Spatial Interference Detection for Mobile Visible Light Communication

Ali Ugur Guler[*], Tristan Braud[*] and Pan Hui[*†]
{auguler@ust.hk, braudt@ust.hk, panhui@cse.ust.hk}
[*]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[†]Department of Computer Science, University of Helsinki

***Abstract*—With the explosion of the Internet of Things (IoT), the number of devices to identify grew up exponentially in recent years and solutions such as visual markers or radio technologies are starting to show their limits. Visible light markers have been proposed to overcome these limitations. However, visible light communication (VLC) is very sensitive to interferences. As such, if two light markers are overlapping, the corresponding signals cannot be recovered. In smart homes, this situation is likely to occur as devices can be placed in very close proximity. In this paper, we design a protocol based on orthogonal codes to detect and isolate adjacent light markers, and individually identify several contiguous objects. We implement the algorithms within an Android application and evaluate their effect both analytically and experimentally. We demonstrate the robustness of our protocol in different conditions, both at the transmitter and the receiver side. Light markers are correctly recovered with low error rates (under 5%), at a pace of 5 frames per second, enough for object identification in most IoT scenarios.**

I. Introduction

Motivation: Even though the idea of unique object identification dates back to the first supermarkets in the 1930s, the recent emergence of pervasive computing considerably extended the usages. For instance, an Internet of Things (IoT) application needs to identify and locate various connected devices. Similarly, Augmented Reality (AR) applications commonly use markers to fix virtual objects on top of the physical world. With the ongoing convergence of both technologies, we expect to encounter an increasing number of scenarios which require simultaneous, real-time and identification and localization of multiple objects. In this paper, we consider a smart home scenario, where the user interacts with various IoT devices through an AR application in a ”point-to-interact” fashion. The setting of such scenario includes 10 to 15 devices per room, among which smart plugs, lamps and switches, security systems, thermostat *etc.*

The application may use radio communication technologies to identify connected devices. Although some studies extract an impressive amount of information from public WiFi networks [24] and achieve an accuracy of up to 0.5 m with iBeacons [12], this level of precision is not suitable for unique object identification. In the case of smart homes, multiple devices may be located in very close proximity. For instance, two smart power outlets can be installed next to each other. In this situation, interacting with each plug independently is cumbersome and error-prone without centimeter-level accuracy. Due to its short range, Near-Field Communication (NFC) presents a better precision but requires the device to be physically accessible, ruling out its usage for smart lights on the ceiling. Vision-based object identification can provide extremely precise location information with more flexibility than NFC, using either systems based on image processing techniques [4] or markers that convey a message. Markerless systems employ computation-heavy object recognition algorithms which require fine tuning to differentiate between similar looking devices such as smart plugs. On the other hand, marker recognition algorithms are lightweight enough to run in real-time on any mobile device, while transmitting specific hard-coded messages such as unique identifiers.

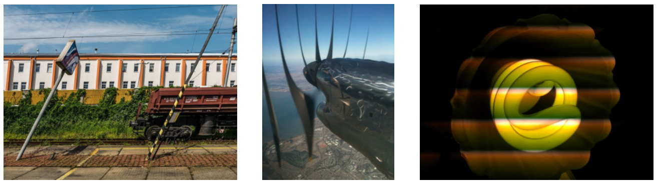

(a) Skew[1]. (b) Smear[2]. (c) Partial exposure[3].

Fig. 1: Several common effects caused by the rolling shutter effect on high speed moving objects.

Quick Response (QR) codes are the most common marker technology. These markers contain a significant amount of information and can be deployed with little effort. However, they also present critical drawbacks: QR codes are not aesthetically pleasing, provide little context to the user and require an unobstructed line of sight, good lighting conditions and a small enough viewing angle. Transmitting over longer distances requires markers of increasing sizes: the camera of a Nexus 5X can decode a 50x50 mm QR code up to 1.5 m only. At 20cm, the marker gets unreadable with 1% obstruction.

Visible Light Communication (VLC) can solve most of these issues. By modulating the intensity of a light source at a higher frequency than the human eye can detect, we can inconspicuously introduce VLC in the lighting or the

[1] Reinhold Möller/Wikimedia Commons/CC BY-SA 4.0
[2] User:Richmilliron/Wikimedia Commons/CC BY-SA 3.0
[3] Sage Ross/Wikimedia Commons/CC BY-SA 3.0

Light Emitting Diodes (LED) of devices. Comparatively to QR codes, VLC continues to operate with increased distance, viewing angle and partially obstructed line of sight while providing similar benefits, namely: (1) unique identification of objects, (2) lightweight algorithms and (3) accessible with only a basic smartphone camera (no photodiodes).

Most smartphone cameras rely on a rolling shutter to take pictures. Rolling shutters scan each line of the picture sequentially instead of capturing the scene as a whole. In other words, the image represents the different states of the object during the extent of the capture. Objects moving faster than this duration will therefore leave distinctive artifacts. Figure 1 shows some of the most common effects: skew(1.a), smear(1.b) and partial exposure(1.c). Partial exposure happens when a light source flickers faster than the rolling shutter speed. The resulting image reveals alternating black and white stripes of size proportional to the flickering frequency. As such, we can encode data in the flickering and recover it by measuring the size of the stripes. This effect, called the *rolling shutter effect* has already been leveraged in the literature to transmit markers and data with LEDs [28], [21], [22]. However, when multiple devices are in close proximity, as it would be the case for smart plugs and switches, the corresponding light markers will overlap, significantly impeding signal recovery.

In this paper, we introduce a new physical layer protocol to identify overlapping light sources and recover the signals. This solution enables easy and quick identification of separate objects in close proximity. However, it also introduces new technical challenges to address: (1) identify which light sources are interdependent, and differentiate them from single light sources, (2) isolate the regions of different transmission and separate signal and interference region, and (3) recover signals in close proximity by recognition of the interference.

Our contributions are as follows:

- A physical layer protocol for easier interference detection.
- Algorithms to isolate the interference regions from the transmission. The combination of the protocol and these algorithms is robust enough to recover the signal in various lighting conditions.
- An Android application exploiting our system design to recover identifiers transmitted over VLC.
- Analytical and real-world evaluation of the system. Our application performs on average with less than 5% error and processing time 100 to 200ms in realistic lighting conditions – a reasonable latency for interactive applications. In suboptimal conditions, we can recover the signal with low error rate by averaging over several frames.

The rest of this article is organized as follows: after reviewing research studies related to VLC and marker-based object identification (Section II), we describe our system design in Section III. We give a few implementation details and the optimization performed in our Android application in Section IV and finally analyze the performances and limitations of our model in Section V.

