LiTell: Robust Indoor Localization Using Unmodified Light Fixtures

Chi Zhang
University of Wisconsin-Madison
czhang296@wisc.edu

Xinyu Zhang
University of Wisconsin-Madison
xyzhang@ece.wisc.edu

ABSTRACT

Owing to dense deployment of light fixtures and multipath-free propagation, visible light localization technology holds potential to overcome the reliability issue of radio localization. However, existing visible light localization systems require customized light hardware, which increases deployment cost and hinders near term adoption. In this paper, we propose LiTell, a simple and robust localization scheme that employs unmodified fluorescent lights (FLs) as location landmarks and commodity smartphones as light sensors. LiTell builds on the key observation that each FL has an inherent characteristic frequency which can serve as a discriminative feature. It incorporates a set of sampling, signal amplification and camera optimization mechanisms, that enable a smartphone to capture the extremely weak and high frequency (> 80 kHz) features. We have implemented LiTell as a real-time localization and navigation system on Android. Our experiments demonstrate LiTell's high reliability in discriminating different FLs, and its potential to achieve sub-meter location granularity. Our user study in a multi-storey office building, parking lot and grocery store further validates LiTell as an accurate, robust and ready-to-use indoor localization system.

CCS Concepts

•**Computer systems organization** → **Special purpose systems;** •**Hardware** → **Signal processing systems; Digital signal processing; Noise reduction;** Sensors and actuators; Sensor applications and deployments; •**Information systems** → *Location based services;* •**Human-centered computing** → *Ubiquitous and mobile computing systems and tools;* •**Computing methodologies** → Camera calibration;

Keywords

Visible light sensing; Visible light localization; Indoor localization; Smartphones; Signal processing; Image processing

1. INTRODUCTION

Indoor localization technology is bringing huge impacts on human activities, in the same way that GPS did in revolutionizing outdoor navigation. Existing market research predicts that location-based services in retail industry alone will generate 10 billion revenues by 2020 [1]. However, after decades of research, there still lacks an indoor localization solution with desired *simplicity* and *robustness*. A recent field test of state-of-the-art localization schemes [2] revealed a common set of problems including high deployment overhead and low reliability. The consensus is that robust meter-level indoor localization remains an open problem, even in a small sandbox environment ($300\mathrm{m}^2$) with simple layout [2].

WiFi based localization techniques are most extensively studied. Prior research has explored fingerprinting [3, 4], propagation modeling [5], and directional antenna triangulation [6, 7]. However, the elusive nature of radio signals renders them less reliable in real environments, specifically due to three major challenges. *(i)* Multipath reflections. Multipath reflections are strongly dependent on the geometries and construction materials in indoor environments, which defeats model based approaches. *(ii)* Environmental dynamics. Minor change in the environment (*e.g.*, adding a new furniture, human presence and mobility) can substantially disturb the received signals strength (RSS) and phase, thus compromising location estimation [2]. Even the body orientation and holding position of devices (which block the antenna in different ways) can cause 6 to 8 meters of location error [8]. *(iii)* Device heterogeneity. Depending on model, co-located smartphone devices can observe up to 10 dB RSS differences [9]. In enterprise environments, access points' dynamic channel selection and power control further varies the signal features.

Visible light (VL) localization techniques hold potential to overcome such fundamental limitations. Using ceiling-mounted LEDs as beaconing devices, VL localization can achieve sub-meter precision [10, 11], and can even determine the orientation of a smartphone [12]. However, almost all existing LED based localization techniques require customized beaconing circuits to be added to the LED driver [13], which involve substantial retrofitting cost and are unlikely to be adopted pervasively in the near term [14].

In this paper, we propose *LiTell*, a simple and robust VL localization scheme, which can be immediately used on unmodified light fixtures and commercial-off-the-shelf (COTS) smartphones. LiTell uses incumbent fluorescent lights (FLs) as location anchors, and smartphones as the receiver devices. The key hypothesis is that an FL's driver acts as an oscillator with a resonance frequency. Due to unavoidable manufacturing variation, different FLs have different resonance frequencies, which will cause each of them flickering at a characteristic frequency (CF), a high frequency (> 80 kHz) that is not perceptible by human and remains relatively stable in practice. LiTell uses the CF as a discriminative feature among different FLs which in turn serve as location landmarks.

We have tested the hypothesis using a customized high-speed light sensor. Our experiments demonstrate that the CFs are highly *diverse*:

MobiCom'16, October 03-07, 2016, New York City, NY, USA
© 2016 ACM. ISBN 978-1-4503-4226-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2973750.2973767

among 500 FLs in an office building, over 99% have a pairwise CF separation of more than 20 Hz. The CF is also highly *reliable*: it is unaffected by typical environment dynamics or human behavior, and varies by only a few tens of Hz across several months. To our knowledge, this represents the first study to characterize the FLs' optical frequency feature and use it to discriminate FLs.

However, many challenges emerge when using COTS smartphones as light sensors: *(i) Low sampling rate*. smartphones are designed for snapshotting low-motion scenes at around 30 frames per second. The rolling-shutter mechanism, which allows a smartphone to capture several thousand samples within one image frame [12,15], may alleviate the problem. But the sampling rate remains far below an FL's CF. *(ii) Low sensitivity to high-frequency light signals*. Camera sensors have a limited dynamic range, majority of which is already occupied by low-frequency but high-power (bright) signals when capturing a real scene. Besides, substantial noise exists in high frequencies, introduced by camera hardware (*e.g.*, heterogeneity of color pixels, salt-and-pepper noise) and spatial patterns of environment structures (*e.g.*, latticed covers around FLs). These factors tend to immerse the CF signals in the camera image's noise floor.

To overcome camera's low sampling rate, LiTell's solution builds on two observations. First, the CF is an extremely sparse feature, representing a single peak in the frequency spectrum. Second, although cameras inherently snapshot low-rate scenes, their analog bandwidth can reach a few hundred kHz. Thus, when sampled at a low rate, the CF will be "folded" back to low frequencies due to aliasing effect [16]. By optimizing the camera's sampling mechanics (*e.g.*, exposure time), along with the rolling shutter effect, we are able to recover CFs well above 80 kHz. In addition, to combat the low-sensitivity issue, we design a set of feature amplification mechanisms that synthesize consecutive captures of the FL, isolate the interference from ambient spatial patterns and mitigate the noise from camera hardware. These mechanisms substantially boost the SNR, allowing the CF to be easily identified in the spectrum.

LiTell's actual location matching mechanism is fairly simple. We first use a light sensor to measure the ground-truth CF of each FL luminary, and store the ⟨*location, CF*⟩ pair in a fingerprinting database. The fingerprinting overhead is bounded by the number of light fixtures, each taking only a few seconds. Moreover, unlike WiFi-based localization, LiTell's fingerprinting procedure can be conducted by anyone, without access to any managed infrastructure. At run time, any smartphone can capture images of a nearby FL, run the above sampling and amplification mechanisms to identify the CF, and look up the location from the database. This affords *light-cell level localization*, which translates into meter-level location granularity, considering the pervasive and dense deployment of FLs in typical public buildings. To improve reliability and counteract occasional CF feature collisions, we use every two consecutive FLs as a location landmark. Furthermore, we take advantage of the scaling relation between the physical size and image size of an FL, and use simple geometrical transformation to derive the smartphone's position relative to the FL. This makes it possible to realize finer-grained localization with *sub-meter precision*.

We have implemented LiTell's CF identification mechanisms as a mobile app on Android smartphones, and also prototyped an indoor navigation app atop. Our micro-benchmark experiments in a 4-storey office building demonstrate that LiTell can identify the CF features with high reliability. The identified features stay within 20Hz of the ground-truth, with up to 2m light-to-phone distance, under various usage behaviors and environment conditions, and across multiple generations of Android phone models built from 2012 to 2015.

To verify LiTell's localization accuracy, reliability and usefulness in the wild, we conduct a user study with 10 volunteer participants. The user study took place in 3 uncontrolled environments: an office building ($\approx 9000\text{m}^2$, 119 FLs), an indoor parking lot ($\approx 2800\text{m}^2$, 91 FLs), and a grocery store ($\approx 1000\text{m}^2$, 162 FLs). LiTell demonstrates the following key features: *(i)* Accuracy and usability. LiTell identifies the FLs with an average accuracy of 90.3% across the 3 sites. Typically, it navigates the user to randomly selected POIs with 50% shorter time and 60% shorter distance consistently, compared with using visual landmarks and labels. It adds marginal overhead, compared with an oracle walking trial (directly towards destination with known routes). *(ii)* Robustness. By using the CF as signature, LiTell's localization accuracy becomes unaffected by heavy human activities and environment dynamics. To our knowledge, LiTell represents the first ready-to-use, real-time localization system to achieve a combination of such desirable traits.

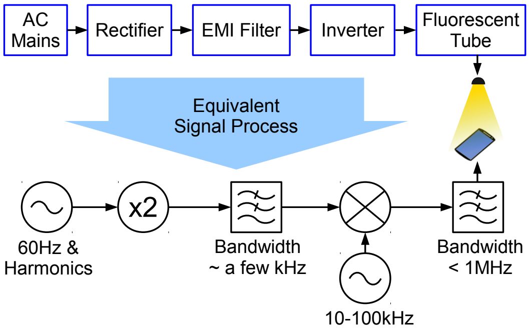

Figure 1: General circuitry model of FL.

The main contributions of LiTell can be summarized as follows: *(i)* We conduct the first comprehensive feasibility study to verify the optical CF as a *diverse* and *reliable* feature to discriminate FLs. *(ii)* We design a set of sampling and signal amplification mechanisms that allow a COTS smartphone to capture the CF feature and single it out from various noises. *(iii)* We develop simple fingerprinting based schemes that enable LiTell to achieve robust and accurate localization, with light level and sub-light level location granularity. The real-time version of LiTell system is implemented on Android, and verified in uncontrolled, multi-floor environment.

2. CHARACTERIZING ELECTRONIC AND OPTICAL PROPERTIES OF FLUORESCENT LIGHTS

2.1 Background: Fluorescent Light and Its Frequency Components

A fluorescent light (FL) produces visible light by striking an arc across a tube lamp, causing the gas and fluorescent material inside to glow. The amount of current passing through the lamp determines the light intensity. FL driver acts as the core circuitry for light generation. It converts the AC mains voltage (110V/220V, 50Hz/60Hz) to a high frequency AC voltage to sparkle the fluorescent tubes [17].

