Wearables Can Afford: Light-weight Indoor Positioning with Visible Light

Zhice Yang[1†] zyangab@connect.ust.hk
Zeyu Wang[1†] zwangas@ust.hk
Jiansong Zhang[2] jiazhang@microsoft.com
Chenyu Huang[3] hcyray@gmail.com
Qian Zhang[1] qianzh@cse.ust.hk

[1]CSE, Hong Kong University of Science and Technology [2]Microsoft Research Asia [3]Wuhan University
[†]Co-primary Authors

ABSTRACT

Visible Light Positioning (VLP) provides a promising means to achieve indoor localization with sub-meter accuracy. We observe that the Visible Light Communication (VLC) methods in existing VLP systems rely on intensity-based modulation, and thus they require a high pulse rate to prevent flickering. However, the high pulse rate adds an unnecessary and heavy burden to receiving devices. To eliminate this burden, we propose the polarization-based modulation, which is flicker-free, to enable a low pulse rate VLC. In this way, we make VLP light-weight enough even for resource-constrained wearable devices, *e.g.* smart glasses. Moreover, the polarization-based VLC can be applied to any illuminating light sources, thereby eliminating the dependency on LED.

This paper presents the VLP system PIXEL, which realizes our idea. In PIXEL, we develop three techniques, each of which addresses a design challenge: 1) a novel color-based modulation scheme to handle users' mobility, 2) an adaptive downsampling algorithm to tackle the uneven sampling problem of wearables' low-cost camera and 3) a computational optimization method for the positioning algorithm to enable real-time processing. We implement PIXEL's hardware using commodity components and develop a software program for both smartphone and Google glass. Our experiments based on the prototype show that PIXEL can provide accurate real-time VLP for wearables and smartphones with camera resolution as coarse as 60 pixel $\times$ 80 pixel and CPU frequency as low as 300MHz.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: ; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design, Experimentation, Measurement

MobiSys'15, May 18–22, 2015, Florence, Italy.

ACM 978-1-4503-3494-5/15/05 ...$15.00.
http://dx.doi.org/10.1145/2742647.2742648.

Keywords

Indoor Localization; Visible Light Communication; Polarization; Mobile Devices; Wearables

1. INTRODUCTION

We envision indoor positioning as being an indispensable feature of smartphones and wearables (*e.g.*, smart glasses) in the near future to support indoor navigation as well as plenty of location based services in shopping malls, supermarkets, office buildings, *etc.* To realize this goal, we need technologies that can provide high positioning accuracy as well as being light-weight enough to run in resource-constrained mobile devices, as these devices (especially wearables) are normally equipped with CPU, memory and camera that are optimized more for power-efficiency than performance.

Visible Light Positioning (VLP) [25, 26] is an emerging positioning technique that broadcasts anchor locations through *Visible Light Communication* (VLC). Benefited by densely deployed indoor lamps, VLP can easily achieve sub-meter accuracy in indoor localization. Compared with Wi-Fi and other RF based approaches which normally provide accuracy in meters, VLP holds the promise for beneficial applications such as retail navigation and shelf-level advertising in supermarkets and shopping malls.

However, we observe that in existing VLP systems, receiving devices are heavily burdened in VLC to avoid light flickering, which is caused by their intensity-based modulation method. As human eyes are sensitive to low rate changes in light intensity, the lamps have to transmit pulses at a high rate (over 1 kHz) to prevent flickering. Since the pulse rate far exceeds the camera's sampling capability (30fps), the design of the receiving side has to incorporate hardware modification to mobile devices – an additional customized light sensor is required and it relies on cumbersome calibration for the received signal strength to work properly [26] . A recent interesting idea is to leverage the rolling shutter effect of the CMOS image sensor to decode high-rate pulses from the high-resolution image [25, 32, 36]. Although its effectiveness has been shown in high-end smartphones with high-resolution (40 megapixel) cameras, the decodable distance is very limited in middle-end smartphones with 10 megapixel cameras or smart glasses with 5 megapixel cameras. Moreover, the high-resolution image incurs huge computation which requires a long processing time or cloud off-loading, therefore it lengthens response time, increases energy cost and sacrifices reliability.

The heavy burden on receiving devices motivated us to try to eradicate the light flickering. Our idea is to modulate the light's polarization instead of its intensity for communication. This paper

presents our VLP system PIXEL, which realizes this idea. PIXEL enables a light-weight VLP that is even affordable by wearables (Google glass), without incurring hardware modifications or computational off-loading. PIXEL also makes other types of illuminating light beyond LED light (even sun light, §3) usable for transmitting location anchors, therefore eliminating the potential barriers to its deployment.

The design of PIXEL's polarization-based VLC is inspired by the *Liquid Crystal Display* (LCD). We borrow *polarizer* (§2.2.1) and *liquid crystal* (§2.2.2) from LCD as PIXEL's major components. A challenge in PIXEL's design is that the SNR in the channel of the polarization-based VLC differs dramatically in different receiving orientations. To address this problem, we propose to add a *dispersor* to PIXEL's VLC transmitter and employ a novel modulation scheme, called *Binary Color Shift Keying* (BCSK) (§3.1). With the help of these two modulation techniques, PIXEL is able to achieve reliable channel quality despite the user's mobility. Moreover, on the receiving side, PIXEL also incorporates novel system designs to combat the uneven sampling in low-cost cameras (§3.2). The idea is to exploit the operating system's clock to obtain more accurate timing for video frames. Finally, in order to make the system as light-weight as possible, we optimize all the algorithms and their implementations. Particularly, we optimize the implementation of the camera-based localization algorithm and achieve orders of magnitude performance gain (§3.3).

We prototype PIXEL's transmitters with commodity components and develop a communication and localization program for both Android smartphone and Google glass. Our experiments based on the prototype system show:

- We enable polarization-based VLC between PIXEL's transmitters and camera-equipped smart devices. Reliable communication over a long distance of 10-meters only requires an image/video capturing resolution of 120 pixel$\times$160 pixel.
- We enable a fast positioning algorithm in mobile devices. Our optimization reduces the processing time of the camera-based positioning algorithm by three orders, from seconds to milliseconds.
- We enable realtime visible light positioning in resource-constrained devices and the average positioning time is less than 3 seconds. Even when the CPU frequency is as low as 300MHz, PIXEL's light-weight design works properly.

In this paper, with the design of the PIXEL system, we make following contributions:

- We identify the problem in existing VLP systems - the heavy burden on receiving devices, and propose a polarization-based VLC to address it. As far as we know, we propose the first system design that leverages polarization-based VLC in open space.
- We propose to add a dispersor to the VLC transmitter and develop a novel VLC modulation scheme called BCSK. These two techniques make it possible for the system to provide reliable communication despite the device's mobility.
- We develop a novel OS-clock based downsampling scheme to handle the uneven sampling problem of cameras.
- We optimize the implementation of the VLC receiving program and the positioning algorithm, thereby making the system light-weight enough for smartphones and wearables.
- We implement a prototype of PIXEL and conduct a systematic evaluation on it. Experimental results validate our design.

2. OVERVIEW

This section provides an overview of PIXEL. We first introduce the background of VLP and the flickering problem which motivates our work. Then we introduce the basics of polarization-based VLC, including techniques for generating/detecting polarized light and modulating light's polarization. Finally, we introduce the idea of PIXEL' system which is inspired by the design of one pixel in LCD screen.

2.1 VLP and the Flickering Problem

Visible Light Positioning (VLP) relies on *Visible Light Communication* (VLC) to broadcast location information through modulated light beams. The receivers, which are usually mobile devices carried by humans, use the light sensor [26] or the camera [25] to receive the location information as well as measuring their relative position to the lamps to perform fine-grained positioning. The mechanism is very similar to Wi-Fi positioning except that the visible light is used as the carrier to carry beacon messages instead of the microwave signal. Benefiting from the high density of the indoor lamps, visible light positioning can achieve much better (submeter level) accuracy than Wi-Fi positioning.

A major headache in VLP is the flickering problem. In current VLP designs, communication is based on modulating the light's intensity. Optical pulses are generated and data bits are encoded using pulse rate or pulse width. The varying light intensity causes flickering which can make people uncomfortable or even nauseous [19, 34]. In order to avoid the negative impact of flickering, a high pulse rate has to be used to make the pulses unnoticeable, though the message size required for localization is small. For example, the existing designs use over-1kHz pulse rates to transmit 7-bit beacons [25] or over-10kHz pulse rates to transmit 16-byte beacons [26].

The forced high pulse rate indeed adds a heavy burden to receiving devices. Considering the camera as the most common visible light sensor, the normal frame rate of 30 frames per second (fps) [22] is much lower than the pulse rate, which means it is far from sufficient to recover the transmitted pulses if we take one light intensity sample from every frame[1]. An interesting idea to tackle this problem is to leverage the rolling shutter effect of CMOS image sensor [25, 36]. The effect means rather than scanning a scene all at once, the image sensor scans one line at a time. Rolling shutter increases the sampling rate by up to thousands of times (from tens of fps to tens of thousands of lines per second), therefore making it possible to cover the pulse rate. However, according to our experience, the usefulness of rolling shutter is limited for two reasons. First, the camera must have very high resolution if it is not close enough to the lamp. This is because the lamp is only a small portion of the image, but a certain number of occupied lines/pixels are required to carry a message. In other words, if the receiving device does not contain a high-resolution camera, the communication distance will be short. For example, on a Samsung Galaxy 4 with a 13-megapixel camera, we can reach up to 1.6 meters, and with Google glass's 5-megapixel camera, we can reach up to 0.9 meters. Second, processing high-resolution image requires very high computational power. As a consequence, the existing design [25] relies on a cloud/cloudlet server to process the image and decode the message. Besides camera and rolling shutter, another solution [26] leverages a customized light sensor which is capable of taking high-rate samples on light intensity. Different from camera-based solu-

[1]Even cameras specially designed for capturing slow-motion video, for example, the 240-fps camera in the latest iphone6, the frame rate is still not sufficient to cover the over-1kHz or higher pulse rate. We also do not expect this slow-motion capable camera to be widely equipped into low-end phones and wearables.

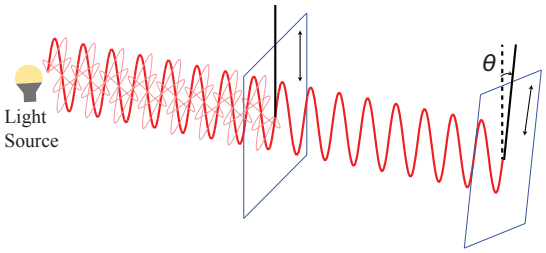


Figure 1: Generating and detecting polarized light.