II. Related Works

Although VLC is most commonly used for network transmission as replacement for other RF technologies [25], other usages include presence detection [29], [30], vehicle to vehicle communication [15], indoors localization [13], and even finger tracking [19], which intersect with many marker applications [23], [5], [7].

At the intersection of these two technologies, VLC marker systems spread among three categories: color codes, spatial codes and temporal codes. As color codes exploit the difference between several wavelengths, they are vulnerable to lighting variations. While they can be used for tracking algorithms, they only provide low data rates [16]. For this reason, they are generally combined with temporal or spatial codes to provide additional information [3], [9]. One of the most common spatial code is the QR code. QR codes have the benefit of high data density and easy application [10]. However, they are prone to errors inherent to lighting and orientation changes. Another study proposes to exploit only light reflection to identify moving vehicles [27]. Temporal codes are more robust to lighting and orientation variation. By modulating LEDs with high-frequency, several teams managed to send data at rates reaching over 1 Gb/s [1], [26]. However, only silicon photodiodes can achieve such high data rates. On the other hand, the camera on a smartphone can only provide a connection with low data rates [12]. Even though communication with a CMOS camera is much slower, there are several reasons to prefer using it over photodiodes for object identification:

- The user has to point the photodiodes directly at the light source and cannot exploit light reflection.
- CMOS cameras are already built in smartphones, while fast response photodiodes are not readily available.
- The amount of information in a marker is usually low.

Due to the Nyquist theorem [20], temporal communication through cameras can only achieve a data rate of 15 Bit/s if the camera can record 30 frames per second. Nevertheless, Danakis et al. have shown that by exploiting the rolling shutter effect on CMOS cameras, it is possible to increase the data rate to 1 Kb/s [2].

The rolling shutter effect has opened many possibilities for VLC applications on smartphones. Panasonic already has a commercial VLC marker project “Light ID” [21]. Luxapose uses the rolling shutter effect to identify different LEDs and use them for accurate indoor positioning [17]. Rajagopal et al. have used modulated ambient light in the 8Khz frequency range (making the flickering invisible to the human eye) and successfully separated 29 channels in the time domain with 0.2khz channel separation [22]. DisCo group has constructed a system that introduces robustness against occlusion, movement, and distance [11]. Yet, these solutions either focus on a single marker or only consider non-interfering light sources.

Several studies have been performed on combining the spatial domain and the time domain to increase visible light communication transmission rates. VRCodes uses binary cod-

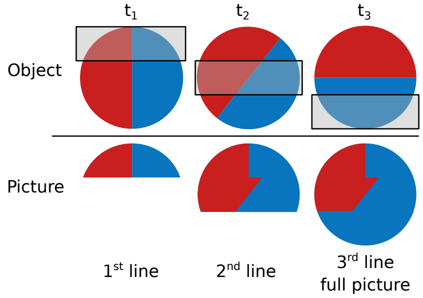


Fig. 2: A fast rotating object (up) as captured by a CMOS Camera (down). As the picture is taken line by line sequentially (gray rectangle at t_1, t_2 and t_3), each line represents the object in a different position. The resulting picture (t_3) is therefore deformed.

ing with color, time and space domain to increase data rates by modulating LCD screens or projectors [28]. COBRA uses a color barcode stream system relying on the color, time, space domains [8]. T. Langlotz and O. Bimber also proposed 4D barcodes projected by screens [18]. However, these works focus on transmission through LCD screens, which are less convenient than integrating the transmission in lighting equipment.

In this paper, we use orthogonal preambles to separate channels in the spatial domain. Our work focuses on identifying interference regions of multiple LEDs used for illumination and introduce spatial parallelism to VLC. We seamlessly integrate VLC markers in the lighting of objects while being able to distinguish between several overlapping sources and develop a smartphone application to recover the signals.

III. System Design

In a smart home environment, smart devices can be placed close by to each other. This may cause the corresponding light sources that transmit identifiers to overlap and interfere with each other. Our system is designed to recover the unique identifiers even in a case where the light marker is subject to interference. In this section, we first describe the rolling shutter effect, then explain the transmitter and receiver functionalities. Finally we discuss how rotation might affect the system.

A. The Rolling Shutter Effect

Most smartphone cameras take pictures using a rolling shutter. The rolling shutter scans the image in one direction, line by line, instead of taking a snapshot of the whole scene. As the camera does not record all parts of the scene at the same instant, fast moving objects and rapid variations of light leave a distinctive, predictable pattern on the image. Figure 2 displays how this phenomenon happens. The top line presents a circular blue and red round object rotating clockwise over time. We represent this object at three successive moments t_1, t_2 and t_3, respectively corresponding to the time the rolling shutter captures the first, second and third line. These lines are depicted as a gray rectangle on the object. As we can see, the object moves fast enough to be in a different position when the rolling shutter captures a new line at t_1, t_2 and t_3. On the bottom line, we represent the picture as it is captured over time. At $t = t_1$, the camera captures the first line, then, at $t = t_2$, the rolling shutter scans the second line and adds it to the first one. Finally, at $t = t_3$, the third line is captured and added to the two first lines, resulting in the final picture. As the camera scans each line at a different time, it captures the object in a different position from the other lines, leading to the deformation in the full picture capture at t_3.

In the case of a LED flashing at a high-frequency on the transmitter side, the light source will leave specific black stripes on the picture. Each stripe corresponds to a time without lighting, and its size is directly proportional to the off duration. Exploiting this effect, we can encode more data bits in a single frame of the camera, not only increasing the effective bandwidth of the system, but in our scenario, permitting to transmit a full identifier over a single frame. Another advantage of this technique resides in the fact that the user does not have to directly point the camera at the light source, but can also exploit the reflected light on a surface. As a result it is possible to embed the identifier into the lightning of the object, making the energy consumption of our solution transparent for the transmitter.

B. Transmitter Side

We introduce a new physical layer protocol at the transmitter side to facilitate the isolation of overlapping signals at the receiver side. This protocol is based on orthogonal preambles, *i.e.* when light sources of same intensity interfere with each other, preambles cancel each other's high-frequency content (Figure 4). The receiver can then easily detect the interference regions by analyzing the high-frequency content (see Section III-C). This protocol is represented by Figure 3.

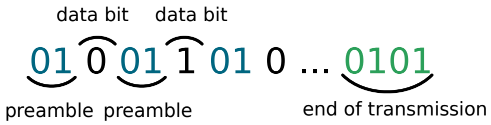


Fig. 3: The transmission protocol: 2 bits preamble before each actual data bit and 4 bits for end of transmission.