Fig. 1 depicts a general breakdown model of FL driver. The rectifier converts AC mains to DC while leaving a residual AC component on top of the DC, whose frequency is doubled from the 50/60Hz AC mains frequency to 100/120Hz. The current then goes through some electromagnetic interference (EMI) filters. Afterwards, the core of the FL driver, an *inverter*, modulates this DC into high frequency AC to drive the fluorescent tube.

The resonance frequency of the inverter is determined by a group of components [18]. Due to manufacturing variation, values of these components usually vary within 5-20% range [19]. As a result, even among FLs of the same model, their resonance frequencies tend to vary significantly. Meanwhile, since FLs are designed to operate

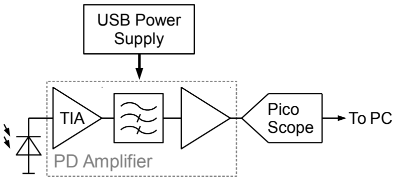

Figure 2: Experimental setup to measure FLs' frequency features.

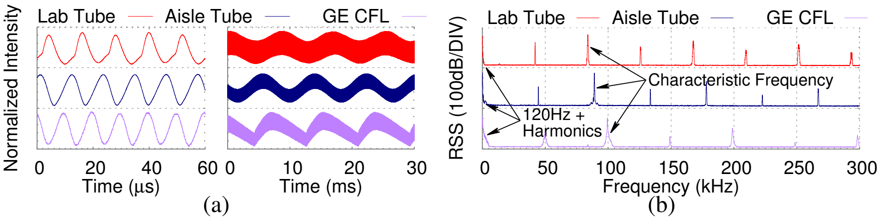

Figure 3: Light intensity (a) waveforms at different time scales and (b) frequency components of FL.

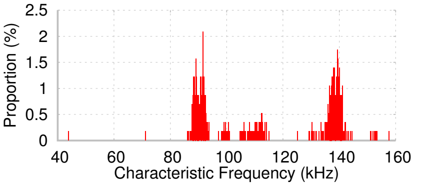

Figure 4: Distributions of characteristic frequencies of FLs in the office building.

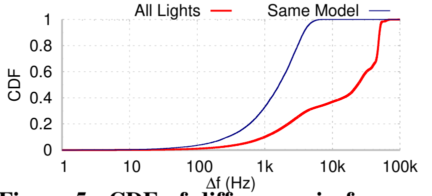

Figure 5: CDF of differences in frequencies (Δf) between all FLs in the office building and FLs of the same model.

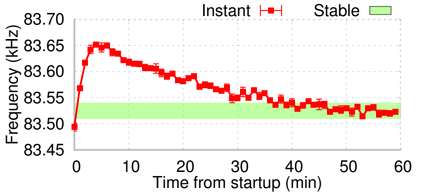

Figure 6: Characteristic frequency variations during the start-up process of one tube FL.

efficiently and consistently over periods of years, such variation is likely to remain stable over time.

The fluorescent tube converts the high frequency AC current (generated by the inverter) into visible light. Due to the frequency response of fluorescence [20], harmonics of more than a few MHz are nearly completely filtered out. In general, the observable frequencies usually span a wide range from above 40kHz (to avoid interfering with infrared communication systems [21]) to 1 MHz.

2.2 Feasibility of FL Identification via Frequency Characteristics

Based on the operating principle of FL circuitry, *we hypothesize that individual FLs can be discriminated by their frequency characteristics*. In this section, we verify this hypothesis through comprehensive experiments, focusing on the uniqueness and stability of frequency features.

2.2.1 *Uniqueness of frequency characteristics*

Fig. 2 illustrates our measurement setup. We sample the FLs' optical emissions using a customized high speed light sensor comprising a BPW34 photodiode (PD) and a 2-stage amplifier, whose analog bandwidth is 1.6MHz and output digitized by a portable oscilloscope (PicoScope 2207A [22]).

Time/Frequency characteristics of FL's optical emission. Fig. 3(a) plots the time series of received signal strength (RSS) from several example FLs, including tube lights and CFLs in an office building. We observe that all the FLs' waveforms manifest quasi-periodic patterns, both at μs and ms scales. The periodicity becomes obvious in frequency domain (Fig. 3(b)). More specifically, all the FLs have a fundamental frequency component within the 40-60kHz range, followed by its harmonics (*i.e.*, integer multiples). Notably, the fundamental frequency of each light can differ by a few kHz, which implies the feasibility of discrimination. The 120Hz AC component (doubled from 60Hz by rectification) and its harmonics are also visible in the spectrum, but only span a few kHz in total.

It is worth noting that the dominate frequency always occurs at 2× the fundamental frequency, and has a 20-30dB higher magnitude. This is because the inverter creates different gains at the positive and negative halves of its sine waves, resulting in a weaker frequency component with half of the frequency [20]. As we will show in Sec. 4, when using a smartphone, it is usually impossible to see the features except for the strongest. Thus, *we select the dominate frequency as the* ***characteristic frequency (CF)*** *of the FL.*

Diversity of characteristic frequencies (CFs). To see how diverse the CFs can be, we measured over 500 tube FLs and compact FLs in our 4-floor office building. Fig. 4 plots the histogram of the CFs. The results imply multiple types of FLs, with most CF features ranging from 80 kHz to 160 kHz. Each type's CFs approximate a normal distribution. Fig. 5 further plots the CDF of pairwise differences in CF (denoted as Δf) among the FLs. We observe that only less than 0.1% of the pairs have $\Delta f \leq$ 10Hz, and 0.2% have $\Delta f \leq$ 20Hz, even among FLs of the same model. This confirms the CF as a strong feature to discriminate FLs at a large scale. In Sec. 5 and 7, we will provide analytical justifications for the probability of feature collision and implications for location discrimination.

We note that LEDs do not necessarily have CFs because they are driven by DC current (Sec. 8). Thus, existing LED based visible-light localization systems (*e.g.*, [10, 12]) have to employ customized drivers to modulate the LED lights, allowing them to emit digital bits as identify information.

2.2.2 *Stability of characteristic frequency*

Start-up profile of CF. To investigate the start-up process of the FLs, we cold start a tube FL in our office building, and measure its CF across 1 hour. Fig. 6 plots the CF (error bars indicate std. across 16 snapshots at each 1-minute timestamp). The frequency first overshoots by a few hundred Hz (likely due to transient behavior of the electronic components), and then gradually decreases as the FL warms up. After around 40 minutes, it converges to a stable CF, with less than 20Hz fluctuation. For most commercial place and office buildings, the internal FLs tend to keep on for a long time after the start-up, so the transient variation may not affect the robustness of CF as a discriminative feature.

Temporal stability of characteristic frequency. To verify the temporal stability of CF, we measure the CF of 8 randomly selected FLs in our office building over short/long term. Fig. 7(a) plots the CF values across 10 days, with 10 measurements per day per FL (error bars denote 90-percentile values). The CFs demonstrate high stability, with variation well below 20 Hz. Since the measurements span different time of the day, with different load on the powerline, the results also imply that the CF features are robust against different powerline loads in large buildings. We also evaluated the temporal stability over 15 weeks, sampling on a random day in each week. The results in Fig. 7(b) show slightly higher variation compared with the 10-day short-term variation, which also increases the risk of feature collision among FLs. As a result, *for robust localization*

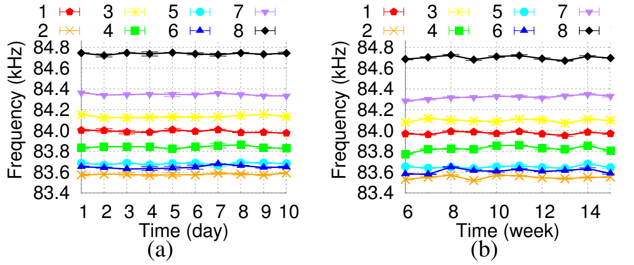

Figure 7: Short-term and long-term stability of characteristic frequency.

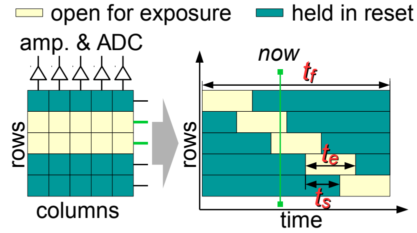

Figure 8: A brief illustration of rolling shutter.

using CF features, we must be able to tolerate small amount of collisions. In Sec. 5, we will introduce LiTell's error tolerance mechanisms.

3. CAPTURING HIGH-FREQUENCY FEATURES USING SMARTPHONES

Amid the photodiode sensors' potential in discriminating FLs' characteristic frequencies, fundamental challenges emerge when we try to capture such high-frequency features using smartphones, which are designed to capture static or low-rate scenes[1]. In this section, we describe how LiTell overcomes such challenges by exploiting camera as a generic optical channel sampler.

3.1 Sampling Light Using Digital Cameras: A Primer

Due to considerations in cost and speed, modern CMOS image sensors used in smartphones usually expose different rows in the image at different time, while pixels within each row are exposed simultaneously (which can be summed into a single *sample*). This is called the rolling shutter effect [23] and can act as a sampling process with much higher sampling rate than the frame rate. As a result, smartphones hold the potential for being used to sample high frequency light signals.

Fig. 8 shows a simplified diagram of CMOS image sensor and its sampling process. The image sensor's rows are opened for exposure sequentially. The delay between the opening of adjacent rows is called *sampling interval*, denoted as t_s. Duration of opening for each row is called *exposure time*, denoted as t_e. The time between the start of exposure of the first row and the end of the last row is the minimum frame time or *minimum frame interval*, denoted as t_f. Denote N as the number of rows, we have $t_s = t_f/N$. Inverting both sides, we can obtain the effective sampling rate of the camera with rolling shutter effect, which is $f_s = NR_f$.