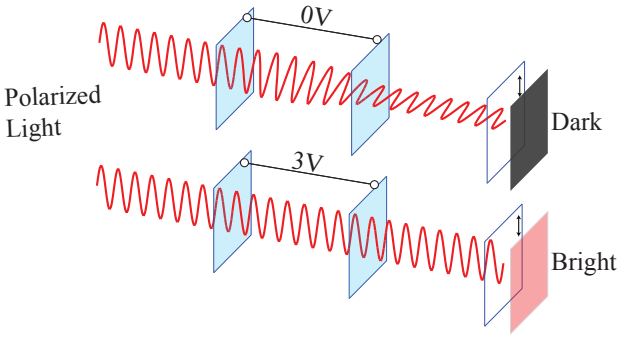


Figure 2: Modulating polarization to encode bits.

tions which use the angle-of-arrival based positioning algorithm, the light sensor based solution relies on *received signal strength* (RSS) technique to estimate the distance, this however are difficult to calibrate. Both hardware customization and RSS calibration are heavy burdens for VLP clients.

Finally, high-pulse-rate VLC also requires the transmitter to be a Light-Emitting Diode (LED), therefore, lamp replacement is needed if the existing lamps are not LED.

In this paper, we propose to exploit light polarization to realize low-pulse-rate VLC, so as to avoid the heavy burdens mentioned above and enable a light-weight VLP design that is affordable on wearable devices.

2.2 Basics of Polarization Based VLC

Polarization is a property of light waves that they can oscillate with more than one orientation. Similar to oscillating frequency which is another property of light, polarization preserves during light propagation in the air. However, different from oscillating frequency which determines color and thus can be perceived by human eyes, light's polarization is not perceptible [21]. This human-imperceptible nature has been proved very useful in that the polarization has led to the invention of Liquid Crystal Display (LCD) techniques, in which screens indeed always emit polarized light [39].

In the following subsections, we introduce two basic techniques in polarization-based VLC which solve the problems of generating/detecting polarized light and modulating light's polarization.

2.2.1 *Generating/Detecting Polarized Light*

Most of the common light sources, including sun light and man-made lighting systems, *e.g.* fluorescent light, incandescent light, *etc.*, produce unpolarized light which contains light of mixed polarization. The technique to convert unpolarized light into polarized light, called *polarizer*, is mature and very inexpensive. For example, the polarizer we use in this paper is a thin film, which costs roughly $0.01 per lamp. Polarizer is also widely used to detect polarized light and filter out certain unwanted light[2]. The theory behind polarizer is the Malus' law. That is, if the polarized light incident to a polarizer has intensity I_0, the passed light will have intensity I_θ, which is determined by the bearing angle θ between the polarization of the light and the polarizing direction of the polarizer:

$$I_\theta = I_0 \cos^2 \theta \tag{1}$$

Therefore, if the incident light has polarization in a direction parallel to the polarizer, *i.e.* $\theta = 0$, the light will pass through the polarizer without being attenuated. Otherwise, if their directions are perpendicular, *i.e.* $\theta = 90^\circ$, the incident light will be entirely blocked by the polarizer. Malus' law can be applied to the case of converting unpolarized light into polarized light by considering unpolarized light to be a mix of polarized light with all possible polarizations. Figure 1 illustrates the process of generating and detecting polarized light according to Malus' law.

[2]For example, both polarized sunglasses and photographic polarizing filter are used to eliminate unwanted reflections.

2.2.2 *Modulating light's Polarization*

Modulating polarization of visible light using *liquid crystal* (LC) is also a mature and inexpensive technique. In this paper, we use Twisted Nematic (TN) liquid crystal which is very popular in the LCD screens of personal computers, digital watches, *etc.* TN liquid crystal costs only $0.03 per cm^2 or $1.5 per lamp in our design.

We explain how to modulate polarization to encode bits using Figure 2. The two surfaces, between which a variable voltage is applied, are the surfaces of a TN liquid crystal layer. TN liquid crystal has the characteristic that when a voltage is applied, its molecules will be realigned – twisted/untwisted at varying degrees depending on the voltage. Most commodity TN liquid crystal components are manufactured to have a 90° twist when no voltage is applied. Therefore a beam of polarized light will be rotated by 90° while passing through the liquid crystal, as shown in the upper sub-figure. If we use a polarizer parallel to the incident light to detect the light, as shown in the figure, the light will be entirely blocked by the polarizer. Then when we increase the voltage, the liquid crystal will untwist and the polarizer will become brighter. Once a certain voltage level is reached, *e.g.* $3V$ as in the figure, the liquid crystal will become totally untwisted and the polarizer will show the brightest state. In this way, we can tune the applied voltage to modulate the polarization and encode '0' and '1' using dark and bright states[3].

2.3 PIXEL – A Pixel of LCD

Our system, PIXEL, is inspired by the design of the pixels in LCD. As shown in the upper sub-figure of Figure 3, a pixel of LCD consists of two polarizer layers and a liquid crystal layer in between. The backlight generates unpolarized light which is converted into polarized light by polarizer 1. The liquid crystal layer twists/untwists the light's polarization according to the applied voltage. Polarizer 2 has a fixed angle of 90° in the polarizing direction with polarizer 1. Therefore, the strength of the emitted light is controlled solely by the voltage applied on the liquid crystal layer. Finally, a lot of pixels each of which has an independent control voltage constitute an LCD screen.

The lower sub-figure illustrates the VLC design of PIXEL. We convert a lamp into PIXEL's VLC transmitter by adding a polarizer and a liquid crystal layer. The transmitter thus emits polarized

[3]In theory, we can encode more bits using multiple levels of brightness. However, in this paper we only encode 1 bit to get rid of complicated calibration.

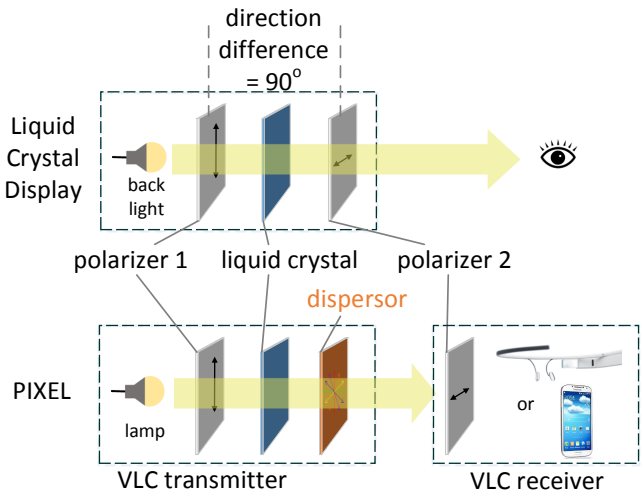


Figure 3: PIXEL is inspired by the design of LCD.

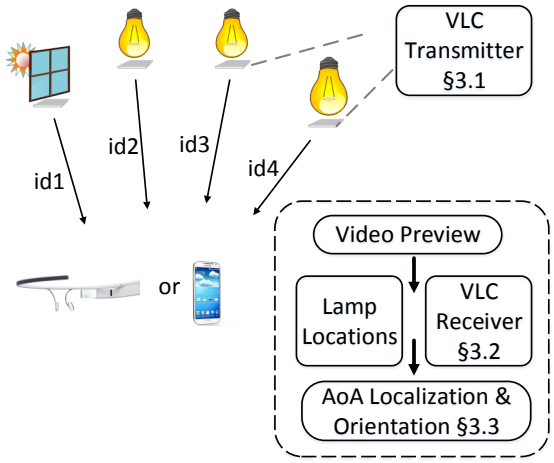


Figure 4: PIXEL Architecture.

light. Light's polarization can be changed by the voltage applied to the liquid crystal layer. The change of light's polarization cannot be perceived by human eyes[4] or directly captured by cameras. PIXEL's VLC receiver is a mobile device which equips a camera, *e.g.* smart glasses and smartphones. We add another polarizer to the camera for the purpose of detecting the changes of light polarization. As discussed in §2.2.2, the polarizer converts polarization changes into brightness changes that can be captured by the camera. We notice that the effect of adding polarizer to the camera, considering normal photo taking and video recording, is very similar to the use of polarizing filter in photography [28], *i.e.*, the polarizer does not bring noticeable changes except a slightly reduction in the strength of the ambient light.

However, from Figure 3, we can observe an important difference between PIXEL and LCD. In PIXEL, the polarizing direction between polarizer 1 and polarizer 2 does not always remain constant, because the receiving device may move and rotate with the person who holds it. This varying angle is one of the major challenges in our design. In the worst case the receiving camera will capture almost the same light intensity in the two polarization states. To tackle this problem, we leverage the camera's ability of capturing chromatic images and add a dispersor to disperse light of different colors to different polarizations. The details as well as other challenges and solutions are elaborated on in the next section.

3. PIXEL DESIGN

PIXEL presents a system design which makes visible-light based accurate indoor positioning affordable in wearables and low-end smartphones. As shown in Figure 4, every location beacon periodically sends out its identity through visible light communication. A wearable device or smartphone captures multiple beacons using the camera's video preview. Relative locations of beacons are obtained directly from the video. The identities of beacons are decoded by PIXEL's VLC receiver. Finally, with the help of a location database indexed by beacon identities, both location and orientation of the receiving device are determined using *angle-of-arrival* (AoA) based algorithms.

In PIXEL, it is possible to make use of any type of visible light source to construct a location beacon. As illustrated in Figure 4, a location beacon may consist of a lamp (not just the LED lamp, but any lamp) and a PIXEL's VLC transmitter. A location beacon may also consist of a window with incident sun light and a PIXEL's VLC transmitter.

In the following of this section, we elaborate on three main components of PIXEL, *i.e.* VLC transmitter, VLC receiver and the AoA based localization/orientation algorithm, to explain the challenges as well as our solutions in the design.

3.1 VLC Transmitter

The VLC design in PIXEL is borrowed from LCD, as has been described in §2.3. The VLC transmitter contains a polarizer and a liquid crystal layer that can convert the unpolarized illuminating light into polarized light and modulate the light's polarization using a control voltage. Then when the light propagates to receiving devices, polarization changes will be converted into intensity changes by another polarizer and identified by the receiving camera. Finally, the VLC transmitter also contains a simple electronic module to encode information, which translates the beacon's location identity into a series of control voltages.