A frame contains a short length of data, up to 8 bits per LED. This protocol thus permits to identify 255 independent objects, even in suboptimal conditions (signal partially obstructed, high cutoff angle or insufficient lighting). The transmitter sends two preamble bits before every symbol, and four bits to signal the end of the transmission, making the full transmission 28 bits long (see Figure 3). As the transmission is at most one byte long, using a short preambles yields similar results to a single long preamble, while protecting each symbol. Moreover, this system avoids misinterpreting the preambles and the end of transmission code as part of the identifier and makes the interference region easier to isolate. Note that if we opt here for a conservative approach for the sake of clarity, we can easily extend this protocol to transmit up to 16 bits per frame, resulting in a much more realistic

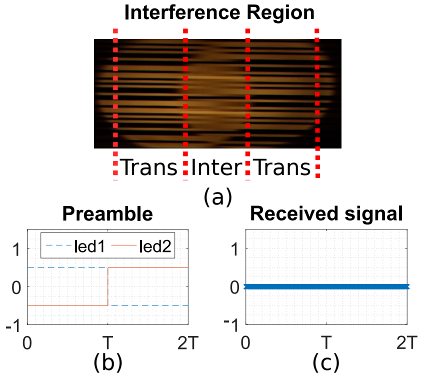


Fig. 4: The orthogonal preambles. (a) transmission and interference regions. Black stripes (high-frequency) are less visible in the interference region. (b) orthogonal preambles of LED1 and LED2. (c) high-frequency content in the interference region. The orthogonal preambles cancel each other.

number of identifiers (65535). However, this would require brighter LEDs and higher resolution cameras (already present in nowadays high-end smartphones).

We transmit data using On/Off keying as it is the easiest and most robust way to transmit information through LEDs. Indeed, the luminous flux does not follow linearly the forward current and varies with the temperature. The light/no light paradigm is therefore easier to detect and less sensitive to luminosity variation. Regarding the modulation, Return-to-Zero (RZ) codes such as Manchester coding are usually favored for their ability to integrate the clock data into the signal. However, Non-Return-to-Zero (NRZ) codes use half of the bandwidth. Therefore, NRZ coding permits us to transmit data with similar efficiency as if we used Manchester Coding with a single long preamble.

C. Receiver Side Process

The signal reception process is composed of three phases:

- Detecting the different light areas.
- Finding the interference regions for each area.
- Demodulation of the signal and recover the message.

Light Source Detection: Our process for light source detection is similar to Luxapose[17]. Yet, we introduce significant differences: (1) Luxapose cannot distinguish between several light sources, whereas our system considers both disjoint and joint sources, (2) if light sources are joint, Luxapose recovers at most one of the signals; moreover, if the the interference region is in the middle of the light source, Luxapose will decode the interference as a signal, giving an erroneous result.

We start the light detection procedure with a grayscale image. Since we use a very short exposure time, the ambient light is filtered. We then apply a Gaussian filter to eliminate the dark strips caused by modulation. After filtering the image, we use the OTSU thresholding algorithm to extract light sources. Finally we find the contours of each white region to identify the locations of light sources.

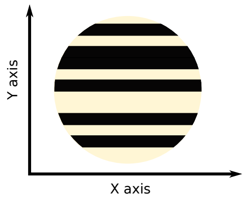


Fig. 5: X-axis (perpendicular) and Y-axis (parallel).

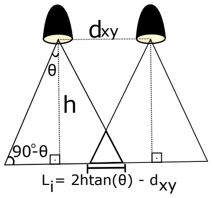


Fig. 6: Intersection region depending on cut off angle.

Finding Interference Regions: The smartphone's camera captures the reflected light from a surface, using a rolling shutter mechanism which shoots the scene line by line sequentially. In this section, and until the end of the paper we will call X-axis the axis perpendicular to the rolling shutter direction, and Y-axis the axis parallel to the shutter direction as represented in Figure 5. Our X-axis is therefore always the axis parallel to the black stripes generated by the LEDs. As such, it contains only spatial information as all pixels are captured at the same time along a line. The Y-axis is parallel to the rolling shutter direction and therefore contains both spatial and temporal information. Indeed, when we move along the Y-axis, both the sampling time and location of the pixels change. For this reason, we first take a window along the Y-axis. Then, we compare the energy levels of the high-frequency content in this window over the X-axis and analyze the local minimums and maximums to detect the different regions.

We start the signal recovery with taking a window along the Y-axis, at the center of the light source. The window size is equal or greater than $3T_d$ where T_d is the number of pixels required to recover a single bit. Since our protocol ensures there is at least one preamble every three bits, this window size guarantees that we can detect the interference regions. To measure the high-frequency energy, we first subtract the mean of the window. After isolating the high-frequency content, we compute the energy level in the window as follows: $E = \sum_{n=0}^{3T_d} |X[n]|^2$, where $X[n]$ is the value of the n^{th} pixel. By repeating this procedure over the X-axis, we plot the energy levels over the illuminated area as shown in Figure 7. In Figure 6, we represent the system projection at the center of the light source on the Y-axis. If the light sources are perpendicular to the surface, we can estimate the interference region size to be equal to $L = 2htan(\theta) - d_{xy}$, as shown Figure 6. We can clearly see that in the transmission region the high-frequency content is visible and the energy levels are also high. However in the interference region, the high-frequency content is eliminated and the energy levels drop. We therefore consider the local minimum as the center of the interference region and the local maximum as the center of the transmission region. After isolating each region, we analyze them as sub-images and process each of them separately.

Demodulation: When the transmission regions are identified, we demodulate the signal along the Y-axis as displayed in Figure 8. First, we convert the 2D image into a 1D signal. We then eliminate the DC content of the signal by using a

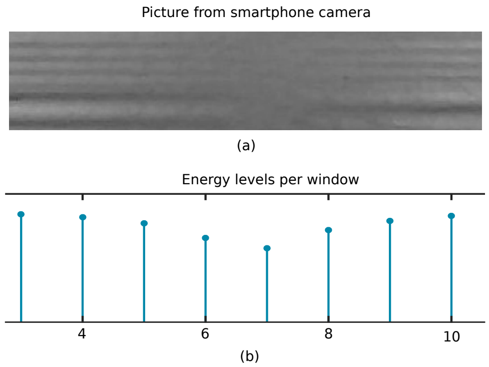


Fig. 7: a) Interference region captured with a smartphone camera b) Energy levels of window slices through X-axis. The energy levels drop around the interference zone

moving average filter. Once the DC content is filtered out, we apply a threshold function to convert it in the $\{0,1\}$ space. As the channel noise is reasonable for low ISO, this threshold function does not cause large jumps between sample points. Finally, the frequent preamble signals satisfy the required high to low, low to high transitions to retrieve the signal clock through a digital early-late clock recovery algorithm [31].