Contemporary smartphones commonly support full HD capturing (1920×1080 resolution at $R_f = 30$ fps frame rate). The corresponding lower bound of sampling rate is thus $f_s = 1080 \times 30 = 32.4$ Ksps, which is far below the Nyquist sampling rate for typical FLs' characteristic frequencies ($> 2 \times 80$ KHz, Sec. 2).

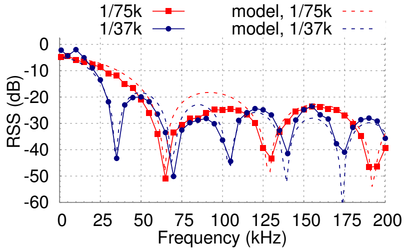

Figure 9: Frequency response of a camera under various exposure time settings.

Meanwhile, this sampling rate is far more reliable than the frame rate, since rolling shutter is controlled by hardware circuits instead of software running on top of an operational system (OS), which introduces significant amount of uncertainty and jitter. We have tested 2 Nexus 5 phones and found their sampling rate to be extremely stable and consistent, varying only a few ppm (parts per million, or less than 1 Hz error for 75 kHz sampling rate) in typical conditions. This makes the rolling shutter a reliable sampling mechanism.

3.2 Adaptive Exposure: Preserving High-Frequency FL Features

Modeling optical channel response of camera. The camera's sampling duration t_s depends on hardware and is unaffected by the exposure time setting t_e. In fact, $t_e \geq t_s$ since exposure of different row can overlap as shown in Fig. 8. However, the exposure process can be considered as an integration (or moving-average filter), so t_e does affect the analog channel response of the camera and determines its capability to capture high-frequency signals.

More specifically, at sampling time t (after the beginning of an exposure), the camera's output is:

$$A(t) = \int_{-\infty}^{+\infty} I(\tau)g(\tau - t)d\tau \tag{1}$$

where $I(t)$ is the light intensity at time t, $g(\tau - t)$ is a gate (rectangular) function that evaluates to 1 during $(t, t+t_e)$ and 0 otherwise. Applying Fourier transform on both sides, we have:

$$\begin{aligned}\mathcal{F}(A) = & \int_{-\infty}^{+\infty} I(\tau)e^{-j2\pi f\tau}d\tau \int_{-\infty}^{+\infty} g(-T)e^{-j2\pi fT}dT \\ = & -\mathcal{F}(I)\cdot\mathcal{F}(g)\end{aligned} \tag{2}$$

where $T = t - \tau$. Now we can obtain the frequency response of the sampling process as:

$$|H(f)| = \frac{|\mathcal{F}(A)|}{|\mathcal{F}(I)|} = |\mathcal{F}(g)| = |\,\mathrm{sinc}(\pi f t_e)| \tag{3}$$

In practice, photoelectron continue to accumulate during the readout phase after exposure is completed [15]. Accordingly, the integration time should be $t_e + t_r$, where t_r is the readout duration. So Eq. (3) should be updated as:

$$|H(f, t_e)| = |\,\mathrm{sinc}[\pi f \cdot (t_e + t_r)]| \tag{4}$$

Model validation and adaptive exposure design. To verify this frequency response model does match actual smartphones, we use the arbitrary waveform generator (AWG) on PicoScope to generate sine tones across a wide range of frequencies, which are then used to drive a white LED, serving as an ideal optical source. We capture the signals using a Nexus 5 phone, which has a sampling rate $t_s =$ 75.1624 Ksps according to our measurement (to be discussed in Sec. 3.3). Fig. 9 plots the RSS of the captured sine signals (relative to the DC baseline) across different frequencies, which reflects the frequency response of the camera.

The results reveal that under a given exposure time setting, the camera's response follows the sinc function, which matches the above model. Whereas Nexus 5's advertised minimum exposure

[1] Note that smartphones' light sensors cannot capture the CF feature because their sampling frequencies are limited to a few Hz.

time $t_e = 1/75586$ s, the first notch of the sinc function appears at around $1/65200$ s, *i.e.*, $t_e + t_r \approx 1/65200 > t_e$, which matches the refined model in Eq. (4).

Note that under a given exposure time setting, the camera's frequency response has notch points at $k/(t_e + t_r), k = 1, 2, \cdots$. If the characteristic frequency of an FL lies near a notch, then the frequency feature will become extremely weak. The *adaptive exposure* mechanism in LiTell is designed to overcome such situations. Specifically, LiTell selects the optimal exposure time t_e^* to maximize the camera's mean response to the characteristic frequencies of all FLs (*e.g.*, within the building of interest), *i.e.*,

$$\begin{aligned} t_e^* &= \arg\max_i \sum_{j=1}^{L} |H(f_j, t_e^i)| \\ &= \arg\max_i \sum_{j=1}^{L} |\mathrm{sinc}(\pi f_j \cdot (t_e^i + t_r))| \end{aligned} \tag{5}$$

where L is the number of FLs. t_e^i indexes the exposure time of the i-th exposure setting. Since all notches can be avoided within limited number of exposure time settings, the optimization can be solved by LiTell's backend server offline through an exhaustive search. Note that t_r is a constant for each camera and can be factory calibrated.

3.3 Aliased Sampling: Recovering High-Frequency FL Features

The foregoing modeling/measurement reveals that the frequency response of the camera remains high beyond the sampling rate. In other words, signals with frequency higher than $f_s/2$ can still be sampled, since the analog bandwidth of the camera is much higher than the digital sampling rate. LiTell leverages this property to recover high-frequency signals.

It is well known that when sampling a high frequency signal at sub-Nyquist rate, the frequency component will be aliased or folded back [16]. The rule of aliasing is simple:

$$f_a = \begin{cases} (N+1)f_s - f & f_s/2 < f - Nf_s < f_s \\ f - Nf_s & 0 \leqslant f - Nf_s \leqslant f_s/2 \end{cases} \tag{6}$$

where f_a is the aliased frequency, f is the original frequency, f_s is the sampling frequency and $N = 0, 1, 2, \cdots$. Given f_s, we can derive a few candidates of f from the measured f_a:

$$f \in \{f_g | f_g = Nf_s \pm f_a,\ N = 0, 1, 2, \cdots,\ f_g \geqslant 0\} \tag{7}$$

where f_g is a candidate estimation of f. For example, $f_s = 75$kHz and $f_a = 15$kHz leads to $f \in \{15, 60, 90, 135, \cdots\}$, and for $f_a = 3$ we have $f \in \{3, 72, 78, 147, \cdots\}$. In practice we find all FLs have f well above 40kHz and below 200kHz, so frequencies beyond the range can be excluded.

Whereas the error in f_a will be carried to the estimation of f as is, the error in f_s can get amplified in the process when N is large. Since the difference among FLs' CF features can sometimes be as small as a few tens of Hz (Sec. 2), this calls for a precise calibration of f_s. In practice, this can be done in factory since the sampling frequency of the camera is supplied by high quality clock sources and varies extremely slowly over time [24, 25].

In our experiments and implementation, we use PicoScope to send a 10 kHz optical calibration signal, keeping other settings identical as in Sec. 3.2. We then run MUSIC (an eigen spectrum analysis algorithm [26]) over the camera-captured signals to get a super resolution estimation of the received normalized signal frequency f_n ($f_n < 0.5$). The camera's sampling frequency is then calibrated as: $f_s = 10\text{ kHz}/f_n$.

4. AMPLIFYING HIGH-FREQUENCY FEATURES

Although it is feasible to sample the FL's high-frequency characteristic signals using a camera, the signals can be extremely weak — inherently, the FL is designed for zero-frequency illumination signals, and camera for snapshotting a still scene. In this section, we present a set of signal processing and camera optimization mechanisms in LiTell that together overcome these challenges.

4.1 Spatial Noise and Interference Suppression

Owing to the rolling shutter effect, LiTell can sample the same FL at high frequency across different rows of an image. This assumes the FL is an ideal homogeneous light source and fully occupies the camera's FoV. But two practical challenges can break this premise: *(i)* "Salt-and-pepper" noise in dark area, though barely visible to human eyes, may create substantial noise at high frequencies. *(ii)* Strong ambient light (*e.g.*, sunlight reflected to ceiling) may also raise the noise floor. In LiTell, we address such challenges by isolating irrelevant pixels from the camera image, which we refer to as spatial noise and interference suppression (*SNIS*). We now describe SNIS' major steps.

Cleaning up image for reliable contour detection. Unlike modulated LEDs in previous works [12, 15], LiTell faces two new challenges: *(i)* it must be able to isolate closely placed FLs, which is common in many buildings; *(ii)* it must be able to process million-pixel images on smartphones with minimum latency. These challenges make it difficult to follow the exact procedures in [12], which applies blur to mitigate noise, and then threshold the image into a black-and-white mask to identify the light. We thus threshold [27] the image into a mask first, and then use morphological opening and closing [28, 29] on the mask, a classical image processing algorithm that recovers a single shape by connecting densely distributed dots, while removing noisy outliers. After such step, the mask will only contain one or a few large connected components, which speeds up light contour detection.

Generating a rigid contour. The morphological processing yields a single shape which can be isolated by finding the largest contour, but it leaves one problem unattended: the shape is irregular, causing different number of pixels in each row to be summed into the sample. Consequently, the gain provided by summing varies across samples, which introduces high-frequency artifacts in the resulting spectrum. To solve the problem, we create a rigid bounding box around the FL image following [29]. Since most light fixtures appear rectangular on cameras, this method effectively minimizes inclusion of noisy pixels. It also ensures each row of the image has the same number of pixels, which provides similar summing gains. Otherwise, spurious frequency components may be introduced by gain variation across rows.

Speeding up processing by sub-sampling. Processing million-pixel images on a smartphone directly costs substantial time, which hinders real-time localization. To speed it up, we observe that the FL shape is quite regular and we can shrink the entire camera image without corrupting the contour of the FL. Therefore, we first subsample the image to 1/256 of its original size, before the thresholding and morphological processing. This allows LiTell's processing speed to grow from less than one picture per second to roughly 10 pictures per second while preserving necessary accuracy. After obtaining the bounding box, we scale up its size/position, and select the pixels inside its boundary for CF identification.