A big challenge to PIXEL's VLC comes from the receiver's mobility, as has been briefly mentioned in §2.3. Unlike the two fixed polarizers in LCD, in PIXEL, though the polarizer in VLC transmitter is fixed (attached to the lamp), the polarizer in the receiver is mobile. Therefore, the two polarizers could end up with arbitrary angles in their polarizing directions. Since this angle determines the intensity of light, which is finally captured by the receiving camera, angle ambiguity will lead to a varying *intensity difference* (intensity-diff) between '1' and '0' states, meaning varying *signal-to-noise-ratio* (SNR) for communication. In the worst case, the intensity-diff will approach zero, therefore the encoded states will be totally undecidable. In order to quantify the relation between the intensity-diff and the polarizing angle between two polarizers, we build a mathematical model using Malus' law. Theoretical results based on this model are shown in the rightmost side of Figure 5. We can see when the angle is roughly in a 20° range around 45°(or 20% possibility), the intensity-diff drops quickly towards zero and can hardly fulfill the SNR requirement for communica-

[4]The change of light's polarization can be perceived if a person wears polarized sunglasses. However, we believe the case that someone wears polarized sunglasses in indoor environments is very unusual.

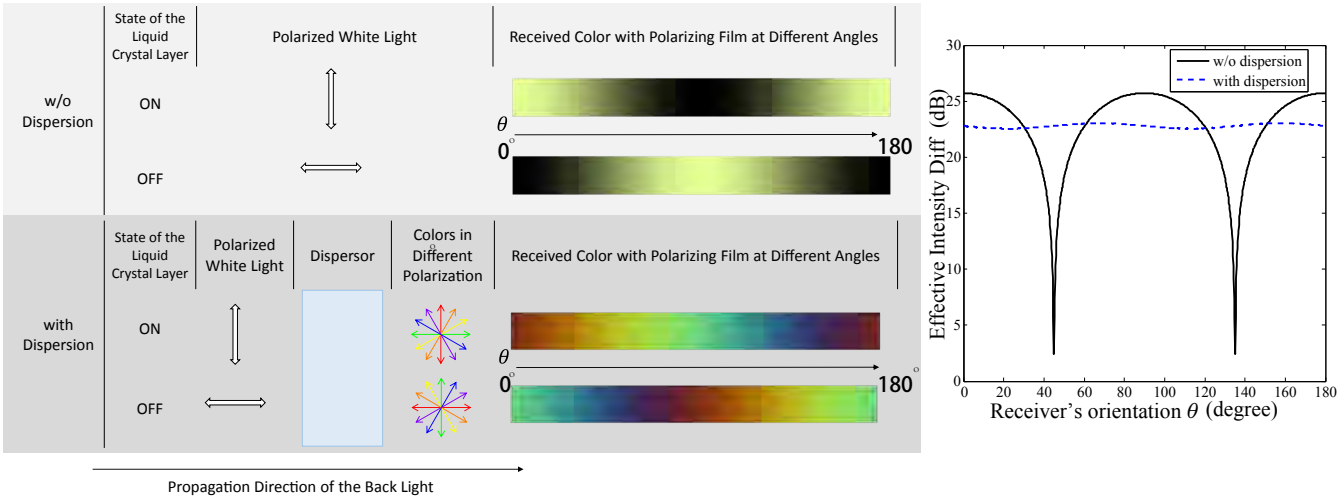


Figure 5: Comparison between BCSK (w/ dispersion) and BISK (w/o dispersion). In BISK as in the upper sub-figure, the captured white light in receiving camera can only be quantified using light intensity. In BCSK as in the lower sub-figure, a dispersor is added in the VLC transmitter, the captured chromatic light can be quantified using a vector in RGB domain (*i.e.* color). The right sub-figure shows in BISK, intensity-diff of two encoded states largely fluctuates with receiver's orientation, whereas in BCSK, the effective intensity-diff is stable.

tion. We also verified the result using experiments, in the angle of 45°, the intensity-diff is indeed very low.

A possible solution is to ask users to rotate the receiving devices. However, we believe it would be a cumbersome task. Since the polarizing angle is not perceivable by eyes, there is hardly any information to guide users on how to rotate. Also, rotating wearable devices could require big weird gestures.

To tackle this problem, we exploit the camera's capability of taking chromatic video. Recall the rainbow, the white light consists of multiple colors of light. The basic idea is to disperse the polarized light emitted from the liquid crystal layer into different colors, so that the receiving camera can capture different colors in different orientations after the filtering of the receiving polarizer. We fine tune the color dispersion to guarantee that, in any orientation, the captured colors in two encoded states will be largely different. We term this color-based modulation as *Binary Color Shift Keying* (BCSK). Next, we describe the dispersor and BCSK in details.

3.1.1 Dispersor and BCSK

Some materials have a property called *optical rotatory dispersion* (ORD), *i.e.* they can rotate the light's polarization differently with different frequencies (colors) of light [10]. These substances may be solid such as quartz, liquid such as alcohol or gas such as Hexadiene. This property has been studied for decades and widely used in industry, chemistry and optical mineralogy. For example, the method for measuring the blood sugar concentration in diabetic people is based on the ORD property of the sugar. In this paper, we use a liquid substance called Cholest [13] as our dispersor for the ease of fine-tuning the thickness and dispersion.

We use Figure 5 to illustrate BCSK by comparing it with *Binary Intensity Shift Keying* (BISK). We can see the beam of the polarized white light emitted from the liquid crystal is dispersed into multiple beams of polarized light with each having a unique color and polarization. The receiving polarizer passes through one beam of light according to its polarizing direction and filters out other beams. Therefore, only a certain color is captured by the receiving camera. When the encoded state is changed by the control voltage on the liquid crystal layer, the color of the captured light changes as well. Although the captured color still differs with the orientation of the receiving device, with proper dispersion we can guarantee the received colors largely differ between two encoded states in any orientation.

Next, we describe how to tune the dispersion for the purpose of guaranteeing a reliable SNR in all orientations.

3.1.2 Tuning Dispersion and Optimizing SNR

Dispersion tuning is performed by choosing the best thickness for disperor. The best thickness is mainly determined by the substance of dispersor. The spectrum of the illuminating light may slightly affect the best thickness, but this effect is very small because most illuminating light sources emit white light which represents a similar flat spectrum. When the substance of dispersor and the type of light is determined, the best thickness is also determined.

The target we optimize for is a metric called *effective intensity-diff*. It quantifies the color difference between two encoded states which is measured by camera. Therefore, it represents the received signal strength of the communication channel. Specifically, as shown in Figure 6, the colors captured by camera are mapped into a *{Red, Green, Blue}* (RGB) space. Given certain receiver orientation, the two colors in two encoded states can be indicated using two vectors $\vec{s_1}$ and $\vec{s_2}$. The difference between two colors is therefore $\vec{\delta} = \vec{s_1} - \vec{s_2}$. Then, effective intensity-diff is defined as $|\vec{\delta}|$.

We formulize the optimization problem and solve the best thickness L_{best} as follows.

When a beam of polarized light passes through a dispersor, it is dispersed into multiple beams of polarized light in which each has a unique frequency (color). The polarization of a beam with frequency f is rotated by an angle $\Delta\phi_f$, which is determined by:

$$\Delta\phi_f = n(f)L, \tag{2}$$

where L is the thickness of dispersor, $n(f)$ is a function of f which describes the rotation of light polarization in a unit thickness. $n(f)$ describes the property of a substance.

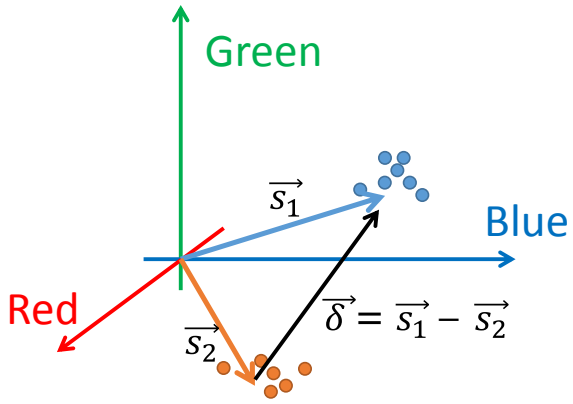


Figure 6: Effective Intensity-diff $|\vec{\delta}|$.

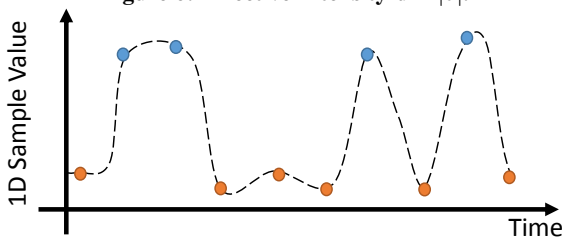


Figure 7: Result of dimension reduction. These 1D samples reduced from 3D samples in Figure 6 constitute the received waveform.

After the filtering by the receiving polarizer, the light captured by camera is described with a vector $\{I_R, I_G, I_B\}$ in RGB space. $\{I_R, I_G, I_B\}$ can be derived using the Malus' law:

$$I_N(\theta, L) = \int_{f_{min}}^{f_{max}} p_N(f) \cos^2(n(f)L - \theta)\, df, N \in \{R, G, B\}$$

where θ is the polarizing angle between transmitter's polarizer and receiver's polarizer. f_{min} and f_{max} are the upper/lower frequency bounds of the light's spectrum. $p_N(f)$ describes the light spectrum projected on the dimention N of RGB space. $p_N(f)$ is determined by the camera's chromatic sensor. From this formula, we can observe that the light captured by camera is determined by both optical rotatory dispersion ($n(f)L$) of the substance and the light spectrum ($p_N(f)$), as mentioned above.

The final goal of the optimization problem is to maximize the minimal effective intensity-diff in all receiving orientations. Considering the two encoded states between which the polarizing angle is rotated by 90°, we can formulize the optimization problem as follows:

$$\arg\max_{L} \min_{\theta} \sqrt{\sum_{N \in \{R,G,B\}} \left(I_N(\theta, L) - I_N(\theta + 90°, L)\right)^2}$$

To solve this problem and get L_{best}, we need to know $n(f)$ and $p_N(f)$ in advance. $n(f)$ and $p_N(f)$ can be obtained from specification or calibration. In our work, we calibrate the $n(f)$ of our dispersor by ourselves and derive $p_N(f)$ from specification [8] and calibration results in literature [2]. We found $p_N(f)$ almost does not affect the vlaue of L_{best}, as shown in Figure 8[5]. We believe that

[5]We use the normalized thickness to plot the trend for dispersors with different materials. Given the condition that the $n(f)$ is approximately linearly related to f, the trend of the Figure holds.