Thanks to this design, the demodulation is almost completely independent from the receiver's characteristics. The rolling shutter effect will indeed happen for any fast blinking signal on any CMOS camera, and due to the NRZ coding, the receiver can self-tune itself for reception. As our signal has identifiable patterns (preambles), even the sampling rate can automatically be tuned with phase loop algorithms.

D. Rotation

As the camera is fixed on the smartphone, the rolling shutter orientation remains the same, whichever the alignment of the smartphone (partial or complete rotation). In our computations, we always consider the interference region to be perpendicular to the rolling shutter direction. If we rotate the phone, as the Y-axis contains both spatial and temporal information, the orthogonal preambles still cancel each other. Interference regions therefore still present a lower energy than transmission regions and can be detected using the same process.

IV. Implementation

We implement the receiver-side elements in an Android application using the Camera2 API. This API permits us to precisely control the parameters of the camera, specifically the exposure time and the sensitivity. Most smartphones supporting this API on Android 6.0 should run the application. We tested the application on two devices: LG Nexus 5X and Samsung Galaxy S8. Both are common, middle to high-end phones with a standard 12Mp rear camera.

To avoid variation in our measurements with external conditions changes (especially lighting), we disable the automatic handling of exposure, white balance, and focus. We set the exposure time manually to a low value ($\frac{1}{12000}\,s$) in order to capture the rolling shutter effect. Similarly, the sensitivity is set to ISO 200 to filter the ambient light and only keep external light sources with minimal noise. Our application takes pictures at a resolution of 4096x3024. For optimization reasons, we first downsample the picture by considering one sample every 5 pixels. This leads to a resolution of 820x604 for the rest of the processing. The light source identification is integrally handled by openCV 3.2. Thanks to openCV, we recover the geometrical center of the light source. We then isolate a slice on the X-axis of height 5 periods and width the contour width, centered around this point and compute the energy levels over 50 sample points. We can further improve performance by using only 20 sample points, at the cost of less precise interference region detection. We isolate the local minimums and maximums energy levels corresponding to the signal and interference region centers. Finally, for each signal region, we select a slice on the Y-axis, centered around at the local maximum, width 20 pixels and length the diameter of the contour. We then decode the signal from this slice. These steps are represented in Figure 9. All sub-figures come from the application and are cropped to the same dimensions.

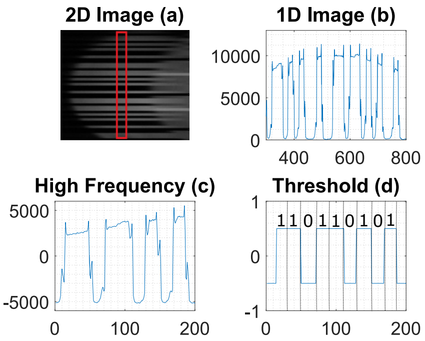
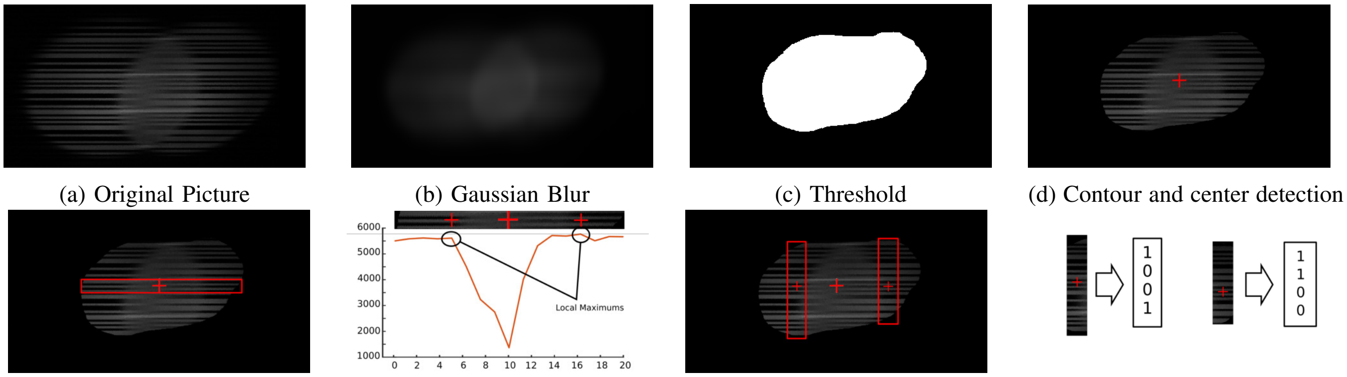


Fig. 8: Signal extraction. (a) The 2D image and the corresponding 1D signal (red). (b) The signal extracted from the 1D image, modulation is visible and DC content is still present. (c) DC content is filtered, only high-frequency remains. (d) Thresholding of the signal.

With these parameters, and despite the use of computer vision algorithms, our application manages to process pictures between 5 and 10 frames per second. This rate is enough for the scenario of this study: identifying objects or devices for further interaction with the user. However, should the need arise, we can greatly improve performance by implementing the image processing operations using Android's Native Development Kit (NDK), or even by offloading the heaviest computations to a cloud server.

V. Evaluation

In this section, we evaluate the parameters affecting the performance of our method. We first perform an analytical evaluation where we model two light sources overlapping in various conditions. We then evaluate our application in real life conditions to verify the performance of our system.



(a) Original Picture (b) Gaussian Blur (c) Threshold (d) Contour and center detection

(e) Window on X-axis (f) Energy levels analysis (g) Window on Y-axis (h) Identifier recovery

Fig. 9: The full image analysis process we implemented it in the Android application. The top row represents the different steps to prepare the image for processing. The second row displays the interference region identification (e-f) and the identifier recovery (g-h).

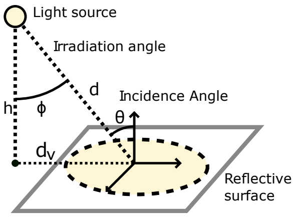


Fig. 10: Lambertian radiance pattern.