Fig. 10 summarizes the entire work flow of SNIS. Since we use RAW images for processing, all pictures are in grayscale. A side product of SNIS is a geometrical outline of the FL, which will be

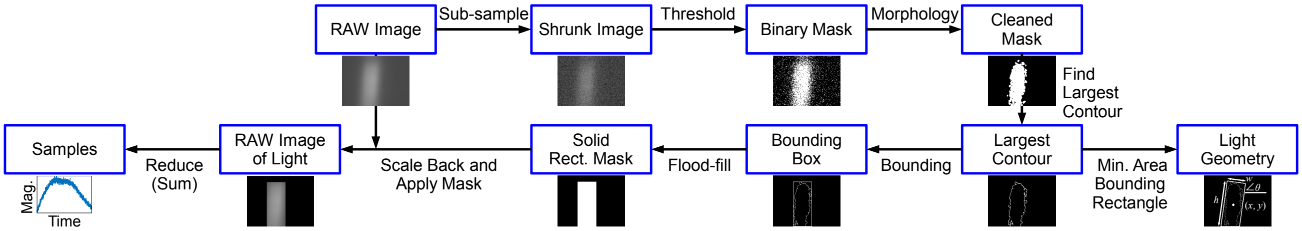

Figure 10: Reducing spatial interference by extracting FL related pixels in the RAW camera image.

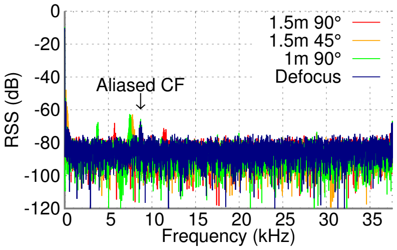

Figure 11: Effects of defocusing.

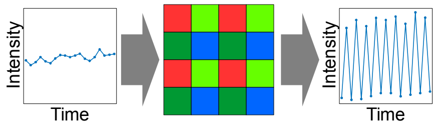

Figure 12: Color filter array and resultant gain mismatch between samples.

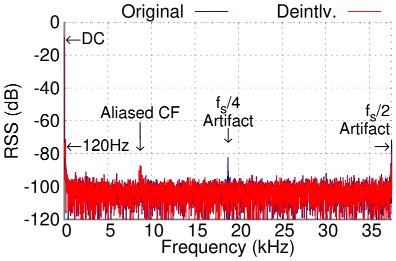

Figure 13: Effects of de-interleaving.

used for sub-light localization (Sec. 5.2). It also isolates the light fixture containing the most pixels in the picture, thus CFs from different lights will not get mixed up.

4.2 Optimizing Camera Configurations to Amplify High Frequency Features

Defocusing: isolating spatial frequency components. Many FLs come with shades or covers that bear certain spatial patterns, which creates noisy frequency components after the rolling shutter sampling. To remove the spatial artifacts, we can use a low pass image filter. Such a filter must only be applied to spatial features alone, *i.e.*, filtering the signal spatially before it enters the camera sensor. Otherwise, it will destroy the temporal frequency components. In LiTell, we use the camera's macro mode to force the focus distance to a minimum, thus defocusing the far-away FL by acting as a spatial low-pass filter.

Fig. 11 plots an example spectrum measurement of an office FL with patterned plastic covers. The spatial features result in multiple peaks that are much stronger than the CF, and vary depending on camera viewing angle and distance. With defocusing, the impact is almost completely eliminated.

Overcoming interleaving in CMOS sensors. A camera's image sensor typically comprises millions of light sensors. The sensors of different colors are interleaved with each other [30] (Fig. 12). In practice, an FL's color may not appear white to the CMOS sensor [31]. Thus, the image sensor's odd and even rows can have quite different responses to the FL's signal, which again translates into unwanted frequency artifacts. To counteract such gain mismatch, we first obtain the mean values of even and odd rows to estimate the different gains applied to each row, and then divide the rows with the corresponding gain. Fig. 13 shows the effectiveness of de-interleaving on a Nexus 5 smartphone. We see that the interleaving induces a $f_s/4$ frequency component, which is even stronger than the characteristic frequency, but completely removed after the gain compensation.

Optimizing ISO. A camera's low-noise amplifier (LNA) is controlled by ISO number. The higher the ISO, the higher the gain, and thus the more sensitive the camera is. Whereas a high ISO allows a camera to capture weak signals, it also creates more noise. In Fig. 14 we show how SNR of a tube FL's CF feature changes with different ISO settings. Overall, a higher ISO always results in better SNR, implying that the improved sensitivity outweighs the increased noise level. Thus, LiTell configures the camera to the highest ISO to maximize SNR. Note that this choice differs from LED based visible light communication systems, which intentionally generate high power flickering signals, sufficient to be captured by the camera even at low ISO [12].

4.3 Sequential Image Combining and Peak Frequency Identification

Amplifying SNR via sequential imaging combination. After the foregoing processing of a camera image, we obtain a column vector, with each element being the sum intensity of a row of pixels. To identify the peak frequency, we can simply run FFT over the vector of samples. However, we found this results in a low SNR of only around 2 dB under typical conditions.

To boost the SNR, we capture multiple camera images, and concatenate their samples into a long sequence as FFT input. Image concatenation brings two immediate benefits: *(i) Higher frequency-resolution.* Suppose f_s and N denote the sampling frequency and number of samples, respectively. Denote L as the length of samples per image, which may deviate from maximum image height due to SNIS. FFT operation's resolution equals $f_{\text{bin}} = f_s/N$ [16]. To discriminate the characteristic frequency of different FLs, our empirical results show that *at least* 10Hz resolution is needed (Sec. 5). So we need to concatenate at least $f_s/10/L$ images, considering SNIS may remove some noisy samples. For example, since Nexus 5 has a sampling rate of roughly 75Ksps and each picture provides about 2000 samples (Table 1), we concatenate 7 pictures to get a sufficient resolution of around 5Hz. *(ii) Higher processing gain.* It is well known that FFT can achieve a "processing gain" with more samples, *i.e.*, the peak frequency's SNR increases logarithmically (in dB) as N increases, assuming the noise power spread evenly over the spectrum (*i.e.*, white noise). Our experimental results, shown in Fig. 15, indeed verifies this quantitative relation. However, we note as N becomes large, SNR improves marginally whereas the image processing time increases substantially. We thus choose N to be the minimum value that satisfies the resolution requirement.

One caveat in image concatenation is that overwhelming discontinuities occur near the edge (*cf.* output samples in Fig 10), which again induce high frequency artifacts that contaminate the CF feature. We identified two underlying reasons: *(i)* Camera lens vignetting, which makes outer area of the image darker than the central areas; *(ii)* smartphone not perfectly parallelizing the ceiling, resulting in

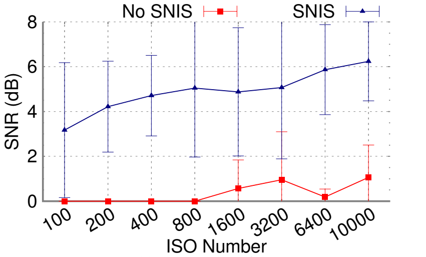

Figure 14: Effects of ISO number, with and without SNIS.

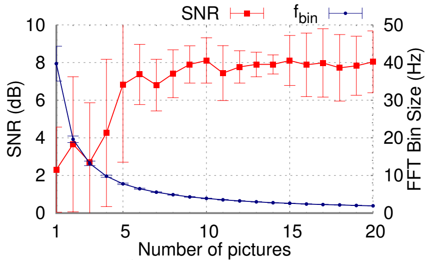

Figure 15: SNR improvement via image concatenation.

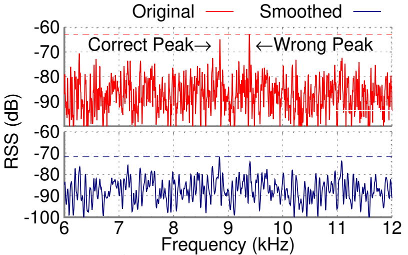

Figure 16: Spectrum smoothing example, dashed lines show maximum.

one end of the image brighter than the other. We use a simple equalization procedure to mitigate these effects. Specifically, we first fit each series of samples to a 6-order polynomial function, which is common in lens correction and can capture slow varying spatial response. Then we divide the samples element-wise by this polynomial function since the distortions are essentially multiplicative scaling. This restores the samples to their undistorted form and avoids huge jumps at concatenation points.

Identifying characteristic frequency in the spectrum. After the foregoing processing on the image samples, most spurious peaks have been removed, making aliased CF the highest peak. We then run a 2-step search to identify the peak frequency. We use Savitzky-Golay filter [32] on the spectrum to get a smoothed spectrum, identify a coarse peak region, and then search for the exact peak position within the corresponding region inside the original spectrum. The 2-step approach helps preventing false positives when SNR is marginal. Fig. 16 shows an example, where without filtering a spurious peak can be misidentified as CF.

5. FEATURE MATCHING AND LOCALIZATION

LiTell uses a simple linear search algorithm to match the smartphone-extracted CF feature with CF fingerprints in the database. Specifically, after obtaining an aliased copy of the CF, f_a, we first derive a collection F of possible original CFs following Eq. (7). For each $f_g \in F$, we find the fingerprint with minimum difference in CF. Finally, the f_g with minimum matching distance is considered as the FL's CF.

To guarantee accuracy, LiTell further uses history information to provide tolerance against CF feature collision. Further, we show that LiTell can also provide sub-light localization accuracy whenever an FL falls into the camera's FoV.

5.1 Multi-Light Matching: Improving Accuracy

To ensure high accuracy in location matching, we leverage the fact that the user typically passes multiple lights consecutively, allowing LiTell to use a group of consecutive lights together as a location landmark as the user travels. From a high level, *multi-light matching provides an error-correction mechanism for LiTell— the strict location relation between FLs dictates that each FL only has a few candidate neighbors*. Thus, even if one FL within a group is mistaken, it may not fundamentally change the Euclidean distance (in feature space).