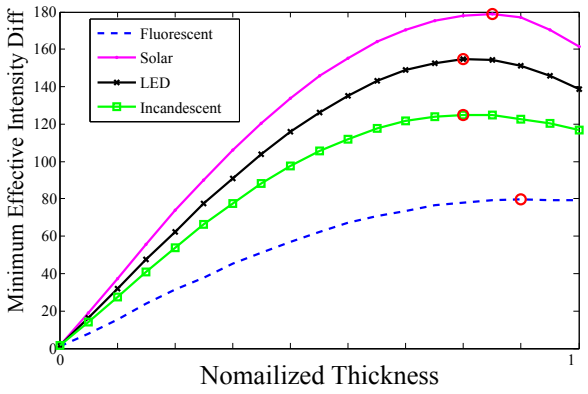


Figure 8: The optimal thickness of dispersor is almost the same with various types of illuminating light, as most illuminating light is white light.

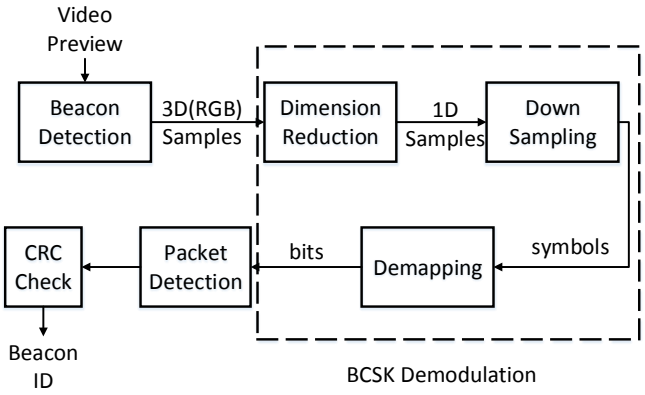


Figure 9: Structure of PIXEL's VLC receiver.

it is because most illuminating light sources emit white light which has a similar flat spectrum.

To compare BCSK with BICK, we calculate the effective intensity-diff under L_{best} in all possible receiver orientation (represented by θ). Results are shown in the rightmost of Figure 5. We can see the effective intensity-diff is quite stable in all the receiver orientations.

To conclude, the dispersor in PIXEL's VLC transmitter and the color-based modulation scheme BCSK well solve the problem of varying SNR caused by users' mobility.

3.2 VLC Receiver

PIXEL's VLC receiver decodes beacon's identity from camera's video preview. Figure 9 shows the structure of the VLC receiver.

The first step is beacon detection. In order to save computation, we first downsample the video to contain less pixels. According to our experience, 100 × 100 pixels will be sufficient for decoding in a distance within 10 meters. Then we calculate the difference between every two consecutive frames to detect the VLP beacons. In order to make the detection reliable, we add up the differences from 10 consecutive frames. Usually a detected beacon contains 10 or more pixels which contain its color value. We finally average these pixels to generate a 3-dimensional ($\{R, G, B\}$) color sample

Length "1" means the the difference of the rotated angles between f_{min} and f_{max} is 180°: $|\Delta_{f_{max}} - \Delta_{f_{min}}| = 180°$ and length "0" means no difference in rotation, *i.e.* L=0.

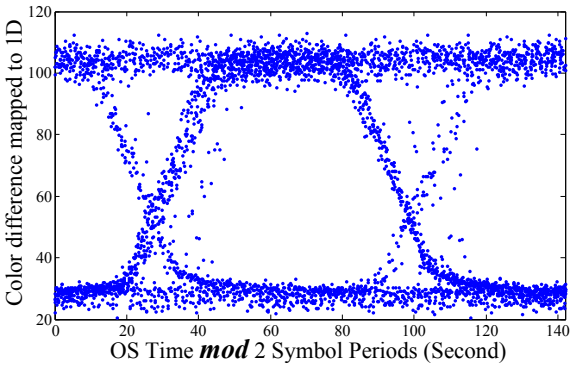


Figure 10: 3000 video samples overlay with their sampling timestamp mod two symbol periods. The eye patten indicates the clock drift between the OS clock and VLC transmitter is negligible.

from every video frame. Once a beacon is detected, one frame without downsampling is buffered for calculating its image location for the localization algorithm in §3.3.

As illustrated in Figure 6, these 3D samples in $\{R, G, B\}$ space are mainly distributed around two points, which represent the two encoded states. Since there are only two points/states, for the ease of further processing, we perform dimension reduction to convert 3D samples into 1-dimensional samples as the first step of BCSK demodulation. The requirement of dimension reduction is to preserve the distance between samples of the two encoded states. Therefore, we perform reduction by projecting these 3D samples to the difference vector between the two states, *i.e.* $\vec{\delta}$ in Figure 6. The projection from a 3D sample $\vec{s}$ to 1D sample s is calculated by:

$$s = \frac{\vec{s} \cdot \vec{\delta}}{|\vec{\delta}|}, \tag{3}$$

Figure 7 shows the 1D sample waveform reduced from 3D samples in Figure 6. To determine $\vec{\delta}$, we calculate difference vectors between every two consecutive 3D samples, then take the average of those difference vectors with large norm and right direction as $\vec{\delta}$.

In PIXEL's transmitter, we set the symbol rate roughly the half of the camera's sampling rate for the purpose of meeting the sampling theorem. As the low cost cameras in wearables are not stable in the sampling interval, the downsampling process is non-trivial and we will elaborate on it in §3.2.1.

The output of the downsampling module is symbols. Since the symbols are in 1 dimension (scalar), demapping is equivalent to bit arbitration which determines '1' or '0' from every symbol. Similar to classic 1-bit arbitrator designs, we set the average of all symbols as the threshold.

The final step is to recover packets from the bit stream. In PIXEL, we use a 5-bit header for packet detection and synchronization. Since the packet is short, we use a 4-bit CRC for error detection and do not perform channel coding or forward error correction. More details about the packet design is shown in §4.

3.2.1 *Combating the uneven camera sampling*

The downsampling (decimation) algorithm in classic digital signal processing [30] cannot be applied to PIXEL's VLC receiver. A basic assumption behind these algorithms is that digital samples are taken with a constant interval from the analog signal. However, this assumption does not hold in the low-cost cameras. As [22] pointed out, cameras in mobile devices do not offer fixed sampling rate (frames per second). Specifically, the sampling rate differs on different cameras, and also differs under different levels of ambient

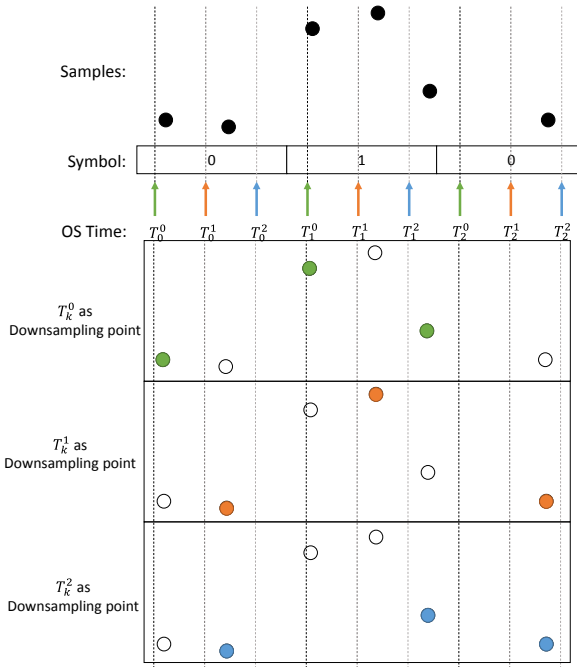


Figure 11: Downsampling example on real samples. Each T_0^j determines a sampling sequence, PIXEL chooses T_0^1 since it has the best decoding SNR.

light. Moreover, the sampling rate may even vary during a single video recording.

In PIXEL, we leverage the device's OS clock to perform downsampling. The core observation is that the clock drift between the receivers' operating system and the transmitter is negligible in the time scale of several packet durations (seconds). This can be proven by Figure 11, where every sample is timestamped by the OS clock and plotted with the time modulo of two symbol intervals. The cross lines between the upper line and the lower line in the figure are the boundary of the symbols. When there is a clock drift between two clocks, the boundary will keeps moving. However, as we can see from the figure, the clock drift can hardly be observed.

We downsample the uneven samples according to the OS clock. Although the sampling interval is changing, the relative optimal sampling time in one symbol period is stable. In Figure 10, samples around 60 can achieve higher SNR than those close the symbol boundary. In order to find the optimal sampling time, we use a method similar as searching. Several candidates of the sampling time are used to downsample the incoming wave simultaneously, and the best candidate in SNR gains weight in being the final choice. We choose the candidate which has the largest cumulative weight as our sampling time for this symbol period. Note that the algorithm can automatically adapt to different start conditions and bootstrap the sampling time to the optimal choice. Once the sampling time is determined, we choose the nearest input point as the downsampling result.

We explain the detailed downsampling method through Figure 11. PIXEL maintains sampling candidates T_k^j with same time interval: $T_k^j = T_0 + kC + jC/N, k \in \mathbb{N}, j \in \{1, .., N\}$, where k represents different symbol period and j represents the incremental searching step within one symbol period. C is the time for one symbol period. Once T_0^0, for example, in the figure is selected as the first sampling time, then $T_k^0 = T_0^0 + kC$ is the corresponding sampling sequence. According to the nearest rule, the selected samples for each T_0^j is shown in the bottom half of Figure 11. Ap-

Algorithm 1 Adaptive Downsampling

Input:

N: The number of the sampling time candidates.
L: The length of historical samples used to calculate SNR.
k: The symbol period k.
$T_k^1 \sim T_k^N$: Sampling time candidates at symbol period k.
$t_i^s \sim t_j^s$: Timestamp of samples, $t_i^s <= T_k^1 < T_k^N <= t_j^s$.
$x_i^s \sim x_j^s$: The corresponding samples at $t_i^s \sim t_j^s$.
$Y_{k-L}^i \sim Y_{k-1}^i$: The historical downsampled samples by T_k^i.
S_k^i: The SNR of the samples sequence $\{Y_{k-L}^i \sim Y_{k-1}^i\}$
E_{k-1}: The index i of the best T_{k-1}^i.