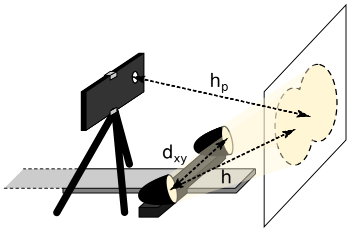


Fig. 11: Experimental setup.

A. Analytical evaluation

We use the Lambertian radiance pattern for LED sources defined by Kahn et al. [14] (see Figure 10):

$$H_i(0) = A\frac{m+1}{2\pi}\frac{cos(\phi)^m cos(\theta)}{d^2} \tag{1}$$

Where A is the surface area of the receiver, m the Lambertian coefficient, d the distance of the light source to the surface, ϕ the irradiation angle and θ the incidence angle. A is constant, and m is equal to 1 when the cutoff angle of a LED is equal to 60 degrees. For simplification we take $\phi = \theta$. Then:

$$H_i(0) = \frac{Ch^2}{(h^2+d_v^2)^4} \tag{2}$$

Where $C = \frac{A(m+1)}{2\pi}$ is constant and equal amongst LEDs when identical LEDs are used. Moreover d_v is a function of h and d_{xy} (Figure 6). As a result the characteristics of the interference region depend on the distance between LEDs (d_{xy}) and the distance from the surface to the LED (h).

We simulate the setup presented Figure 11 for two identical LEDs located at an equal distance from the surface. In Figure 12 we show the variation of the high-frequency content over the simulated region. The energy levels are computed and normalized using the method presented in Section III. We can see that the high-frequency content has the highest energy in the area closest to LEDs (in this case pixels 150 and 225). We also observe the effects of orthogonal preambles canceling each other: the high-frequency content is lower at the center of the overlapping region.

We define two regions in Figure 12: the transmission region and the interference region with respective lengths L_t and L_i. The transmission region is the area in which we can safely decode the signal. On the contrary, in the interference region, the algorithm cannot decode the signal due to the interference. To separate these regions we choose a cutoff energy level $E_c = \frac{E_{max}+E_{min}}{2}$ and characterize the transmission as the region where $E > E_c$ and the interference as the region where $E < E_c$. Note that the interference region and the physical intersection region are different. When one LED has a higher signal strength than the other, the algorithm can recover the signal even in the overlapping area. As such, the interference region is usually smaller than the physical intersection.

In Figures 13 and 14 we compare the energy ratios $\frac{E_{max}}{E_{min}}$ and the size of the interference region as a function of the size of the transmission regions for different d_{xy} and h. Both d_{xy} and h curves display the same behavior for the same h/d_{xy} ratio. The interference region can indeed be characterized by the ratio between h and d_{xy}. We divide Figures 13 and 14 in three areas: (1) light sources are contiguous, with high $\frac{h}{d_{xy}}$ and act as a point source. The algorithm cannot distinguish the transmission from the interference. (2) light sources are at a medium distance, a clear interference region appears due to the cutoff angle. (3) light sources are at far away from each other with wider cutoff angle and low $\frac{h}{d_{xy}}$ ratio. The interference region is large with low energy.

In the first case, the light intensities of the sources are approximately equal on the illuminated surface. This causes the majority of the illuminated area to appear as an interference region. Recovering the transmitted signal in these condition becomes challenging. In the second case, the interference region borders are clearly visible due to the sudden drop of energy. It is easy to detect both the interference and the transmission regions. In the third case, we observe a

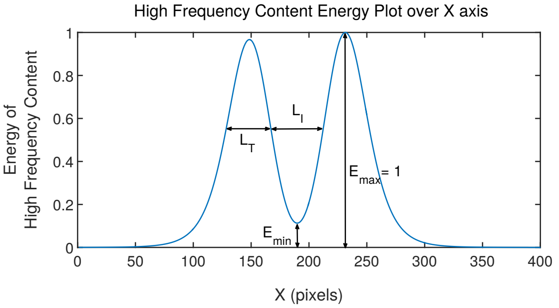


Fig. 12: Simulated energy plot L_t corresponds to length of transmission region in pixels, L_i corresponds to length of interference region in pixels.

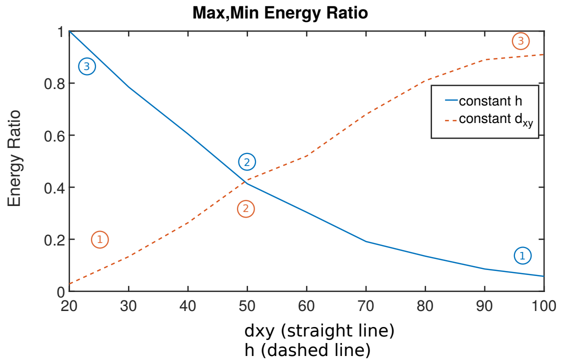


Fig. 13: Energy ratio of maximum and minimum points. The higher the ratio, the easier it is to detect regions.

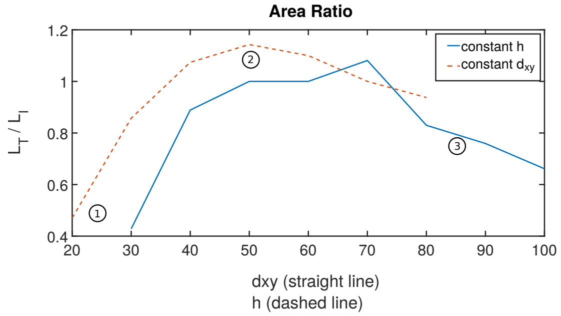


Fig. 14: Transmission area to interference area ratio. As the d_{xy} LEDs raises, the transmission becomes bigger comared to the interference, before the cutoff angle gets too high. For $h > 50$, the illuminance gets too low for detection.

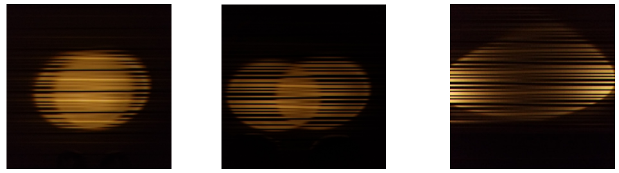
smooth transition of energy levels between the interference and transmission regions. However the area of the interference region becomes larger than the transmission region as changes on x,y locations are not affecting the total distance as much as changes in height. Cases two and three show that the light intensity of the interference region is strongly dependent on the position of the LEDs, and can even get lower than the transmission region. In such a scenario, a method relying on the energy of the frequency contents would be more robust than detecting interference by light intensity.