Asymptotic collision probability. To understand the asymptotic benefit from multi-light matching, we first analyze the probability that a group of m consecutive lights is not colliding with another group. As discussed in Sec. 2.2.1, we can reasonably assume frequency of all N_l lights follow the same normal distribution, *i.e.* $f_n \sim N(\mu, \sigma), n \in \{1, 2, \ldots, N_l\}$. A Gaussian fitting shows that the 2 types of lights have $\mu = 90.25$ kHz, $\sigma = 1.62$ kHz and $\mu = 138.25$ kHz, $\sigma = 2.34$ kHz, respectively. Consider the scenario when one of 2 groups has CFs very close to the mean value μ (*i.e.*, $f_{n,1} \approx \mu$). Since in normal distribution probability density peaks at μ, this represents the worst case scenario where the collision probability is the highest. The Euclidean distance between their characteristic frequencies equals:

$$D = \sqrt{\sum_{i=1}^{m} \Delta f_{ni}^2}, D^2 = \sum_{i=1}^{m} \Delta f_{ni}^2 \tag{8}$$

(D^2/σ^2) follows $\chi^2(m)$ distribution since $\Delta f_n = f_{n,1} - f_{n,2} \sim N(0, \sigma^2)$. For LiTell to distinguish 2 groups, we need $D > a\sqrt{m}$, or $(D^2/\sigma^2) > (a\sqrt{m}/\sigma)^2$, where a is the tolerance, *i.e.* minimum distinguishable frequency difference. Denote $P(k, m)$ as the CDF of $\chi^2(m)$ distribution, then the probability of 2 groups' feature not colliding with each other equals:

$$P_g(m, a) = 1 - P\left[m\left(\frac{a}{\sigma}\right)^2, m\right] \tag{9}$$

For the case of pairs ($m = 2$), $P_g = 1 - P\left[2(a/\sigma)^2, 2\right]$. Further, the worst-case probability of one particular pair *not* colliding with *any* other pair is:

$$P_n(N_p, a) = \prod_{i=1, i \neq n}^{N_p} P_g(2, a) = P_g(2, a)^{N_p} \tag{10}$$

where N_p is number of consecutive pairs and usually $N_p < 4N_l$ since each light can at most make 8 pairs at 8 directions (including diagonals), and each pair is shared by 2 lights.

When $m = 1$, D reduces to normal random variable Δf and $P_n = \left\{\text{erfc}\left[a/(\sqrt{2}\sigma)\right]\right\}^{N_l - 1}$. The asymptotic collision probabilities for single light and pair matching under 10 Hz and 20 Hz tolerance for $\sigma = 1.62$ kHz are shown in Fig. 17.

Choosing group size. *In LiTell, to avoid excessive sampling, we set $m = 2$, i.e., using each consecutive light pair as location feature.* Yet even this small m is sufficient to achieve high accuracy. For example, with the same parameter $a = 0.01, \sigma = 1.62$ and the worst case scenario for pairs $N_p = 4N$, Eq. (10) gives $P_n \approx 98.5\%$ for a region of 100 lights, and $P_n \approx 97.0\%$ for a region of 200 lights.

Empirical validation. We verify the effectiveness of single-light and light-pair patching in an office building. The database contains each ceiling FL's coordinate, and characteristic frequency measured using the photodiode (Sec. 2). We first evaluate the single-light matching by testing over 4 regions containing 22, 28, 33 and 36 FLs, respectively (Fig. 18). The non-collision probability turns out to be 86.4%, 81.5%, 82.9% and 80.5%, respectively, slightly lower than the asymptotic analysis. In contrast, with light-pair matching, across 3 regions with 28, 64 and 119 lights, we achieve an accuracy of 92.9%, 91.7% and 90.8%, much higher than single-light matching. The results are summarized in Fig. 19.

We emphasize that LiTell can distinguish which light in the pair the user is currently at as long as CFs for the 2 FLs in the pair are different, thus matching pairs will not comprise localization granularity.

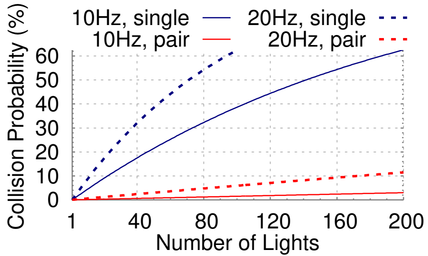

Figure 17: Asymptotic collision probability of single and pair matching, under different tolerances.

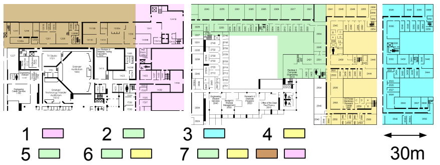

Figure 18: Map of small-scale test areas. Area numbers are followed by regions covered.

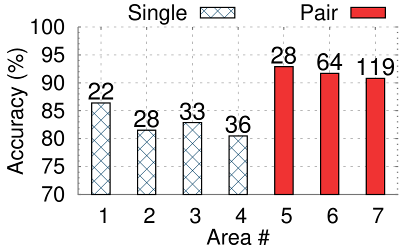

Figure 19: Measured non-collision probability of single and pair matching. Numbers show number of lights in area.

Also note that LiTell uses a pair of characteristic frequencies as a single landmark, so wrong localization results will not propagate across different pairs.

5.2 Achieving Sub-light Granularity

Cameras are designed to produce undistorted images. From geometry relations in the image, we can derive information that enables sub-light level accuracy, which decouples LiTell's granularity from the density of FLs. To simplify the problem, we focus on the case when user holds the smartphone roughly at level position in parallel to the ceiling fixture. Then, the center of the camera image C corresponds to smartphone location, since it points to the smartphone's projection on the ceiling.

We first observe that the geometry of lighting fixtures can be easily known and added to LiTell's database. Meanwhile, the SNIS in Sec. 4.1 can provide the geometry of the light in the image. This allows us to establish the relation between physical size and number of pixels in the image as a ratio g_i. Thus, we can get the smartphone's position in terms of pixels and then map it to physical location on the 2D plane.

Fig. 20 illustrates a smartphone's FoV which captures an FL but is not perfectly aligned with it. We use w, h and θ to denote the width, height and tilt angle of the FL's image. L denotes the center of the FL, whose physical location is known. C and P denote the center of the image and its projection onto the light's axle. The coordinate of L, C and P are denoted as (x_l, y_l), (x_c, y_c) and (x_p, y_p), respectively. By definition, $\mathbf{LP} \perp \mathbf{PC}$, thus $\mathbf{LP} \cdot \mathbf{PC} = 0$. In addition, $\tan(\theta) = (y_p - y_l)/(x_p - x_l)$, and (x_l, y_l), (x_c, y_c) are known. Consequently, we have 2 equations to solve 2 unknown variables x_p and y_p, which in turn lead to $|\mathbf{LP}|$ and $|\mathbf{PC}|$ in terms of pixels. We then translate them back to meters with g_i. The resulting physical coordinate of C gives the smartphone's position relative to the center of the light.

6. IMPLEMENTATION AND SYSTEM EFFICIENCY

CF fingerprinting. We use the photodiode setup (Sec. 2) to collect 2 seconds of samples from each FL and identify its ground-truth CF. We then manually mark the FL's location on a map, whose zero coordinate is defined at an anchoring FL near the building entrance. The map is later digitized into ⟨*location, CF*⟩ and entered in LiTell's database backend. The whole fingerprinting process takes one student volunteer around 3 hours for a medium-sized grocery store ($\approx 1000\text{m}^2$, 162 FLs), which is already an order of magnitude faster than radio-based fingerprinting [2]. Overall, *LiTell's fingerprinting process is simple, efficient and non-intrusive, and involves no management/infrastructure cost in realistic environments.*

Receiver (smartphone) side implementation. We have implemented LiTell's sampling (Sec. 3) and feature amplification mechanisms (Sec. 4) on Android. Further, we prototype a simple navigation app that finds the shortest path towards a destination FL, and navigates users with basic on-screen instructions (*e.g.*, forward, left). Our experiments run on several popular Android smartphones, whose camera capabilities along with f_s (measured following Sec. 3.3) are summarized in Table 1. Unless noted otherwise, we use Nexus 5 for testing, with default ISO 10000, 7 image samples per FL, and exposure time set by the adaptive exposure scheme (Sec. 3.2).

Sever backend. We run a MATLAB backend that maintains a spreadsheet as database. Rather than sending RAW images to the server (which makes wireless network the bottleneck), LiTell's smartphone app first processes the image following Sec. 3 and 4. Upon receiving the processed samples, the server extracts aliased characteristic frequency (CF) f_a (Sec. 4.3), generates possible unaliased CFs (Sec. 3.3) and match them to a specific FL using the pair matching algorithm (Sec. 5.1). It then returns navigation instructions to the smartphone app. In case the matching confidence is low, the app will instruct the user to move to the next light for rematching.

Energy consumption. We profile the energy consumption of the LiTell app using the Monsoon Power Monitor [34]. The Nexus 5 consumes 741.84 ± 11.52mW when idle, and 2001.60 ± 28.50mW when the camera's viewfinder is active. When the LiTell app is capturing, processing and sending simultaneously, the power consumption is 2669.69 ± 29.47mW, but a localization attempt only lasts less than 2 seconds and consumes 1.20 ± 0.01mWh energy. Considering that Nexus 5 has a 8.74Wh battery [35], it may support over 1000 localization attempts per battery cycle.

Computational load and latency. On Nexus 5, our LiTell implementation uses OpenCV [36] to process the images for SNIS (Sec. 4.1). Processing each full-resolution RAW image takes 92 ± 35ms, and around 700 ms for 7 images. We find that a majority of the processing time is due to suboptimal implementation (Java instead of native C) of the gluing operations, which cost around 70ms per image. We hence believe that with proper optimization, the computation can be done within one frame interval (33ms [37]), allowing 7 pictures to be processed within 233ms.

For the database server, including frequency estimation and matching (Sec. 3.3, 4.3 and 5.1), all computation takes 112ms per request on average, with 138 MB memory usage while running on a Dell Latitude E5430 laptop with Intel Core i5 3340M CPU. Experiments in our office building show an average location query-response time of 1.62s in total, which includes the app's processing time, database server, network latency and other system level overhead. Also, it is feasible to migrate the server side processing entirely into the app by preloading the database or loading the database from storages such as NFC tags, allowing localization when network connection is unavailable, or when location privacy is of utmost concern.

7. EXPERIMENTAL EVALUATION

LiTell's localization performance depends on two main factors: *diversity* of characteristic frequencies (CFs) across FLs, and *stability* of the CF feature in each FL. Our foregoing discussion has profiled

Table 1: Summary of camera parameters for the Android phones in our experiments.