Output:

E_k: The index i of best T_k^i.

for each $i \in \{1, ..., N\}$ **do**
  Find the nearest t_j^s so that $t_j^s <= T_k^i <= t_{j+1}^s$.
  Let Y_k^i=x_j^s.
end for
for each $i \in \{1, ..., N\}$ **do**
  Calculate the S_k^i.
end for
$E_k^{'} = i$ where $S_k^i = \max_{i \in [1,N]}(S_k^i)$.
Calculate E_k by averaging E_{k-1} with $E_k^{'}$.
return E_k

parently, the choice of the T_0^j affects the decoding quality. PIXEL selects them according to the decoding SNR. We define the SNR of a sequence of L samples $\{x_i\}, i \in \{0, ..., L-1\}$ as:

$$SNR = \frac{\big((mean(X_1) - mean(X_0)\big)^2/4}{\big(var(X_1)|X_1| + var(X_0)|X_0|\big)/L} \quad (4)$$

where $X_1 = \{x_i : decode(x_i) = 1\}$ and $X_0 = \{x_i : decode(x_i) = 0\}$. PIXEL keeps tracking and updating its choice of the sampling sequence to maintain the highest decoding SNR. The detailed algorithm is shown in Algorithm 1.

3.3 Positioning Algorithm

As a benefit of using a camera as the light sensor, comparing with other light sensors [26], PIXEL is able to derive both location and orientation from a captured video containing three or more location beacons. The positioning algorithm used is the same as Luxapose [25]. In this subsection, we describe our effort to optimize the implementation of the algorithm for the purpose of accelerating its execution, especially for wearables which equip low-frequency CPU.

We briefly describe the algorithm. The algorithm is a camera-based *angle-of-arrival* (AoA) localization method. Similar to other AoA based algorithms, a big advantage is that it does not require the difficult distance estimation between the receiver and location beacons. Instead, it leverages the properties of optical imaging that the mutual distance relations of beacons are distorted in the image according to the camera's location. Based on this knowledge, it takes two steps to localize the camera. First, the algorithm extracts the mutual distance relation of beacons in the image and searches for the optimal scaling factors K_i to match the real mutual distance relation. K_i describes how much the beacon i is zoomed in the image. Second, the real location $[T_x, T_y, T_z]$ of the camera is calculated according to K_i and each beacon's real location.

The camera-based AoA algorithm requires a large searching space to optimize the location value. When using normal searching methods, *e.g.* evaluating the optimization function at each point of a multidimensional grid of points, the localization process could bring

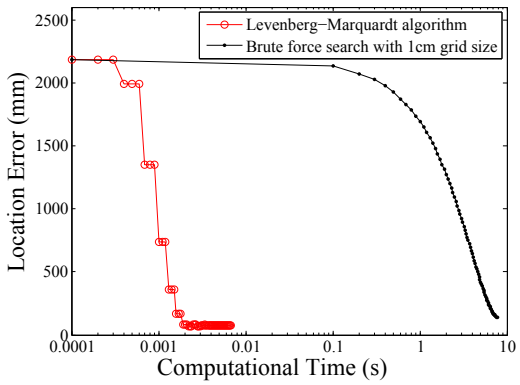


Figure 12: Time cost for different optimization methods.

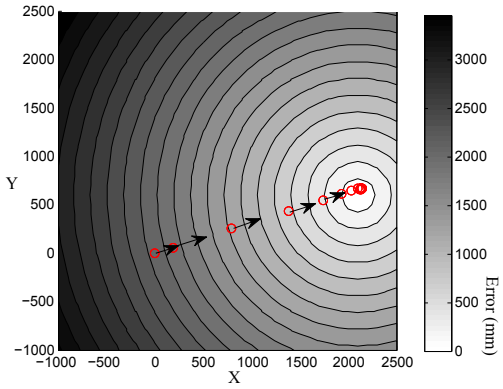


Figure 13: Optimizing the searching speed with gradient. This 2D map illustrates the searching space of the localization problem in the horizontal dimension. The value of the object function, *i.e.* localization error, is represented in brightness. The real location with minimum error is the brightest area. Initial searching starts from $[0, 0]$ and arrows plot the optimal direction for finding the real location.

significant calculation overhead to the mobile device, and thus the processing delay would be unacceptable. Dotted line in Figure 12 shows the time consumed in our smartphone when the normal searching is used. In order to find the location within 10cm accuracy, the processing time exceeds seconds, which is harmful for the user experience.

PIXEL reduces the searching space by leveraging the gradient of the optimization function to guide searching. Notice that the two optimization problems in AoA estimation are both non-linear least square problem, as they both have the following optimization object function:

$$\vec{x}_{opt} = \arg\min_x \sum_i f_i(\vec{x})^2$$

Optimization for the above function can be accelerated by searching along the direction of the gradient, which towards the minimum value(Figure 13). PIXEL adopts the widely-used Levenberg-Marquardt algorithm [29] which has a similar intuition. Circled line in Figure 12 shows significant acceleration of three orders in optimizing camera's location.

4. IMPLEMENTATION

Our implementation includes hardware part and software part. We prototype PIXEL's VLC transmitter using commodity compo-

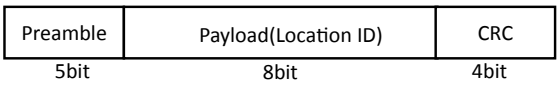


Figure 14: Message format of the location beacon.

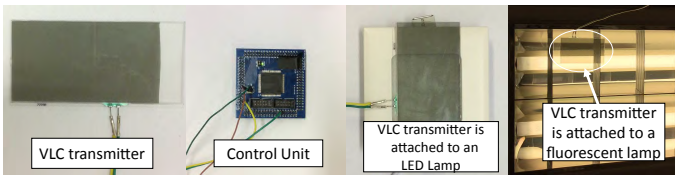


Figure 15: Photos of PIXEL's components. From left to right: (1) VLC transmitter including a polarizing film, a liquid crystal layer and a dispersor. (2) The control unit with Altera Cyclon 2 FPGA. (3) VLC transmitter attached to a LED lamp. (4) VLC transmitter attached to a florescent lamp.

nents except the dispersor which are constructed using liquid and transparent container. On the other side, we implement the VLC receiver and the positioning algorithm in a software program for both Android phones and Google glass.

4.1 PIXEL's Hardware

One VLC transmitter consists of three transparent optical layers and one control unit. The first layer is a linear polarized film [6], which functions as a polarizer to polarize the illuminating light. The second layer consists of two pieces of glasses with liquid crystal in between. We add electrodes on both glasses to control the applied voltage. This layer is a kind of simplified LCD with only one pixel, and many LCD manufacturers have the service for costuming LCD [4]. The third layer is a dispersor. We prototype this layer by filling the optical rotation material (Cholest) into a glass box. We choose the liquid material because it is easy to tune the thickness, but solid materials [7] with appropriate thickness could be a good choice for the real deployment. We adopt Altera's Cyclone II FPGA to control the voltage of the liquid crystal layer. Its clock rate is 40MHz and the output voltage is 0V/3.3V. The two electrodes of the liquid crystal layer are directly connected to two output ports of the FPGA to modulate the light's polarization.

Each PIXEL's transmitter works independently by facing to normal light sources such as solar light, LED light, fluorescent light and so on. As Figure 15 shows, normal light sources can thus be easily upgraded to have VLC ability with PIXEL's transmitter design.

The hardware of PIXEL's client devices requires no modification or customized accessories. The only requirement is a small piece of off-the-shelf linear polarizing film, and it is placed in front of the camera to detect and receive VLC signals.

4.2 PIXEL's Software

The packet format of VLC is designed for the indoor localization purpose. PIXEL's transmitter continuously broadcasts its location identity to receivers. The beacon packet contains a 5-bit preamble, an 8-bit data payload and a 4-bit CRC. The preamble is used for packet synchronization. Once the receiver detects a preamble from cross-correlation, it will start to decode the following packet payload which contains the identity of the beacon. We choose 8-bit as the length of an identity by considering the lamp density of the real world deployment. For the 4-bit CRC, the probability of false-positive detection is only 1.7×10^{-8} [24] in our scenario, which is small enough.

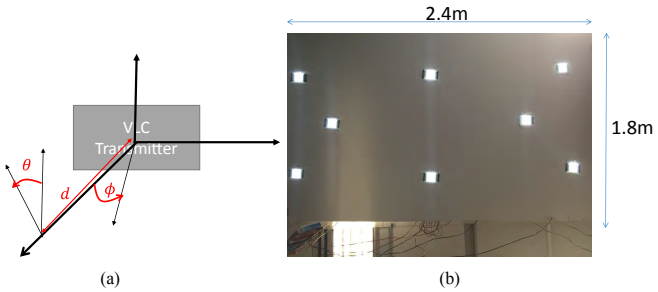


Figure 16: Evaluation Setting. (a) is the settings for single VLC pair. d is the distance from the receiver to the transmitter. θ is the orientation of the camera. ϕ is the viewing angle of the transmitter. (b) is the testbed for the VLP evaluation.

We implement the VLC decoding and the positioning algorithm in both Android system and an offline simulator[6]. The code can be easily ported to other platforms since it does not depend on specific APIs. In Android 4.1.4, it takes 1200 lines of codes and the size of the executable file (apk) is about 320kb. We use the preview data from the camera to process decoding in realtime. As the system will automatically adjust camera's exposure time according to the intensity of the received light, which may affect the BCSK decoding. We walk around this issue by manually locking the exposure time in the software. This is not an issue in new Android release [1], since more flexible control of exposure time is provided. Once enough beacon packets are received, we use the buffered video frame to calculate the locations of lamps in the image for the localization algorithm. Other required parameters (*i.e.* the focal length of the lens and the size of the image sensor) can either be read from the EXIF stamp of the camera or obtained from online specifications.

5. EVALUATION

In this section, we evaluate PIXEL's design based our prototype implementation. We first describe the evaluation setup, then we conduct experiments to verify the component design in PIXEL. After that, we thoroughly evaluate different impact factors on PIXEL's VLC subsystem and the computational cost of each part of the client application. Lastly, we present the overall performance of our VLP system.