Figure 13 shows that h and d_{xy} have inversely proportional effects. If the LEDs get too close to each other, the interference region gets difficult to differentiate from the transmission regions. On the other hand, putting the LEDs at a lower height enables optimal conditions for interference detection.

B. Experimental evaluation

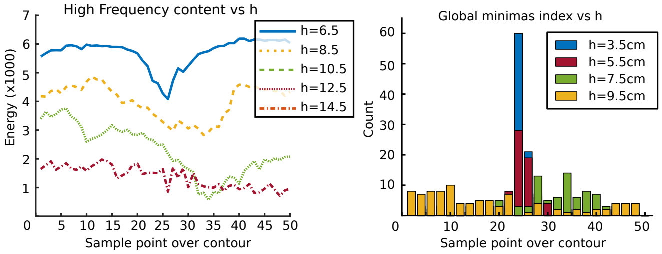
To confirm the results presented in the previous section, we setup a simple test bench to characterize the limits of our application. To this purpose, we connect consumer market LEDs to an Intel Galileo which handles the modulation and synchronization. The LEDs have 0.3 W power, $\Phi_v = 10$ lm luminous flux and work with 5 V from a USB outlet. The Intel Galileo modulates the LEDs at 8 KHz. We mount the LEDs on rails to vary h and d_{xy} (see Figure 11). Due to the low power of the LEDs, we keep both parameters at low values, under 200 mm: over $h > 200\,mm$, the illumination gets too low for the phone to detect the light source. However, the area of the light source is directly proportional to the luminous flux. For instance, with our setup, for $h = 50mm$ the source's diameter is 40 mm and the illuminance $E_v = \frac{\Phi_v}{A} = 7,961\,lux$. For the same illuminance, the area diameter reaches 120 and 340 mm respectively, and h goes from 150 to 425 mm. We perform the measurements on the walls of a typical office, with measured illuminance ranging between 30 and 60 lux. In practice, the typical indoor illuminance indoors is in the 50-500 lux range. Comparatively, typical illumination appliances will use 1 to 10 W LEDs, with respectively 100 lux and 800 lux light flux which makes them easier to distinguish than our experimental conditions. Furthermore, with the short exposure required for exploiting the shutter effect, our solution filters out the ambient light, and only light sources directly pointed at the surface will appear on the picture. We perform the evaluation on a blank white wall as this scenario is most likely to happen in a smart home: devices will either be hanged on a clear surface or placed next to a clear surface. All the measurements were done using our Android application running on a Samsung Galaxy S8, a fairly recent smartphone presenting a full compatibility with the Camera2 API. As we stated in Section III-B, other encoding schemes either double the necessary bandwidth or require some manual tuning for the receiver to properly recover the signal. As such, we discarded these schemes at the design stage and will not consider them in this section.

1) Interference Point Detection: Figure 15 displays the effect of high, medium and low $\frac{h}{d_x}$. In Figure 15.a, both LEDs are too close to each other compared to the distance to the surface (Zone 1 of Figure 14). On Figure 15.b, both interference and transmission regions are clearly distinguishable and the signal can be decoded. Finally, Figure 15.c displays the effect of a low $\frac{h}{d_{xy}}$. Due to the low power LEDs and the short exposure time, this effect is harder to display on a picture. This image shows what happens when LEDs are too far away from each other, for instance when illuminating an object from all sides in a display case. However, for IoT applications, we expect LEDs to be close to each other. Figure 15.b reflects



(a) High $\frac{h}{d_{xy}}$. (b) Medium $\frac{h}{d_{xy}}$. (c) Low $\frac{h}{d_{xy}}$.

Fig. 15: Interference regions on images. (a) LEDs act as a single point source, interference region covers the illuminated area. (b) Interference and transmission regions are clearly defined. (c) Interference region is at a very low energy area, hard to separate regions.



(a) Energy levels (b) Center of inter. region

Fig. 16: Energy levels and detected center of the Interference region for variable h ($h_p = 20\,cm$, $d_{xy} = 5\,cm$, ISO200). For $h > 7.5\,cm$, the noise becomes too important for distinguishing the local minimums and maximums (a). For $h > 6.5\,cm$, the detected center distribution spreads out of the actual interference region (b).

this situation. Figure 16.a represents the energy levels of the high-frequency content over the X-axis and confirms the results from the theoretical analysis As h increases, we can observe that energy difference between the local minimums and maximums is decreasing. Meanwhile, the area of the interference region increases with h. Figure 16.b shows the distribution of the detected interference points for different h values. The deviation of local minimum indexes increases with h, resulting in interrupted or erroneous transmissions.

2) *Signal Recovery:* We classify signal recoveries in three categories: no signal, erroneous and correct. No signal happens when the signal cannot be decoded. A large interference region or a small surface area causes the majority of these invalid recoveries, discarded by the application. Erroneous recovery corresponds to the application decoding a false signal. We consider the transmission as successful when the proportion of correct signal recoveries is significantly higher (at least 5 times) than the erroneous signals. Indeed, we are considering a master system, in which we can choose the most frequent signal over several attempts as the identifier. Since we recover the IDs in a single frame, we evaluate our system on a frame per frame basis. We measure the accuracy of our application for 200 processed frames over the following parameters: distance between LEDs d_{xy} (5 and 10 cm), ISO sensitivity of the camera (ISO 200 and 800), distance between the phone and the surface h_p (20 and 35 cm), and distance between LEDs and the surface h (3.5 to 18.5 cm). We show the results in Figure 18; in optimal conditions, our system reaches more than 70% correct signal recovery, with less than 5% erroneous signals. We identify four factors affecting signal recovery:

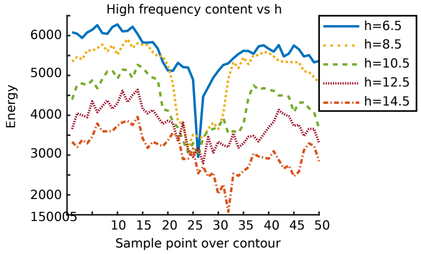
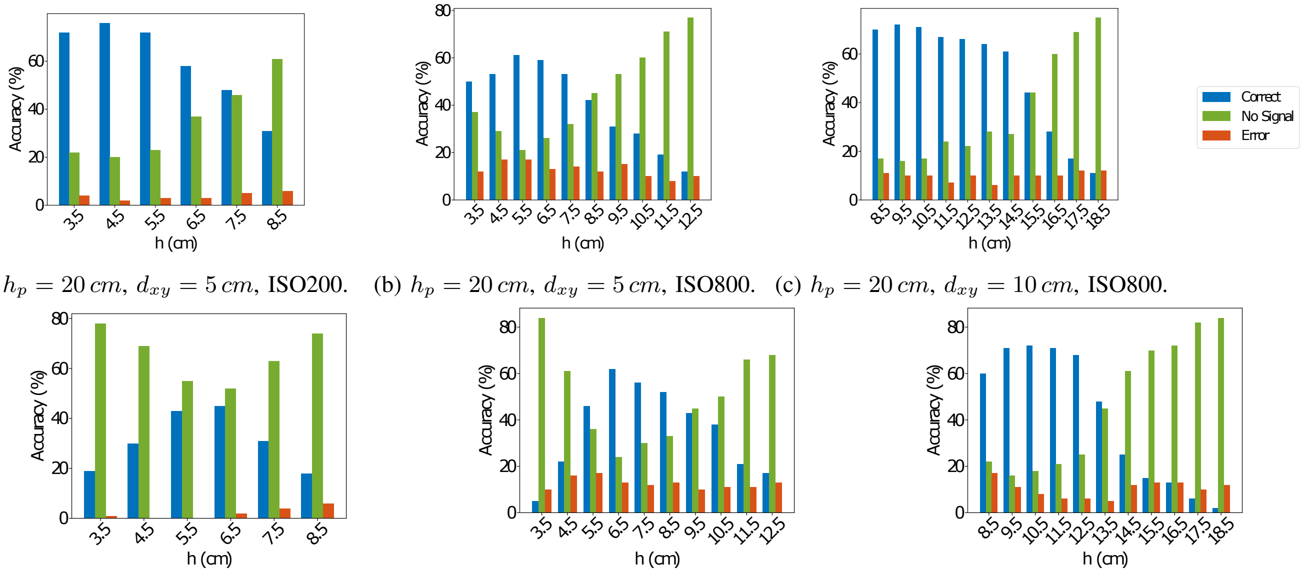


Fig. 17: Energy levels for a sensitivity of ISO 800.

Illuminance intensity: A camera requires a certain amount of illuminance to detect light sources, that depends on parameters such as exposure time and ISO sensitivity. If the illuminance is below that threshold, the contour detection algorithm does not work and the signal recovery fails. Figures 16.a and 17 show the change of energy levels over varying h respectively for ISO200 and ISO800. For a sensitivity of ISO200, the illuminance is too low to detect the interference region when $h > 7.5\,cm$. With a higher ISO sensitivity, h can reach up to 11.5 cm, a 23% increase in range. Figures 15.d and 15.e confirm this outcome. We consider the case where $h = 8.5\,cm$. In Figure 15.d correct signal recoveries are substantially lower than no signal recoveries, while the system recovers more than 50% of the signals correctly with a higher sensitivity (Figure 15.e). However, using a higher ISO sensitivity leads to higher noise in the system: erroneous recoveries increase from 1-2% to over 10%.

Contour area and h_p: In our experiments, one period of an 8 Khz signal corresponds to a vector length of 20 pixels on the Y-axis. The protocol we are using requires 13 periods to transmit a 4-bit long identifier. Therefore, in order to guarantee a full transmission, the minimum area of a detected contour has to be at least $2 \times 13 \times 20 = 520$ pixels long. Otherwise, the system cannot guarantee the complete recovery of the signal. The pixel representation of contour area is proportional to h and inversely proportional to h_p. To highlight this phenomenon, we vary the phone to surface distance h_p between 20 cm and 35 cm. In Figure 18.d ($h_p = 35\,cm$) for lower h values, the largest slice on the Y-axis is barely enough to contain a full transmission. Therefore no signal rates peak at $h = 3.5cm$, reaching almost 80%. Comparatively, no signal rates are at their minimum for $h_p = 20\,cm$ and $h = 3.5\,cm$ (Figure 18.a) and grow with h.

Interference region to transmission region ratio: In Figures 18.c and 18.h, we display the impact of a higher d_{xy} on the transmission. The LEDs we use have a cutoff angle of 60 degrees. Therefore two light sources interfere with each other at $h = 3.5\,cm$ and $h = 8.5\,cm$ for $d_{xy} = 5cm$ and $d_{xy} = 10\,cm$ respectively. We perform the measurements with



(a) $h_p = 20\,cm$, $d_{xy} = 5\,cm$, ISO200. (b) $h_p = 20\,cm$, $d_{xy} = 5\,cm$, ISO800. (c) $h_p = 20\,cm$, $d_{xy} = 10\,cm$, ISO800.

(d) $h_p = 35\,cm$, $d_{xy} = 5\,cm$, ISO200. (e) $h_p = 35\,cm$, $d_{xy} = 5\,cm$, ISO800. (f) $h_p = 35\,cm$, $d_{xy} = 10\,cm$, ISO800.

Fig. 18: Accuracy percentages depending on h_p (vertical), the sensor sensivity and the distance between the LEDs (horizontal). We start the measurements at higher h for $d_{xy} = 10\,cm$ as the light sources start to overlap when $h = 8.5\,cm$.

an ISO sensitivity of 800 to capture enough light and decode the signals for $h > 8.5$ cm. In Section V-A we mentioned an optimal ratio $\frac{h}{d_{xy}} = 1$ for interference region detection and signal recovery. In Figure 16.a and 17, local minimums are harder to identify as the h to d_{xy} ration increases. In Figures 15.b and 15.c the highest signal recovery rates happen for $\frac{h}{d_{xy}} = 1.1$ and $\frac{h}{d_{xy}} = 0.95$ respectively. For a constant d_{xy}, the correct signal ratio decreases as h increases as shown in the simulation. We did not test the case where $d_{xy} >> h$, as the cutoff angle of 60 degrees prevents such occasions.

Amount of overlap between light sources: Using the overlap equation presented Figure 6, we compute the amount of overlap for which the ratio of correct to erroneous signal recoveries is smaller than 0.8. In Figures 18.b and Figure 18.c, the signal recovery can happen for an overlap of up to 40%. In Figure 18.a, the point for which the ratio of correct to erroneous recoveries is smaller than 0.8 is not reached as the no signal recovery quickly takes over. However, this Figure shows that with a lower ISO, our algorithms can recover the signal for major overlaps (over 50%).

Another interesting finding from Figures 18.b,c,e, and f is the percentage of erroneous recoveries. This percentage is evenly distributed over the different parameters changes, around 10%, suggesting that the erroneous recoveries mostly come from the noise caused by the increasing ISO sensitivity.