Phone	Year	Resolution and Framerate	Estimated f_s (Sec. 3.1)	f_s (Sec. 3.3)	SNR (Sec. 7.1)
Nexus 4	2012	3280 × 2464 (8MP) @ 22.5 fps	55.440 Ksps	56.3347 Ksps	4.24 dB
Moto X [a]	2013	4320 × 2432 (10MP) @ 30 fps	72.960 Ksps	60.6981 Ksps	9.20 dB
Nexus 5	2013	3280 × 2464 (8MP) @ 30 fps	73.920 Ksps	75.1624 Ksps	10.26 dB
Nexus 5X[b]	2015	4032 × 3024 (12MP) @ 34.97 fps	104.910 Ksps	105.1089 Ksps	6.12 dB

[a] Sensor might be down-clocked. The best configuration available in its software is 1920 × 1080 @ 60 fps, corresponding to 64.800 Ksps.
[b] This phone has non-linear RAW output (gamma compression [33]) and its SNR value is not comparable with others.

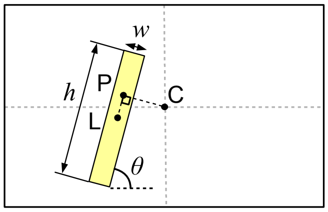

Figure 20: Sub-light localization scheme in LiTell.

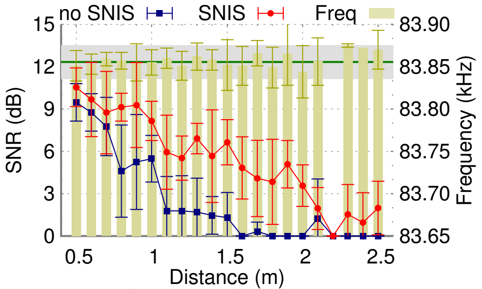

Figure 21: SNR (lines/points) and CF (yellow bars) detection accuracy.

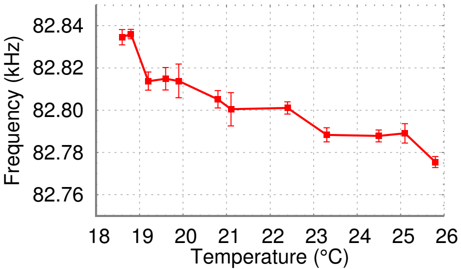

Figure 22: Impact of room temperature.

the diversity factor and its implications on location discrimination (Sec. 2 and 5). In this section, we first conduct experiments to verify the stability factor based on LiTell's smartphone-side algorithms (Sec. 3 and 4), and then use field tests to evaluate LiTell's overall performance.

7.1 Mirobenchmark Experiments

Robustness across different light-to-camera distances. We first put a smartphone under a single FL, vary the light-to-camera distance (measured using a laser ranger [38]), and measure the SNR and CF. Each measurement repeats 5 times. Fig. 21 shows that SNR decreases proportionally to distance, but remains high ($>$ 3 dB) even at a distance of 2m. Fig. 21 also plots the 90-percentile CF variation under different noise conditions, with dark green line showing ground-truth CF and gray area showing 20Hz tolerance (Sec. 5). As long as the SNR exceeds 3 dB, the variation stays within the 20Hz tolerance. This implies that *LiTell can robustly capture the CF features for a light-to-camera distance of up to 2m, which is longer than ceiling-to-smartphone heights in most cases*.

It is worth noting that SNIS nearly doubles the usable range for LiTell. In some cases, we found the spectrum smoothing in Sec. 4.3 helps identify CF even under negative SNR, which indicates its importance when SNR is marginal. Moreover, we found when LiTell misidentifies CF, it tends to pick up a random frequency with low confidence (far away from all fingerprints), and thus gets rejected.

Robustness across temperature. Recall an FL's electronic component ratings and hence stability of its CF is affected by temperature. We evaluate the impact in a small office, where we control the temperature settings via a thermostat and verify the change using a DS18B20 sensor [39] ($0.1°C$ precision). Fig. 22 shows that the CF decreases as temperature increases, which is consistent with the heat-up behavior in Fig. 6. However, within 4°C temperature gap, the CF fluctuates by less than 20Hz, well within LiTell's stability tolerance (Sec. 5). Since the temperature in most public buildings tends to be tightly regulated, this experiment implies that *LiTell is robust across typical temperature changes in indoor environments*. Our field tests will provide more evidence (Sec. 7.2). Meanwhile, we also tested LiTell in an outdoor parking ramp and do find the CF becomes unusable due to large ($>$ 10°C) air temperature variations.

Robustness across human behavior. We also verify that LiTell can work consistently across different users, whose usage behaviors may vary. In the experiment, we sample an FL from various directions and while walking. To cover extreme scenarios, we hold the smartphone within 1m to the FL, so the viewing angle can change rapidly with position. Then we create the following test cases: *(i) normal test case:* the phone is placed level right under the FL and aligned with its tube direction. *(ii) rotate case:* the phone is misaligned with the tube by about 30°; *(iii) tilt axial / lateral case* ("T. Ax." and "T. Lat."): the phone photos the fixture sideways from the axial and lateral direction of the tube; *(iv) walking axial / lateral case* ("W. Ax." and "W. Lat."): the user walks across the FL along its axial and lateral directions.

Fig. 23 shows that LiTell successfully detects the CF in all the cases, with 90-percentile deviations of less than 10 Hz. The SNR is consistently above 3 dB with a large margin. This shows that *LiTell's CF detection and amplification mechanisms make it robust to the extra variation on light intensity and effects caused by elusive usage behaviors*. More specifically, the sampling and processing work fast enough to ensure the CF remains visible during capture of 7 images. LiTell's SNIS is also robust against placement and shape, and the macro defocusing mechanism can successfully filter spatial features, leaving only the temporal frequencies in the spectrum.

Different smartphone models. LiTell's CF identification performance may be affected by different smartphone models, with varying resolution and speed (which determine the rolling-shutter sampling rate) and sensor quality (which determines SNR). Whereas it is infeasible to exhaust all smartphones, we test 4 of them representing different generations that were launched in the past 4 years, by sampling a tube FL 1 meter away. Table 1 summarizes the measurement results. Despite the different camera capabilities, all the phones can detect the CF with high SNR ($>$ 6 dB except for Nexus 4). It is worth noting that Nexus 4 and Moto X do not allow exposure time or ISO configuration, and only allow JPEG imaging which may distort the CF features due to non-linear processing [40]. The SNR of detected CF is relatively lower, but still exceeds LiTell's 3 dB minimum requirement, based on LiTell's other camera optimization mechanisms.

Impact of ambient light. To evaluate LiTell's robustness against ambient light interference, we put the smartphone 1.5m away from

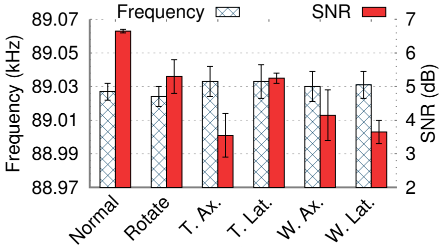

Figure 23: Impact of human behavior.

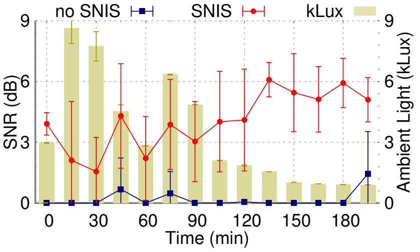

Figure 24: SNR variations with ambient light intensity.

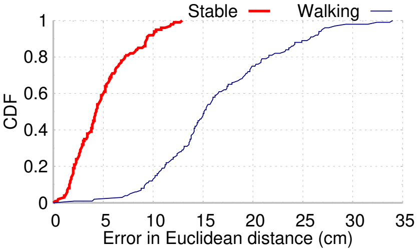

Figure 25: Performance of sublight localization.

Figure 26: Field test venues. From left to right: office building, parking lot, grocery store.

a tube FL close to a window. The experiment started on a sunny day at 2:30pm until sunset. A Lux meter [41] is placed nearby to track ambient light intensity. Fig. 24 shows that LiTell consistently captures the CF with more than 3 dB SNR except for extreme cases. In contrast, when LiTell's SNIS is disabled, the SNR drops to 0 dB under sunlight interference, making it impossible to discriminate CF from noise. Therefore, *SNIS not only improves SNR, but also makes LiTell more robust under ambient sunlight interference.*

Granularity of sub-light localization. To evaluate the granularity of sub-light localization, we use a light fixture with a 1.2m-long tube FL, 1.8m above the phone. We place the phone on a motored slider [42] and move it to 20 locations 5cm apart from each other in a 1.1m × 0.4m area (limited by slider length and camera FoV). We record the localization error of 10 attempts, and then repeat the experiment with a user holding the phone while "walking" on the spot. Fig. 25 shows that LiTell can achieve 10cm accuracy 90% of the time if placed still and level. When held by a walking user, the phone is no longer strictly level, which affects LiTell's geometrical model (Sec. 5.2). But the median precision is still as high as 15cm and 90% at 25cm. We believe the accuracy can be further improved when combined with phone orientation sensing and compensation mechanisms.

As the distance gets larger, the physical distance that a pixel covers also increases linearly, so the error will grow linearly with distance. Considering most buildings have heights less than 5m and users usually hold the smartphone at least 1m above the floor, the error of sublight localization should remain below 0.5m in most cases.

7.2 User Study and Field Tests

To test LiTell's robustness and accuracy in the wild, we conducted field trials across 3 different venues (Fig. 26): an office building ($\approx$ 9000m^2, 119 FLs, ceiling height $\approx$ 3.0m), an indoor parking lot ($\approx$ 2800m^2, 91 FLs, ceiling height $\approx$ 2.8m), and a medium-sized grocery store ($\approx$ 1000m^2, 162 FLs, ceiling height $\approx$ 4.0m). Our field trials were conducted approximately two months after FL fingerprinting, and span over 2 weeks. We recruited 10 volunteer users who frequent the venues, but may not have precise knowledge of all the points of interest (*e.g.*, room number, parking spot, and storage shelf of goods). Average height of the users is around 1.7m. The trials are conducted across different time of the day, with heavy human activities around most of the time. In each test, we ask a participant to start from the building entrance, and find a random POI which may be on a different floor. We perform 3 sets of trials: *(i) Controlled test:* A user finds the POI with common sense or help available (maps on walls, direction signs and people walking by). *(ii) LiTell test:* A user (different from the controlled one) uses the LiTell navigation app to find the same POI. *(iii) Oracle baseline*. After the above two tests, we let all users know the shortest path and walk directly to the POI.