5.1 Experiment Setting

Evaluation involving one single pair of transmitter and receiver is performed in controlled conditions. Figure 16 (a) shows controlled parameters related to user's mobility. d is the distance between the transmitter and the receiver. θ is the receiver's orientation whose rotation plane is parallel to the receiver's plane. ϕ is the viewing angle whose rotation plane is vertical to the transmitter's plane. Other impact parameters such as the downsampled resolution[7] and the exposure time are also controlled in the evaluation. If not mentioned, we take $d = 3m$, $\theta = 0°$, $\phi = 0°$, downsampled resolution = 120×160 and exposure time = $1/100s$ as default settings.

Evaluation involving multiple pairs of transmitter and receiver is performed with the testbed in Figure 16 (b). We fix eight PIXEL's

[6]We extract Presentation Time Stamp (PTS) from `ffmpeg` [3] to simulate timestamps of frames in recoded videos.

[7]The downsampling is used to reduce processing overhead as we mentioned §3.2, we fix the downsampled resolution to certain values, such as 120×160. The reason is to normalize different resolutions across cameras.

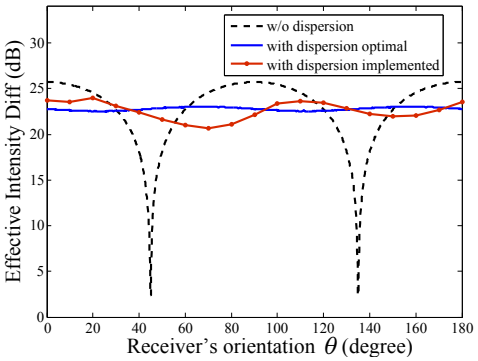


Figure 17: Evaluation of the dispersor design

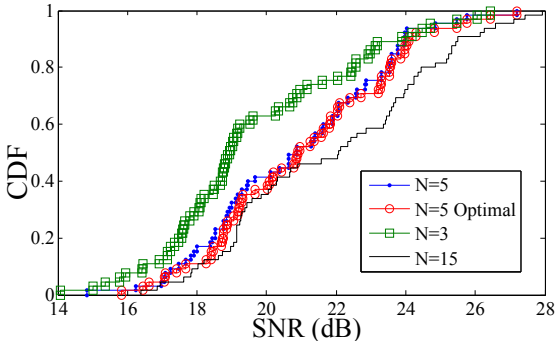


Figure 18: SNR of adaptive downsampling. We choose the appropriate number of searching candidates $N = 5$ in our design to balance the accuracy and computational cost. Moreover, our adaptive downsampling algorithm almost has the same performance with the optimal downsampling choice.

transmitters into a board which is elevated to the ceiling. The location of each transmitter is measured and mapped to the location ID in client's database. When the testbed is used for positioning evaluation, transmitters transmit their location ID through frame defined in §14. Under the testbed, we grid another board to serve as ground truth reference for client's location.

We use controlled random bit stream for transmission to measure BER and SNR. In these experiments, the transmitter repeatedly transmits a random pattern with a length of 100 bits, while the receiver keeps logging the received color values and the decoded bits. Since the length of the random pattern is long enough, we determine the start of the transmitted pattern through correlation. Therefore, the decoded bits can be used to calculate BER by comparing with the known random pattern. With the known random pattern as ground truth, the received color values can be used to calculate SNR according to formula (4).

All the experiments are done without a fixed ambient light source. In the morning, the ambient light is solar light and becomes fluorescent light in the evening. We adopt the 5W 10cm×10cm LED lamp as the light source of the transmitter for convenient setup. We also have performed similar evaluation experiments for fluorescent light and solar light, but we found little difference from LED.

5.2 Visible Light Communication

5.2.1 *Dispersor and BCSK*

Experiment: We study the effectiveness of the dispersor design in PIXEL. The evaluation is taken in a static situation, *i.e.* the transmitter stops transmitting and is either in on or off stage. At each stage, we take pictures of the light in different orientations from 0° to 360° in step of 10°. From the two pictures at the same angle but different stages, we compute the effective intensity diff according to the extract color of the light.

Results: The lined dot in Figure 17 shows the effective intensity diff of our dispersor prototype. Results from 180° to 360° are symmetrical to 180° to 0° and omitted. Compared with the dash line, which is the results without the dispersor, the dispersor design significantly increases the detectable difference for the on/off stages in degrees around 45°. Compared to the solid line, which is the situation with theoretical optimal dispersor design, our current prototype can be improved. The reason is not the hardness for manufacturing the dispersor, but lies in the fact that we can only purchase glass boxes with fixed thickness in the market. We choose the best existing thickness to prototype and we believe its performance is enough to illustrate our design.

5.2.2 *Adaptive Downsampling*

Experiment: We evaluate the adaptive downsampling algorithm through real data and the offline processing. 65 logged cases are collected by receiving VLC data from different locations and orientations. Each log lasts for 30s. According to the logged color sequence and timestamps, we use offline simulator to adjust the number N of the potential choices of the sampling time. In each choice of N, we calculate SNR for the whole sequence.

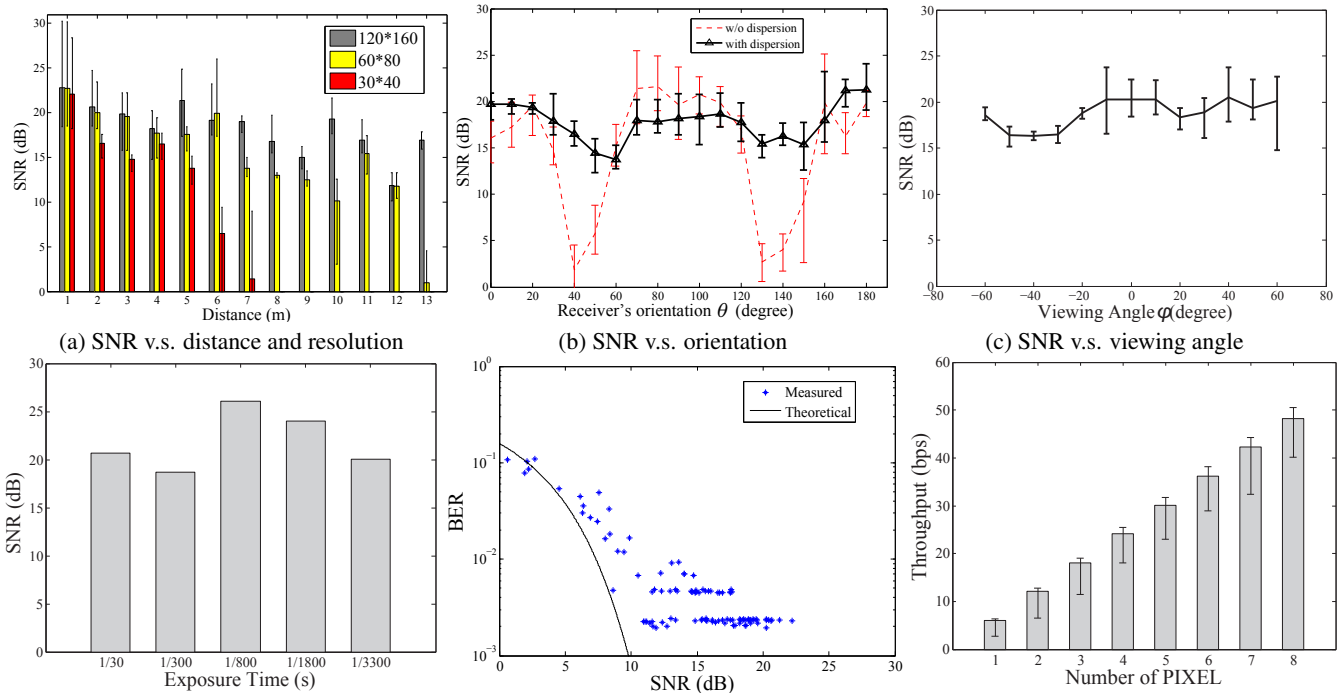
Results: Figure 18 shows the results. First, we compare the adaptive downsampling with the fixed downsampling, where a fixed sampling time is chosen during the whole decoding. According to our previous description for Figure 10, the optimal choice of the fixed sampling time is fixed but unknown. Adaptive downsampling bootstraps and approximates such optimal choice. Figure 18 shows that when $N = 5$ the adaptive downsampling has little difference from the optimal choice of fixed sampling. We note that the same conclusion holds for different N, but we omit the curves for clarity.

Second, we study the trend of the decoding SNR when N increases. Reasonably, the SNR curve of larger N is better. This is simply because the choice for sampling time is more fine-grained and flexible with larger N. However, the gain is bounded by the intrinsic noise in the sampling timestamp and the communication channel. Therefore, the SNR gain from $N = 3$ to $N = 5$ is more significant than the gain from $N = 5$ to $N = 15$, and the gain is even smaller when N is increased to larger values. According to the measurement of our receivers, the standard derivation of the sampling time is in $6ms$ level. So we choose $N = 5$, *i.e.* $1s/14/5 = 14ms$, to bound the random timing error. Other devices can easily obtain appropriate N according to their local measurement.

5.2.3 *End-to-End Communication*

Experiment: We study PIXEL's VLC subsystem against five impacting factors caused by user mobility and device diversity: the distance d, the receiver's orientation θ, the viewing angle ϕ, the downsampled resolution and the exposure time of the camera. The experiment for certain factor varies that factor to all the reasonable values, and fixes other factors.

Results: Figure 19 (a) shows the SNR against the distance and the downsampled resolution. The downsampled resolution and the distance determine the number of pixels of the transmitter in the received image, and thus affect the SNR in a similar way. Therefore, we list them in the same figure for better illustration. From the results, our system achieves the SNR higher than 15dB under 120×160 resolution within $13m$, which is sufficient for most in-



(a) SNR v.s. distance and resolution

(b) SNR v.s. orientation

(c) SNR v.s. viewing angle

(d) SNR v.s. exposure time

(e) BER v.s. SNR

(f) Rate v.s. number of PIXEL

Figure 19: PIXEL communication evaluation

door localization scenarios. In addition to that, the VLC system works even when the downsampled resolution is as low as 30×40. Notice that the SNR of the downsampled resolution of 60×80 and 30×40 drops to 0 at $13m$ and $7m$ respectively. The reason is that our decoding algorithm needs enough pixels to extract color. However, when the distance from the transmitter increases, the transmitter's pixels in the image decrease. This is not a problem since even $13m$ is enough for VLP in a normal building. Moreover, the size of the transmitter or the intensity of the background light can easily be increased to satisfy the possible requirement.