VI. Conclusion

In this paper, we proposed a new method to decode visible light signals interfering with each other using the rolling shutter effect of smartphone cameras. We designed a complete system for finding the spatial interference regions and recovering the signal from the transmission areas. Through this study, we used this technique to replace visual markers in ubiquitous networks; specifically, we focused on IoT scenarios, where multiple devices in close proximity have to be identified and signals are likely to interfere with each other. By exploiting both the spatial and the temporal domain, we were able to distinguish between several overlapping yet independent transmissions and decode the corresponding signals.

We demonstrated the robustness of our own transmitter side protocol and the receiver recovery system to variations in the transmission parameters. When the LEDs are positioned at a reasonable distance from the surface relatively to their power (which should be the case for any light used for object illumination) our system could recover the identifiers with less than 5% of erroneous signal recoveries and ran on a smartphone at a minimum rate of 5 frames per seconds, a reasonable latency in most IoT situations.

In future works, we plan to design another protocol exploiting spatial parallelism for longer transmissions, and integrate it in high-power light bulbs in a similar way as LiFi [6]. To achieve this goal, we should focus on recovering the signal from the overlapping region itself, which would especially be useful when the interference gets bigger than the transmission region. Another track would be increasing the amount of data in the marker by linking multiple LEDs together: n LEDs each transmitting k bits can convey up to kn bits of information (2^{kn} IDs) per frame if one differentiates between linked and independent overlapping LEDs.

VII. Acknowledgements

The authors thank our shepherd and the anonymous reviewers for their insightful comments. This research has been supported, in part, by projects 26211515 and 16214817 from the Research Grants Council of Hong Kong.

References

[1] A. H. Azhar, T. Tran, and D. O'Brien. A gigabit/s indoor wireless transmission using mimo-ofdm visible-light communications. *IEEE Photonics Technology Letters*, 25(2):171–174, 2013.

[2] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas. Using a cmos camera sensor for visible light communication. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1244–1248. IEEE, 2012.

[3] T. Fath, F. Schubert, and H. Haas. Wireless data transmission using visual codes. *Photon. Res.*, 2(5):150–160, Oct 2014.

[4] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab. Markerless tracking for ar: A learning-based approach. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 295–, Washington, DC, USA, 2002. IEEE Computer Society.

[5] L. R. P. Gomez, N. Fairfield, A. Szybalski, P. Nemec, and C. Urmson. Transitioning a mixed-mode vehicle to autonomous mode, Dec. 13 2011. US Patent 8,078,349.

[6] H. Haasn. Wireless data from every light bulb, 07 2011.

[7] O. A. Hammadi, A. A. Hebsi, M. J. Zemerly, and J. W. P. Ng. Indoor localization and guidance using portable smartphones. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 337–341, Dec 2012.

[8] T. Hao, R. Zhou, and G. Xing. Cobra: color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 85–98. ACM, 2012.

[9] P. Hu, P. H. Pathak, X. Feng, H. Fu, and P. Mohapatra. Colorbars: Increasing data rate of led-to-camera communication using color shift keying. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '15, pages 12:1–12:13, New York, NY, USA, 2015. ACM.

[10] Information technology – automatic identification and data capture techniques – qr code bar code symbology. Standard, International Organization for Standardization, Feb. 2015.

[11] K. Jo, M. Gupta, S. K. Nayar, D. Liu, J. Gu, Y. Hitomi, T. Mitsunaga, Q. Yin, D. Iso, B. Smith, et al. Disco: Displays that communicate. *Technical Report No. CUCS-028–14*, pages 1–20, 2014.

[12] A. Jovicic, J. Li, and T. Richardson. Visible light communication: opportunities, challenges and the path to market. *IEEE Communications Magazine*, 51(12):26–32, 2013.

[13] S. Y. Jung, S. Hann, and C. S. Park. Tdoa-based optical wireless indoor localization using led ceiling lamps. *IEEE Transactions on Consumer Electronics*, 57(4):1592–1597, November 2011.

[14] J. M. Kahn and J. R. Barry. Wireless infrared communications. *Proceedings of the IEEE*, 85(2):265–298, 1997.

[15] D. R. Kim, S. H. Yang, H. S. Kim, Y. H. Son, and S. K. Han. Outdoor visible light communication for inter- vehicle communication using controller area network. In *2012 Fourth International Conference on Communications and Electronics (ICCE)*, pages 31–34, Aug 2012.

[16] G. Koutaki, S. Hirata, H. Sato, and K. Uchimura. [poster] marker identification using ir leds and rgb color descriptors. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 96–99. IEEE, 2015.

[17] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458. ACM, 2014.

[18] T. Langlotz and O. Bimber. Unsynchronized 4d barcodes. In *International Symposium on Visual Computing*, pages 363–374. Springer, 2007.

[19] S. Ma, Q. Liu, C. Kim, and P. Sheu. Lift: Using projected coded light for finger tracking and device augmentation. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 153–159, March 2017.

[20] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[21] Panasonic. Light id, 2016. Accessed 17-06-2017.

[22] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 249–260. IEEE Press, 2014.

[23] C. Roeding and A. T. Emigh. Method and system for presence detection, Apr. 19 2010. US Patent App. 12/762,556.

[24] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjærgaard. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 130–138, March 2014.

[25] A. Sarkar, S. Agarwal, and A. Nath. Li-fi technology: Data transmission through visible light. *International Journal*, 3(6), 2015.

[26] D. Tsonev, H. Chun, S. Rajbhandari, J. J. McKendry, S. Videv, E. Gu, M. Haji, S. Watson, A. E. Kelly, G. Faulkner, et al. A 3-gb/s single-led ofdm-based wireless vlc link using a gallium nitride μled. *IEEE Photon. Technol. Lett.*, 26(7):637–640, 2014.

[27] Q. Wang, M. Zuniga, and D. Giustiniano. Passive communication with ambient light. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 97–104, New York, NY, USA, 2016. ACM.

[28] G. Woo, A. Lippman, and R. Raskar. Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 59–64. IEEE, 2012.

[29] Y. Yang, J. Hao, J. Luo, and S. J. Pan. Ceilingsee: Device-free occupancy inference through lighting infrastructure based led sensing. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 247–256, March 2017.

[30] C. Zhang and X. Zhang. Pulsar: Towards ubiquitous visible light localization. 2017.

[31] K. S. Zigangirov. Synchronization of pseudorandom signals. In *IEEE Press Series on Digital & Mobile Communication*, pages 255–299. John Wiley & Sons, Inc., Hoboken, NJ, USA, April 2004.