procedure is labor intensive, requiring a blanket survey of all location spots [36, 37] This should not be confused with the much simpler landmark registration procedure which marks landmark (*e.g.*, WiFi APs or ceiling lights) positions within a floor map.

Model-driven RF localization can directly compute the line-of-sight distance/angle between APs and the mobile device, based on propagation time [2, 38] or angle-of-arrival (AoA) [1, 3]. However, due to the intrinsic instability of the wireless channel, the reliability of such approaches remains an issue: despite the decimeter-level median precision, the 90-percentile error remains at 2 to 10 meters when tested in real buildings [1–3]. Hence, they may suffice for long-term navigation, but will impair user experience in other applications that require instant and precise location fix, such as item localization and targeted advertisement in retail stores.

Since the early conceptual development in 2004 [15], existing VLP research focused intensively on two issues: light identification and device localization. Almost all the VLP solutions in the past decade used modulated smart LEDs that send digital identification beacons [7–11]. Although LiTell [13] obviates the need for such specialized LEDs, it only works for FLs with natural flickering frequencies. LiTell's frequency-based features have high confusion rate (60% even when discriminating a small population of 100 lights). So it has to combine multiple lights sequentially to enrich the feature, resulting in longer latency. Its limitation to low ceilings and back-cameras also hampers its real-world usage. In contrast, iLAMP reduces the confusion rate close to 0, and works readily with low-resolution front-cameras and high ceilings. As for device localization, PD based VLP systems follow the Lambertian radiation model to compute distance using RSS, and location using trilateration. But the RSS-distance model no longer holds for tube lights [7], and when collimating or diffusing covers/lenses are used for uniform illumination [39]. Camera based VLP [10,11] overcomes the limitation using AoA-based photogrammetry. LiTell [13] builds on this approach and transforms the distortion of lamp shape into camera position. Yet its model is applicable only for tube lights, and when the camera is held flat (Sec. 5). Besides, LiTell cannot identify the phone's azimuth orientation.

Motion sensors can track a user's relative movement via dead-reckoning [17, 40, 41], but need to be calibrated by other approaches that provide absolute location fixes [30]. State-of-the-art vision-based robotic systems integrate motion sensors with a camera to realize visual-inertial odometry (VIO) or SLAM [34], which tracks user movement continuously via image differential [42]. But the performance suffers in environment with uniform visual features (*e.g.*, office hallways) or dynamic scenes (*e.g.*, retail stores) while requiring both specific camera and continuous power consumption by video recording [34]. iLAMP hints to a new principle that can benefit the vast research in VIO and SLAM: using a computational imaging approach, many of the seemingly homogeneous scenes can become distinguishable, and hence contribute to higher precision in VIO and SLAM with commodity smartphones. Using lights instead of the ambient scenes also brings several key advantages. In particular, the lights have high contrast from the background, and are deployed regularly at discrete points. These properties simplify landmark registration and lower the image processing overhead, thus enabling accurate, real-time, and energy efficient localization.

10. CONCLUSION

Despite decades of research, accurate, robust, and low-cost indoor localization is still recognized as a grand challenge in mobile computing [43]. In this paper, we proposed iLAMP as a novel visible light localization system to confront this challenge. iLAMP uses a smartphone to efficiently extract the intrinsic visual features in unmodified LED/FL lamps, and identify each lamp as landmark with close to 100% confidence. iLAMP further introduces a sensor assisted photogrammetry technique to estimate the smartphone's 3D location (heading direction) with a small 90-percentile error of 3.5 cm (2.8°). Our Android implementation also demonstrated iLAMP as a low-latency and energy efficient localization system readily usable in today's buildings. Future research is also required to improve the robustness and extendability of our prototype and combine with other VLP or RF localization schemes.

11. REFERENCES

[1] J. Xiong and K. Jamieson, "ArrayTrack: A Fine-grained Indoor Location System," in *Proc. of USENIX NSDI*, 2013.

[2] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level Localization with a Single WiFi Access Point," in *USENIX NSDI*, 2016.

[3] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," in *Proc. of ACM SIGCOMM*, 2015.

[4] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the Limit of WiFi Based Localization for Smartphones," in *Proc. of ACM MobiCom*, 2012.

[5] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home Gesture Recognition Using Wireless Signals," in *ACM MobiCom*, 2013.

[6] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A Visible Light Based Positioning System," in *Proc. of USENIX NSDI*, 2014.

[7] B. Xie, K. Chen, G. Tan, M. Lu, Y. Liu, J. Wu, and T. He, "LIPS: A Light Intensity–Based Positioning System for Indoor Environments," *ACM Transactions on Sensor Networks*, vol. 12, no. 4, 2016.

[8] B. Xie, G. Tan, and T. He, "SpinLight: A High Accuracy and Robust Light Positioning System for Indoor Applications," in *Prof. of ACM SenSys*, 2015.