In each test, we follow the user and record the walking time and path on a map. Then, we derive the *extra distance and percentage of extra time spent on navigation for test (i) and (ii), by subtracting the oracle baseline from them.* We use percentage in walking time to compensate for different walking speeds. Since the prototype LiTell implementation uses back cameras, the users sample the light with the phone facing down when requested by the app, and then turn the phone upright again to read instructions, which causes some unnecessary overhead in time consumption.

7.2.1 Accuracy

Our field trials focus on accuracy of FL identification. This is because most public indoor venues have densely deployed FLs (1.5 to 3 m separation), which allows us to achieve meter-level granularity, sufficient for navigation purpose even without the sub-light localization module. Fig. 27 plots the localization accuracy. Across all randomly selected POIs, LiTell can discriminate the FLs along the way, with mean accuracy of 90.3% and a small variation of 11% from the best to the worst, which is very close to results obtained in controlled environments (Fig. 19), despite different light models and ceiling heights. In particular, FLs in the grocery store are closely placed in lines 2 meters away from the phone, resulting in multiple lights being captured simultaneously. However, LiTell's SNIS mechanism can successfully isolate the lights, and only include the nearest one, which has the largest area in the image. Additionally, busy customers in the grocery store and moving cars in the parking lot did not degrade LiTell's accuracy, since the CF feature is deterministic even when RSS varies significantly. The results also show temperature in most indoor environments is stable enough for LiTell to operate, and verify that *LiTell is consistent and reliable in multiple types of realistic environments*, with negligible deployment efforts [2].

We also notice that most errors are caused by single light sampling, which are accidentally introduced by users. When the location matching confidence is low, LiTell advises the user to sample a single neighboring light to get the correct result. Thus, *occasional location errors do not actually cause navigation failures*.

7.2.2 Performance for Indoor Navigation

To test LiTell's effectiveness in indoor navigation, we randomly select 6 targets in the office building: 1, 4 and 5 are on the 1st floor, while 2, 3 and 6 are on the 2nd. In addition, 5 and 6 are in a corner

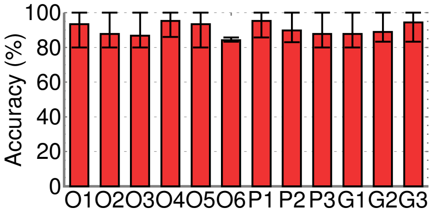

Figure 27: Accuracy across all venues. "O": office building, "P": parking lot, "G": grocery store. Error bars show min/max values across 10 users' trials.

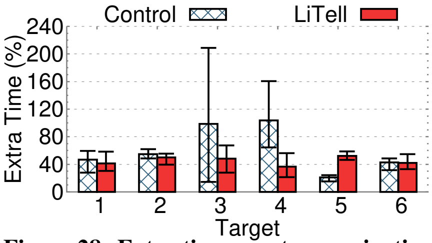

Figure 28: Extra time spent on navigation by users. Error bars show min/max values.

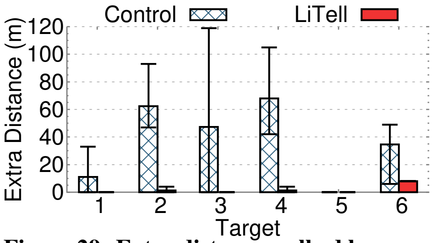

Figure 29: Extra distance walked by users. Error bars show min/max values.

of the building not far from the entrance. The oracle routes vary from 35m to 132m including the stairs, each containing at least one turn.

Fig. 28 shows the extra time cost *w.r.t.* the oracle. Compared with the controlled tests, LiTell saves different amount of time depending on how hard the target can be found. Particularly, target 5 is only about 35m away from the entrance point, and even the controlled group can easily find it with only 7 seconds of extra time compared with the oracle baseline. However, when the user is unfamiliar with the target POI, LiTell can save over 50% time on average compared with the control group, for both single-floor and cross-floor navigation. On the other hand, LiTell's extra time cost *w.r.t.* oracle is consistent at roughly 40% across all targets, regardless of distance, usage habits, area, number of turns or floors. This again confirms that *LiTell's CF extraction mechanism is robust against practical usage scenarios*.

Fig. 29 further plots the extra distance cost *w.r.t.* the oracle. Extra distance needed by LiTell is typically only a few meters, owing to its high localization precision. Interestingly, we find even for the control group users who frequent the building (for classes), their trajectory can be highly suboptimal. And LiTell can help prevent them from traveling to wrong places or along detour paths. The average distance saving can be 50 m to 70 m for POI 2, 3 and 4, and up to 120 m for certain users. Compared with such distance saving, the time saving in Fig. 28 might seem relatively small. This is mainly due to LiTell's processing overhead in each localization attempt. We believe an optimized implementation (Sec. 6) can cut the time overhead and enable seamless real-time navigation.

8. LIMITATIONS AND FUTURE WORK

Ubiquity of FL vs. LED. While LEDs are likely to replace FLs eventually, current progress is slow. The main reason is that FL has already saturated the market, and its low cost and availability still lead to new deployment. Moreover, LED hardware still bears certain limitations, such as lower color quality, narrow beam angle and lower lumen output [43]. Thus, the US Department of Energy [14] predicts that FLs will continue dominating the market, occupying more than 60% of the market till 2020, and 30% to 40% till 2030. From our experience, buildings with any LED lightings remains scarce for now, and even in these buildings the major force of lighting is still FL. In addition, LEDs used for commercial lighting may differ from "smart" bulbs that can send unique beacons. Due to cost, size and management issues, we expect "smart" bulbs and regular LED bulbs or even FLs to be deployed together in the future. This means systems that work with regular LED bulbs will still have unique values.

Most existing LEDs are driven by constant current drivers [44], which oscillate much like FL drivers [45]. Since LEDs are driven by DC, manufactures may add capacitors to the output of the driver to suppress flickering introduced by the AC mains voltage. This, however, apparently attenuates the characteristics frequency (CF) even more severely. From our measurements, some LEDs exhibit CFs that are 60dB weaker than the 120Hz and DC signal, while others do show very strong CF. Due to the limitation of SNR, current version of LiTell may not be able to work with all varieties of LEDs. However, once high-SNR, high-speed light sensors (*e.g.* photodiodes used in visible light communications (VLC) [46]) are available on smartphones, LiTell will also work with majority of the LEDs.

Front-facing cameras. Current LiTell implementation relies on back cameras, which are usually more powerful than the front-facing ones. We expect LiTell will need to concatenate more images to overcome the lower resolution and quality of front cameras, and eventually reach sufficient SNR. However, driven by popular applications such as selfie, the specifications for front-facing cameras have improved dramatically over past a few years, jumping from 720p to full-fledged 5-mega-pixel cameras. With up-to-date hardware, LiTell should be able to perform as well with such cameras.

Front cameras of modern smartphones can already capture pictures at nearly 30 FPS as needed by video calls, so collecting enough number of samples should not require substantially longer time, while the time spent on processing may reduce due to reduced number of pixels. As a result, using the front cameras should not increase LiTell's localization latency substantially.

Compensating for temperature. LiTell is unsuitable for environments where temperature varies dramatically across days, which changes the CF feature accordingly (Sec. 7.1). Currently, we see offices and shopping centers as LiTell's target application scenarios, where room temperature is always tightly regulated. For example, during a period covering winter and early summer, CFs of FLs in our office building did not drift significantly enough for recalibration to be necessary. However, for places with large dynamics in temperature, it is possible to derive a CF-vs-temperature model for each FL model, and use it to calibrate the fingerprinting database based on current temperature. Since LiTell mainly targets stable indoor environment, we leave such exploration as future work.

Energy consumption. We choose to build LiTell on top of the camera due to the fact that nearly every smartphone today has built-in camera, while none is known for embedding photodiodes or other high-speed light sensors. This leads to unnecessarily high power consumption since information from most pixels in the camera is discarded and wasted. The availability of such high-speed light sensors on mobile devices, however, may see growth in the future as VLC techniques mature. LiTell can easily be adapted to leverage photodiodes as high SNR, alias-free and energy-efficient light sensors. On the other hand, even though cameras are energy-hungry, LiTell can use low-speed, low-power and ubiquitous ambient light sensors to determine when to turn the camera on, which may help reduce overall energy consumption considerably.

Towards large-scale and long-term deployment. While LiTell's location matching mechanism (Sec. 5.1) gives it some tolerance to collisions, the probability of collisions among group of lights will eventually be too high when number of lights gets larger and larger. One solution is a multi-level hierarchical localization system: radio-based (*e.g.* Wi-Fi) localization system provides a rough region which limits LiTell's collision domain, and then LiTell provides fine-grained localization result within each domain.

Across our nearly half-year observation, the CFs of experimented FLs did not change significantly. But over a longer term, the CF change might eventually happen, *e.g.* due to severe aging or replacement of worn-out lights. When such change does occur, LiTell will either give low confidence in matching (with single or pair matching), or find user's trajectory to be discontinuous. Users may choose to allow the localization server to analyze such matching confidence, and then isolate and correct for the changes.

9. RELATED WORK

Radio and sensor based localization. Fingerprinting based approach is arguably the most accurate way to realize radio-based localization [8]. Using RSS [3, 4] or more detailed physical-layer information [47], the location granularity can reach 1 ft^2. However, suffering from environment changes, human presence/mobility, *etc.*, RF fingerprinting is highly unreliable – typical 80-percentile error fall in 6 to 8 meters [7, 8]. More importantly, the training overhead is unbounded because under multipath effects, nearby locations do not have a deterministic signature correlation. For example, latest WiFi fingerprinting method needs to employ a robot to survey each 1 ft^2 location spots for 100 times to average out signature variations [47]. The fingerprints will soon become stale in practical environments [4]. The daunting deployment/maintenance overhead prevents fingerprinting from being adopted.