Figure 19(b) shows the VLC performance under different orientations. Without the dispersor, the SNR becomes undecodable when the view angle between the transmitter and receiver is around 40° or 130°. This result is in accord with our previous analysis about effective intensity diff in §5.2.1. On the other hand, the result is improved a lot when the dispersor is deployed. The SNR of the received data is around 16dB in all the orientations, implying that the VLC quality is stable despite different orientations.

Figure 19(c) implies that the performance of VLC is stable under different viewing angles from -60° to 60°, which is sufficient for normal localization scenarios. From the results, we also observe that the variance of the SNR in the positive and negative sides are asymmetric. This is caused by the viewing angle limitation of the prototyping TN LCD, and can be solved by using better TN LCD or other LCD such as In-Plane Switching LCD.

Figure 19(d) shows the SNR against different exposure time settings of the camera. The result implies that there is no obvious relationship between exposure time and SNR. The reason is that our decoding algorithm will use different areas for decoding according to the exposure time. When under short exposure time, the amount of the received light from the transmitter is small and the decoding algorithm uses the center part of transmitter to extract color. When under long exposure time, the center of the transmitter is overexposed and displays white. However, the color can be extracted from the glow of the transmitter's image. Therefore, our VLC is robust under different exposure settings.

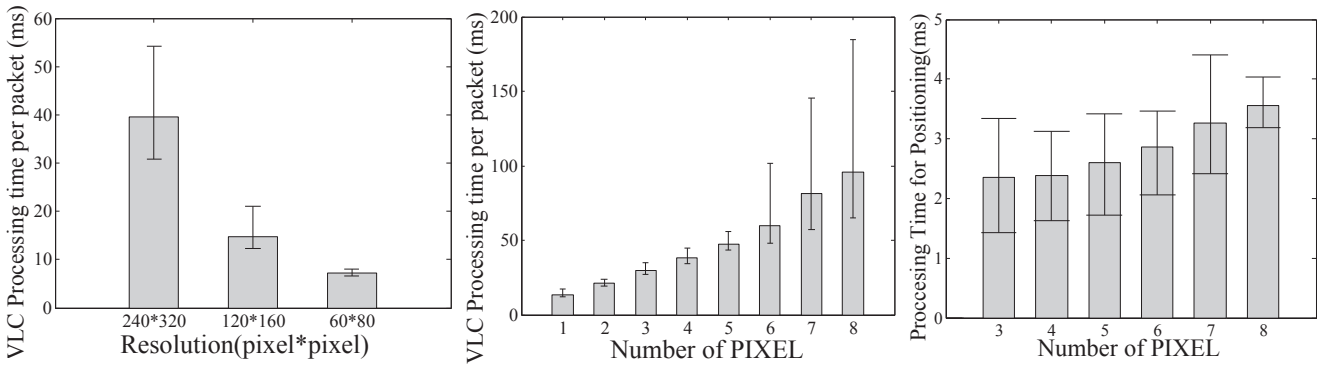
Figure 19(e) summarizes the data from all the experiments and get the relationship between BER and SNR. The theoretical curve for BCSK is calculated according to BPSK. However, since the camera of our smartphone is not stable enough and may occasionally drop frames during the capturing, the bit error rate is slightly higher than the theoretical result. Another reason is the number of samples for statistics is not enough. It can be observed from the two "horizontal lines" in the bottom of the Figure 19(e). Since each logged data only contains hundreds of bits on average, the upper "line" is plotted by cases with 2 error bits and the other "line" is formed by cases with 1 error bit.

Figure 19(f) shows the VLC throughput versus the number of transmitters. The rate is counted by the received payload, which does not contain the preamble and CRC. The result shows that the throughput increases linearly with the number of transmitters. It implies that concurrent transmissions don't affect the performance of our VLC. This is reasonable, since each transmitter is independent and the decoding process for each transmitter is also independent.

5.3 Visible Light Based Positioning

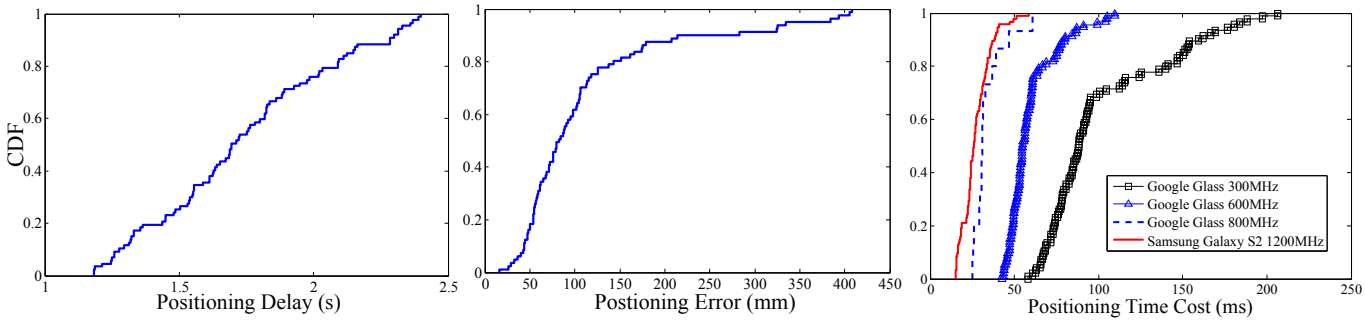
5.3.1 *Computational Cost*

Experiment: We study the computational cost of PIXEL's client application in detail. The evaluation is performed with Samsung Galaxy SII. Its CPU has dual cores with maximum frequency of 1200 MHz. We fix the frequency to 1200 MHz though an app `SetCPU`. It has two cameras and we use the back one with 8 Megapixel for this evaluation. The smartphone is fixed in one location to perform VLP and its distance to the testbed in Figure 16(b) is $3m$. We log the start and end time exclude the sleep period in each part of our program to calculate the processing time. This method may be affected by the CPU scheduling, so we close all the other applica-



(a) Processing time for VLC v.s. resolution (b) Processing time for VLC v.s. number of PIXEL (c) Processing time for localization algorithm v.s. number of PIXEL

Figure 20: Computational cost of each component



(a) Positioning Delay (b) Positioning Error (c) Processing time of VLP

Figure 21: VLP overall performance

tions and unnecessary services in the smartphone to keep the results as accurate as possible.

Results: Figure 20(a) shows the computational cost under different resolutions with single transmitter. Each packet is 17 bits and takes $1200ms$ air time for transmitting, but our decoding algorithm costs less than $50ms$ even when the resolution is 240×320. The result indicates our decoding algorithm performs fast enough. The figure also shows that it takes less time for processing when the downsampled resolution is lower. This is intuitive, because it takes less pixels for processing color extraction.

Figure 20(b) shows the time cost when the client is receiving from multiple transmitters. It shows that the total time of decoding 8 concurrent packets is less than $200ms$, which is still far less than the packet air time $1200ms$. The trend shows the decoding time increases linearly with the number of transmitters. This is because the decoding process for each transmitter is independent and scalable. We also note that the 8 cases share a small portion of basic processing cost, which is actually contributed by the shared downsampling process.

Figure 20(c) shows the time cost of the localization algorithm. Since it requires at least 3 anchors, the number of transmitters starts from 3. The result demonstrates that the localization can be finished within $5ms$ after the client receives enough beacons. The processing time slightly increases with the number of anchors, this is caused by the increased dimension in the object function.

5.3.2 *VLP Performance*

Experiment: We study the overall VLP performance of PIXEL in three aspects: positioning delay, positioning error and positioning overhead. We use Samsung Galaxy SII and Google Glass as client devices while the transmitters in Figure 16(b) serve as VLP anchors. We place client devices to 100 random locations with arbitrary orientations and ensure that at least 3 transmitters can be captured by the camera. The start time and the end time of each test are logged for delay analysis. The localization results and the corresponding ground truth are recorded to estimate localization error. Like previous subsection, we log the processing time of the application to analyze the overhead. Since Google Glass has different frequency settings, we traversal all the configurations in our tests.

Results: Figure 21(a) shows that the average time from the launch of the application to a successful positioning is about $1.8s$. It is reasonable, because the time from the launch of the application to the earliest arrival of the beacon packet is uniformly distributed between 0 and 1200ms. Figure 21(b) shows the localization error is less than 300mm in 90% test cases. The accuracy is in accord with existing results [25]. Figure 21(c) shows the CDF of the computational cost under different CPU settings. As mentioned above, the computational cost includes the VLC decoding and the localization algorithm. Even with a limited processing ability such as 300MHz, our design can finish all the calculation within 150ms on average, which is still far less than the transmission time of one location beacon (1200ms). Therefore, the results validate that PIXEL can provide smart devices with light-weight and accurate localization ability.

6. RELATED WORK

Visible Light Communication. VLC has been studied with different context and design goals. Comparing with radio communication using microwave or mmwave, VLC uses visible light which has much higher frequency and much wider bandwidth. Typical VLC research aims to provide very high link throughput communication [23] for the purpose of wireless access [33]. Normally the receiver requires a photodiode to receive the modulated light. Different from this goal, our design focuses on providing a light-weight

VLC solution for camera-equipped and resource-constrained mobile devices.

In recent literatures [17, 25, 32, 36] it has been proposed to leverage rolling shutter effect of CMOS camera to conduct VLC. It makes use of the fact that LED light with different pulse rates can generate different line patterns in the images captured by camera. However, according to our experience, the rolling shutter based VLC requires high camera resolution and high computational power. In contrast, the VLC in PIXEL allows very low camera resolution and only requires light computation.

Using camera to decode static or dynamic bar-code [20,22,31,37] is a special case of VLC. It has enabled a lot of interesting applications. Comparing to bar-code, PIXEL's VLC transmitter allows longer communication distance and is human impercptible.

Visible Light Based Indoor Positioning. Indoor positioning has been studied for decades [9, 38]. Recent literature [25, 26, 32] shows that receiving location beacons from lighting infrastructure provides a promising means for accurate indoor localization. Comparing to existing designs, PIXEL aims for a more ambitious goal of enabling light-weight visible light positioning that is even afforable in wearables. Comparing to existing LED-based VLC transmitter, PIXEL leverages light polarization to avoid the flickering problem, therefore significantly reduces the requirement on transmitting pulse rate. As a result, the requirement on the VLC receiver is largely reduced that only low resolution camera is needed.

PIXEL shares the same AoA-based positioning algorithm with other camera based solutions [25, 32]. However, we spent a lot of effort to optimize the implementation of the algorithm, therefore make it possible to conduct realtime indoor positioning in wearables.