[9] N. Rajagopal, P. Lazik, and A. Rowe, "Visual Light Landmarks for Mobile Devices," in *Proc. of ACM/IEEE IPSN*, 2014.

[10] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proc. of ACM MobiCom*, 2014.

[11] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Wearables Can Afford: Light-weight Indoor Positioning with Visible Light," in *Proc. of ACM MobiSys*, 2015.

[12] U.S. Department of Energy, "Energy Savings Forecast of Solid-State Lighting in General Illumination Applications," Aug. 2014.

[13] C. Zhang and X. Zhang, "LiTell: Robust Indoor Localization Using Unmodified Light Fixtures," in *Proc. of ACM MobiCom*, 2016.

[14] R. Lukac, *Computational Photography: Methods and Applications*. CRC Press, 2016.

[15] S. Horikawa, T. Komine, S. Haruyama, and M. Nakagawa, "Pervasive Visible Light Positioning System using White LED Lighting," *Technical report of IEICE DSP*, vol. 103, no. 719, 2004.

[16] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proc. of ACM MobiCom*, 2014.

[17] J. Ã. B. Link, P. Smith, N. Viol, and K. Wehrle, "FootPath: Accurate Map-Based Indoor Navigation Using Smartphones," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[18] T. Q. Khanh, P. Bodrogi, Q. T. Vinh, and H. Winkler, *LED Lighting : Technology and Perception*. Willey, 2014.

[19] S. Pimputkar, J. S. Speck, S. P. DenBaars, and S. Nakamura, "Prospects for led lighting," *Nature Photonics*, vol. 3, no. 4, pp. 180–182, 2009.

[20] M. W. Burke, *Image Acquisition: Handbook of machine vision engineering: Volume 1*. Springer, 1996.

[21] G. B. Garcia, O. D. Suarez, J. L. E. Aranda, J. S. Tercero, and I. S. Gracia, *Learning Image Processing with OpenCV*. Packt Publishing, 2015.

[22] B. Gary and K. Adrian, "Learning opencv: Computer vision with the opencv library," *O'Reilly USA*, 2008.

[23] D. B. Goldman and J.-H. Chen, "Vignette and Exposure Calibration and Compensation," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2005.

[24] E. Keogh and A. Ratanamahatana, "Everything You Know About Dynamic Time Warping is Wrong," *SIG-KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[25] H. S. Fairman, M. H. Brill, H. Hemmendinger *et al.*, "How the cie 1931 color-matching functions were derived from wright-guild data," *Color Research & Application*, vol. 22, no. 1, pp. 11–23, 1997.

[26] C. CIE, "Commission internationale de l'eclairage proceedings, 1931," 1932.

[27] R. Narasimhan, M. D. Audeh, and J. M. Kahn, "Effect of Electronic-Ballast Fluorescent Lighting on Wireless Infrared Links," http://wireless.stanford.edu/papers/Ravi/iee1996ir.pdf, 1996.

[28] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006.

[29] Android Developers, "Gravity Sensor," https://source.android.com/devices/sensors/sensor-types.html#gravity, 2016.

[30] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No Need to War-drive: Unsupervised Indoor Localization," in *Proc. of ACM MobiSys*, 2012.

[31] Avago, "Avago 9930 linux driver," 2016. [Online]. Available: http://github.com/CyanogenMod/android_kernel_lge_hammerhead/blob/cm-13.0/drivers/misc/apds993x.c

[32] Google Tango Developer Guide, "https://developers.google.com/tango/overview/motion-tracking."

[33] Monsoon Solutions, Inc., "Monsoon Power Monitor," https://www.msoon.com/LabEquipment/PowerMonitor/.

[34] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age," *CoRR*, vol. abs/1606.05830, 2016.

[35] P. Bahl and V. Padmanabhan, "RADAR: an In-Building RF-Based User Location and Tracking System," in *Proc. of IEEE INFOCOM*, 2000.

[36] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based Indoor Localization," in *Proc. of ACM MobiSys*, 2012.

[37] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information," in *Proc. of ACM MobiSys*, 2012.

[38] S. Sen, D. Kim, S. Laroche, K.-H. Kim, and J. Lee, "Bringing CUPID Indoor Positioning System to Practice," in *International Conference on World Wide Web (WWW)*, 2015.

[39] A. J. W. Whang, Y. Y. Chen, and Y. T. Teng, "Designing Uniform Illumination Systems by Surface-Tailored Lens and Configurations of LED Arrays," *Journal of Display Technology*, vol. 5, no. 3, 2009.

[40] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors," in *Proc. of ACM Conference on Ubiquitous Computing (UbiComp)*, 2012.

[41] Q. Xu, R. Zheng, and S. Hranilovic, "Idyll: indoor localization using inertial and light sensors on smartphones," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 307–318.

[42] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, 2016.

[43] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned," in *Proc. of ACM/IEEE IPSN*, 2015.