Model based approaches can avoid the labor-intensive fingerprinting. Existing work along this direction has employed statistical propagation models [5, 48], crowd sourcing [49], as well as lateral sensing modalities such as motion sensors [50]. The location precision varies depending on testing environment, anchor density, *etc.* But due to inherent vulnerability of RF signals, the robustness problem remains unsolved [2].

Visible light localization. Visible light (VL) localization has been fueled by the growing popularity of LED luminaries, and the emerging VL communication standard 802.15.7 [51]. Earlier works [52–54] used light fixtures as landmarks to achieve cell-level location precision. State-of-the-art VL localization has further exploited colocated LEDs [12, 55], spin-lights [56], polarized LEDs [11], and a combination of light and motion sensors [10] to achieve sub-meter location precision. However, all these schemes need customized LED drivers to emit identity beacons, which further increases LEDs' hardware/retrofitting cost – a major roadblock to the adoption of LEDs [14]. High precision VL localization schemes often require well-shaped LEDs [10, 55, 57] (for propagation modeling) or ultradense deployment [10] (multiple LEDs within camera FoV), which limits their use cases. In [58], Xu *et al.* proposed to use LEDs as periodic landmarks to complement pedestrian dead-reckoning. However, the accuracy of dead-reckoning becomes the performance bottleneck and the error often rises to 8-10 meters.

Upgrading existing mobile handset with new hardware is a challenging problem. Therefore, many systems use smartphones as receiver for VL localization [12] and communication [15, 59]. These systems aim to detect low-rate beacons (a few bytes/second) from LEDs, and can be realized by directly processing the rolling-shutter pixels. In contrast, LiTell imposes stringent requirements on the sampling rate and SNR, which need new mechanisms to optimize the capture settings. It leverages the inherent signatures of FL luminaries and needs no dedicated VL communication hardware. Many existing video sampling algorithms take advantage of rolling shutter effects to achieve super-time resolution, but they commonly rely on customized camera hardware, which can configure exposure/readout time at pixel granularity [60].

Electrical appliance identification. LiTell's key idea was inspired by research in electromagnetic interference (EMI) detection [61]. When interfacing with the 50/60 Hz powerline signals, different electrical devices may echo weak EMI back to the powerline. In particular, switching mode power supply (SMPS) devices (*e.g.*, FLs, LEDs, chargers) exhibit continuous harmonic signatures in the current waveform. These signatures can be measured by current probes on powerlines, and used to disaggregate the energy usage of different types of appliances [62]. In contrast, LiTell can extract signatures from the same type of appliances, *i.e.*, FLs, by discriminating tiny differences in the optical frequency.

10. CONCLUSION

In this paper, we explore the feasibility of using unmodified FL fixtures to build a robust indoor localization system. We design and implement LiTell, a system that can discriminate subtle differences in the weak, high frequency characteristics of FLs' emission, utilizing COTS smartphones' cameras augmented with customized sampling/amplification algorithms. Our field tests show that LiTell holds the promise as a ready-to-use, easy-to-deploy indoor localization system that is robust against environment and user habits. While refinement is ongoing, LiTell leads to a new direction for robust, infrastructure-free indoor localization with visible light.

11. ACKNOWLEDGEMENT

We appreciate insightful comments and feedback from the shepherd and the anonymous reviewers. We would like to thank Walgreens for providing the test field. The work reported in this paper was supported in part by the NSF under Grant CNS-1506657, CNS-1518728, CNS-1343363 and CNS-1350039.

12. REFERENCES

[1] ABI Research, "Indoor Location in Retail: Where Is the Money?" 2013.

[2] 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.

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

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

[5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor Localization Without the Pain," in *Proc. of ACM MobiCom*, 2010.

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

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

[8] 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.

[9] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I Am the Antenna: Accurate Outdoor AP Location Using Smartphones," in *Proc. of ACM MobiCom*, 2011.

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

[13] J. Armstrong, Y. Sekercioglu, and A. Neild, "Visible Light Positioning: a Roadmap for International Standardization," *IEEE Communications Magazine*, vol. 51, no. 12, 2013.

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

[15] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "RollingLight: Enabling Line-of-Sight Light-to-Camera Communications," in *Proc. of ACM MobiSys*, 2015.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing, 3^{rd} Edition*. Prentice-Hall, 2009.

[17] R. I. Sasaki, "The impact of electronic ballast compact fluorescent lighting on power distribution systems," *ECE Technical Reports*, p. 197, 1994.

[18] Philips Electronics N.V., "Self Oscillating 25W CFL Lamp Circuit," http://www.nxp.com/documents/application_note/AN00048.pdf.

[19] R. Spence and R. S. Soin, *Tolerance design of electronic circuits*. World Scientific, 1988.

[20] R. Narasimhan, M. D. Audeh, and J. M. Kahn, "Effect of electronic-ballast fluorescent lighting on wireless infrared links," *IEE Proceedings - Optoelectronics*, vol. 143, no. 6, 1996.

[21] S. Ronnberg, "Power Line Communication and Customer Equipment," Ph.D. dissertation, Lulea University of Technology, 2011.

[22] Pico Technology, "Pico Oscilloscope Range," https://www.picotech.com/products/oscilloscope.

[23] C. K. Liang, L. W. Chang, and H. H. Chen, "Analysis and Compensation of Rolling Shutter Effect," *IEEE Transactions on Image Processing*, vol. 17, no. 8, 2008.

[24] OmniVision Technologies, Inc., "OV5640 datasheet," https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf.

[25] Semiconductor Components Industries, LLC., "1/6-Inch 720p High-Definition (HD) System-On-A-Chip (SOC) Digital Image Sensor," http://www.onsemi.com/pub_link/Collateral/MT9M114-D.PDF.

[26] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, 1986.

[27] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979.

[28] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, 1987.

[29] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[30] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, 2005.

[31] E. Y. Lam, "Combining gray world and retinex theory for automatic white balance in digital photography," in *Proc. of International Symposium on Consumer Electronics (ISCE)*, 2005.

[32] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical Chemistry*, vol. 36, no. 8, 1964.

[33] Bruce Fraser, "Raw Capture, Linear Gamma, and Exposure," 2004.

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

[35] iFixit, "Nexus 5 Teardown," https://www.ifixit.com/Teardown/Nexus+5+Teardown/19016.

[36] OpenCV Dev Team, "OpenCV4Android SDK," http://docs.opencv.org/2.4/doc/tutorials/introduction/android_binary_package/O4A_SDK.html.

[37] Sony Corporation, "IMX179 Product Brief," http://www.sony.net/Products/SC-HP/IS/sensor1/img/products/ProductBrief_IMX179_20140910.pdf.

[38] Robert Bosch GmbH, "Bosch DLE 40 Professional," http://www.bosch-pt.co.in/in/en/laser-measure-dle-40-131500-0601016370.html.

[39] C. Zhang, "Access Dallas 1-wire bus on your PC with simple and cheap hardware," https://github.com/dword1511/onewire-over-uart.

[40] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, 1992.

[41] FLIR Commercial Systems, "Extech 407026 Heavy Duty Light Meter with PC Interface," http://www.extech.com/instruments/product.asp?catid=10&prodid=64.

[42] Cinetics, "Axis360 Pro," http://cinetics.com/axis360-pro/.

[43] U.S. Department of Energy, "Cost-Effectiveness of Linear (T8) LED Lamps," May. 2014.

[44] Maxim Integrated Products, Inc., "Why Drive White LEDs with Constant Current?" Aug. 2004.

[45] H. van der Broeck, G. Sauerlander, and M. Wendt, "Power driver topologies and control schemes for leds," in *Annual IEEE Applied Power Electronics Conference and Exposition*, 2007.

[46] J. Zhang, X. Zhang, and G. Wu, "Dancing with Light: Predictive In-frame Rate Selection for Visible Light Networks," in *Proc. of IEEE INFOCOM*, 2015.

[47] 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.

[48] A. Goswami, L. E. Ortiz, and S. R. Das, "WiGEM: A Learning-based Approach for Indoor Localization," in *Proc. of ACM CoNEXT*, 2011.

[49] Z. Yang, C. Wu, and Y. Liu, "Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention," in *Proc. of ACM MobiCom*, 2012.

[50] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless Indoor Localization Without Site Survey," in *Proc. of IEEE INFOCOM*, 2012.

[51] S. Rajagopal, R. Roberts, and S.-K. Lim, "IEEE 802.15.7 Visible Light Communication: Modulation Schemes and Dimming Support," *IEEE Communications Magazine*, vol. 50, no. 3, 2012.

[52] J. Randall, O. Amft, J. Bohn, and M. Burri, "LuxTrace: Indoor Positioning Using Building Illumination," *Personal Ubiquitous Computing*, vol. 11, no. 6, 2007.

[53] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Fiatlux: Fingerprinting Rooms Using Light Intensity," in *Proc. of Pervasive*, 2007.

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

[55] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao, "Pharos: Enable Physical Analytics Through Visible Light Based Indoor Localization," in *ACM Workshop on Hot Topics in Networks (HotNets-XII)*, 2013.

[56] 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.

[57] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending Mobile Interaction Through Near-Field Visible Light Sensing," in *ACM MobiCom*, 2015.

[58] Q. Xu, R. Zheng, and S. Hranilovic, "IDyLL: Indoor Localization Using Inertial and Light Sensors on Smartphones," in *Proc. of ACM UbiComp*, 2015.

[59] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas, "Using a CMOS Camera Sensor for Visible Light Communication," in *Proc. of IEEE GlobeCom*, 2012.

[60] M. Gupta, A. Veeraraghavan, and S. Narasimhan, "Optimal Coded Sampling for Temporal Super-Resolution," in *Proc. of IEEE CVPR*, 2010.

[61] M. Gulati, S. S. Ram, and A. Singh, "An in Depth Study into Using EMI Signatures for Appliance Identification," in *Proc. of ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys)*, 2014.

[62] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel, "Disaggregated End-Use Energy Sensing for the Smart Grid," *IEEE Pervasive Computing*, vol. 10, no. 1, 2011.