Polarization Based Modulation. VLC has been widely studied in the context of fiber communication and space communication. In the large body of the literature, some [11, 18, 35] has mentioned polarization based modulation. However, most of them do not contain a system design. Moreover, as far as we know, none of them has addressed the problem caused by the varying angle between transmitter and receiver because usually a fixed angle is assumed.

7. DISCUSSION

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

Bit Rate. In our current implementation, the baud rate of the VLC transmitter is set to 14Hz to allow most camera-equipped devices reliably decode. As a result, the bit rate of our implementation is 14bps. Though it is already sufficient for positioning purpose, the bit rate still can be increased in two ways. First, since the VLC transmitter is not the bottleneck (off-the-shelf LCD screens can achieve 1000Hz [5] refreshing rate), increasing the sampling rate of camera will make higher baud rate possible. For example, if Apple iPhone 6 that contains a 240-fps slow-motion camera is used as the receiver, the baud rate can be increased to 120Hz. Second, if we define more states of the polarizing direction, *i.e.* using higher modulation density, we can encode multiple bits into one symbol thereby increasing the bit rate. However, higher modulation density can make the color-based demodulator much more complicated. Moreover, it may also require calibration for the relation between polarizing direction and the captured vector in RGB space. We leave this study as our future work.

Effect to Illuminating. In our current design, the polarized light is generated by attaching a polarizing film to the illuminating light sources. Although the polarizing film blocks half of the incident light, we found that it does not incur noticeable effect to illuminating because the whole VLC transmitter is small and only covers a small portion of the light source. Alternatively, we can totally avoid the effect to illuminating by leveraging some special light sources (LED [16] or OLED [14]) which directly emit polarized light.

Deployment Issues. Our VLP design has the merit that it is compatible with most of the existing lighting infrastructures because it does not require the light source to be LED. To further facilitate the deployment, we can add a solar-based energy harvesting module into the system to solve the power supply problem. We leave this study as our future work. On the receiver side, the polarizing film can be integrated with the screen protection film, *i.e.* the material around the front camera is made by the polarizing film. Moreover, since the polarizing film is very inexpensive (0.001\$ per camera), it can be used in an one-time use manner, *e.g.* the positioning service provider provides the polarizing film for free.

Beacon Message Size. The beacon message size in our current design is 8-bit. Therefore, we can support up to 256 unique IDs. The message size is similar to existing design [25] and sufficient to cover one floor of normal buildings, *e.g.* shopping malls. In certain cases that require more/less unique IDs, the message size can be increased/decreased. As a consequence, the positioning delay will increase/decrease accordingly.

Viewing Angle. In our current design, we take TN Liquid Crystal as the liquid crystal layer. A limitation of TN LCD is the view angle, *i.e.* when viewing from a very large angle ($|\phi| > 80^\circ$), the light will be blocked by the liquid crystal layer. However, we note that this is not a fundamental problem, because advanced LCD techniques, *e.g. In Phase Switching* (IPS) LCD, has well solved this problem. The same technique can be applied to our design as well.

Power Consumption of Camera. Our VLC receiver relies on camera's video preview, therefore, we need to keep the camera on during the positioning process. A normal camera consumes around 300mW [27] power, which is comparable to the screen [12, 15]. However, since the overall positioning process is less than three seconds, the consumed power by the camera is limited.

Interference to Polarization Based Devices. PIXEL does not interfere with the existing polarization based devices, such as LCD and polarized sunglasses. However, the polarization change of PIXEL's VLC transmitter may be observed through these devices because they also contain polarizer like PIXEL's receiver. As an interesting observation, when an LCD is off, the modulated light by PIXEL's VLC transmitter can be observed when it is reflected by the LCD. However, since the reflected light is weak, we believe it does not bring any big issue. Moreover, when the LCD is on, the polarization change will be totally invisible through reflection. This is because the polarization of the incident light (to LCD) is changed by the crystal layer and eventually be blocked by the second polarizer layer. Finally, people who wearing polarized sunglasses can observe the polarization change of PIXEL's VLC transmitter. However, we do not think it is common that people wear sunglasses in indoor environment.

8. CONCLUSION

This paper presents the system design of PIXEL, which provides a light-weight visible light positioning solution for resource constrained mobile devices. Our idea is to use polarization-based visible light communication which does not have the the light flickering problem as intensity-based visible light communication. In this way, we successfully eliminate the heavy burden on receiving devices caused by high pulse rate. We also enable other illuminating light sources beyond LED lamps to be used for visible light positioning.

To address the design challenges, PIXEL incorporates a noval color-based modulation scheme, an adaptive downsampling algo-

rithm and computational optimization methods to handle the problems of user mobility, uneven camera sampling and limited computational resource. We implemented a prototype of PIXEL, in which the hardware was constructed using commodity components and the software program was built for both Android phones and Google glass. We conducted systematic evaluation based on this prototype. Experiment results validated our idea as well as the system design.

Acknowledgement

We would like to thank our shepherd, Dr. Anthony LaMarca for his help with the final version. We also thank our group members and all the anonymous reviewers for their insightful comments. We are grateful to Su Pan for helping us with LCD and Hang Wu for helping us with the deployment on Google Glass. This work was supported by grants from 973 project 2013CB329006, RGC under the contracts CERG 622613, 16212714, HKUST6/CRF/12R, and M-HKUST609/13, ITS/194/12, and the grant from Huawei-HKUST joint lab.

9. REFERENCES

[1] Android 5.0. https://developer.android.com/about/versions/android-5.0.html, Mar 2015.

[2] Colour rendering of spectra. http://www.fourmilab.ch/documents/specrend/, Mar 2015.

[3] ffmpeg/ffprobe. http://www.ffmpeg.org/ffprobe.html, Mar 2015.

[4] Lcd. http://www.alibaba.com/product-detail/3D-Shutter-glasses-lcd-with-90_1909619852.html?s=p, Mar 2015.

[5] Pi cell. http://www.liquidcrystaltechnologies.com/tech_support/Pi_Cell.htm, Mar 2015.

[6] Polarizing film. http://www.alibaba.com/product-detail/Polarizer-film_458324790.html?s=p, Mar 2015.

[7] Quartz glass. http://www.alibaba.com/product-detail/clear-optical-quartz-plate-glass_846087937.html, Mar 2015.

[8] Spectrum data. https://plot.ly/~arducordermini, Mar 2015.

[9] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

[10] N. Berova, K. Nakanishi, and R. Woody. *Circular dichroism: principles and applications*. John Wiley & Sons, 2000.

[11] R. Calvani, R. Caponi, and F. Cisternino. Polarisation phase-shift keying: a coherent transmission technique with differential heterodyne detection. *Electronics Letters*, 24(10):642–643, 1988.

[12] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, pages 1–14, 2010.

[13] S. Chandrasekhar. Optical rotatory dispersion of crystals. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 259(1299):531–553, 1961.

[14] A. C. Chen, S. W. Culligan, Y. Geng, S. H. Chen, K. P. Klubek, K. M. Vaeth, and C. W. Tang. Organic polarized light-emitting diodes via förster energy transfer using monodisperse conjugated oligomers. *Advanced Materials*, 16(9-10):783–788, 2004.

[15] X. Chen, Y. Chen, Z. Ma, and F. C. Fernandes. How is energy consumed in smartphone display applications? In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, page 3. ACM, 2013.

[16] V. Cimrova, M. Remmers, D. Neher, and G. Wegner. Polarized light emission from leds prepared by the langmuir-blodgett technique. *Advanced Materials*, 8(2):146–149, 1996.

[17] 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.

[18] E. Dietrich, B. Enning, R. Gross, and H. Knupke. Heterodyne transmission of a 560 mbit/s optical signal by means of polarisation shift keying. *Electronics Letters*, 23(8):421–422, 1987.

[19] D. Giustiniano, N. Tippenhauer, and S. Mangold. Low-complexity visible light networking with led-to-led communication. In *Wireless Days (WD), 2012 IFIP*, pages 1–8, Nov 2012.

[20] 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.

[21] G. Horváth and D. Varjú. *Polarized light in animal vision: polarization patterns in nature*. Springer, 2004.

[22] W. Hu, H. Gu, and Q. Pu. Lightsync: Unsynchronized visual communication over screen-camera links. MobiCom '13, pages 15–26, New York, NY, USA, 2013. ACM.

[23] T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using led lights. *Consumer Electronics, IEEE Transactions on*, 50(1):100–107, 2004.

[24] P. Koopman and T. Chakravarty. Cyclic redundancy code (crc) polynomial selection for embedded networks. In *Dependable Systems and Networks, 2004 International Conference on*, pages 145–154. IEEE, 2004.

[25] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. MobiCom '14, pages 447–458, New York, NY, USA, 2014. ACM.

[26] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 331–343, Seattle, WA, Apr. 2014. USENIX Association.

[27] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. Mobisys '13, pages 69–82. ACM, 2013.

[28] M. MacKay. Camera lens and filter adapter assembly, May 4 1993. US Patent 5,208,624.

[29] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[30] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al. *Discrete-time signal processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.

[31] S. D. Perli, N. Ahmed, and D. Katabi. Pixnet: interference-free wireless links using lcd-camera pairs. MobiCom'10, pages 137–148. ACM, 2010.

[32] 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.

[33] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. Led-to-led visible light communication networks. MobiHoc '13, pages 1–10. ACM, 2013.

[34] E. Star. *ENERGY STARÂő Program Requirements for Residential Light Fixtures*. 2010.

[35] X. Tang, Z. Ghassemlooy, S. Rajbhandari, W. Popoola, C. Lee, E. Leitgeb, and V. Ahmadi. Free-space optical communication employing polarization shift keying coherent modulation in atmospheric turbulence channel. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pages 615–620, July 2010.

[36] H.-M. Tsai, H.-M. Lin, and H.-Y. Lee. Demo: Rollinglight - universal camera communications for single led. MobiCom '14, pages 317–320, New York, NY, USA, 2014. ACM.

[37] A. Wang, S. Ma, C. Hu, J. Huai, C. Peng, and G. Shen. Enhancing reliability to boost the throughput over screen-camera links. MobiCom'14, pages 41–52. ACM, 2014.

[38] Z. Yang, X. Feng, and Q. Zhang. Adometer: Push the limit of pedestrian indoor localization through cooperation. *Mobile Computing, IEEE Transactions on*, 13(11):2473–2483, Nov 2014.

[39] P. Yeh and C. Gu. *Optics of Liquid Crystal Displays*. Wiley Publishing, 2nd edition, 2009.