InfoLED: Augmenting LED Indicator Lights for Device Positioning and Communication

Jackie (Junrui) Yang
Stanford University
Stanford, CA, USA
jackiey@stanford.edu

James A. Landay
Stanford University
Stanford, CA, USA
landay@stanford.edu

(a) Diagnose a malfunctioning appliance.

(b) Control a light switch equipped with InfoLED.

(c) Connect a smart button with two light switches to establish control.

Figure 1: InfoLED applications.

ABSTRACT

Augmented Reality (AR) has the potential to expand our capability for interacting with and comprehending our surrounding environment. However, current AR devices treat electronic appliances no different than common non-interactive objects, which substantially limits the functionality of AR. We present InfoLED, a positioning and communication system based on indicator lights that enables appliances to transmit their location, device IDs, and status information to the AR client without changing their visual design. By leveraging human insensitivity to high-frequency brightness flickering, InfoLED transmits all of that information without disturbing the original function as an indicator light. We envision InfoLED being used in three categories of application: malfunctioning device diagnosis, appliances control, and multi-appliance configuration.

We conducted three user studies, measuring the performance of the InfoLED system, the human readability of the patterns and colors displayed on the InfoLED, and users' overall preference for InfoLED. The study results showed that InfoLED can work properly from a distance of up to 7 meters in indoor conditions and it did not interfere with our participants' ability to comprehend the high-level patterns and colors of the indicator light. Overall, study subjects prefer InfoLED to an ArUco 2D barcode-based baseline system and reported less cognitive load when using our system.

Author Keywords

Augmented Reality; Indicator Light; Visible Light Communication; Select and Control; Smartphone; Internet of Things.

INTRODUCTION

Augmented Reality (AR) has the potential to expand our capability for interacting with and comprehending our surrounding environment. For example, Google Maps AR [5] overlays navigation directions onto the user's view of the physical world, and IKEA Place [3] allows the user to visualize virtual furniture in their home. However, current AR devices treat electronic appliances, ranging from microwaves to smart speakers, no different than common non-interactive objects, which substantially limits the functionality of AR. For example, AR has the potential to control any electronic appliance in view, allow the user to see and edit the underlying logic of smart home devices, and provide a rich and intuitive interface for

UIST '19, October 20-23, 2019, New Orleans, LA, USA.

ACM ISBN 978-1-4503-6816-2/19/10 ...$15.00.
http://dx.doi.org/10.1145/3332165.3347954

diagnosing problems with digital appliances. To achieve this, AR needs to have a solution for identifying and positioning those appliances.

Existing communication methods for smart appliances, such as Bluetooth and ZigBee, can tell roughly how close an appliance is to the user, but not exactly where the appliance is in the 3D world. Other positioning solutions, such as ultra-wideband (UWB) [31, 27] or marker-based and vision-based solutions [35, 21], either require a significant appearance or hardware change to the appliances or can't distinguish appliances of the same kind that are near each other. An ideal solution for integrating existing appliances in AR should not disturb the original look of the device. In addition, the solution needs to provide the functionality for positioning the appliances relative to the AR device and transmitting any information necessary for the AR device to establish a connection to the appliances in view.

Towards this end, we propose InfoLED[1], an AR positioning and communication system that leverages the LED indicator lights that already exist on a variety of devices today and overlays high-frequency brightness information on top of the existing signal. By carefully adjusting the amplitude of our high-frequency signal according to the original brightness of the LED, we can allow AR devices to read out information from the appliance without hindering the user's ability to recognize the original information conveyed via the indicator light.

The InfoLED system consists of a smartphone app (InfoLED Scanner) implemented on a commercial-off-the-shelf smartphone equipped with a 240 fps camera (common among recent smartphones) and an embedded controller library that can transform any indicator LED that is connected to a Pulse Width Modulation (PWM) output into a working InfoLED. The InfoLED Scanner can track the relative position of each InfoLED, and record data packets transmitted from the LEDs. This data can then be sent to an application for further processing.

With InfoLED, we envision three types of applications: 1) to diagnose malfunctioning appliances, 2) to control appliances from far away, and 3) to configure the underlying logic so multiple devices can work together. For the first type of application (Figure 1a), we can use InfoLED to transmit extra information, in addition to the original pattern, to show that the appliance is not working properly. Then, the user is able to scan the InfoLED and get step by step instructions on how to fix this specific problem without the need for a manual. The second type of application (Figure 1b) uses the InfoLED to transmit information that is required for establishing communication with smart appliances through other channels (BLE, Wi-Fi, etc.). This allows users to intuitively select the appliances they want and control their states or acquire information from them. The third type of application (Figure 1c) leverages multiple devices that are equipped with InfoLED so that users can easily configure the relationship between devices and assign metadata to them. For example, the user can assign a smart button to control a smart light by creating a virtual linkage between them.

The contributions of this paper include:

1. InfoLED, a data encoding scheme and embedded system library that can encode information in indicator lights without disturbing human recognition of the original patterns and colors.
2. InfoLED Scanner, a smartphone app, based on commercial-off-the-shelf hardware, that can track and read information (60 bps) from multiple InfoLEDs simultaneously, up to 7 meters in an indoor environment.
3. A demonstration of the potential of InfoLED through three applications, and a user study verifying that users prefer the InfoLED system over a baseline 2D barcode solution.

RELATED WORK

InfoLED is related to work in three categories: AR/VR tracking systems, select-and-control methods, and Camera to LED communication methods.

AR/VR tracking systems

The AR/VR tracking systems that are most relevant to this project can be divided into two categories: sensor-based tracking and vision-based tracking [45]. Sensor-based tracking uses different types of sensors (e.g., ultrasonic [9], optical [2], or magnetic field [1]) attached to the appliance that the user wants to track and interact with. These solutions require extra hardware to be installed on the appliance and often require either multiple devices or fixed base stations with known locations to triangulate the appliance's position.

Vision-based tracking systems use an image-based processing method to locate where the device of interest is. They can either depend on a purposefully built marker, such as 2D barcodes [35, 21, 12, 30], retro-reflective or light-emitting points [16], or other patterns [24, 42]. The detection of those markers relies on spatially encoded information, either the shape and layout of black and white blobs in 2D barcodes, spatial layout of dots for retro-reflective or light-emitting points, or image feature points for printed patterns. As a comparison, InfoLED uses information encoded in the time domain, which allows us to decode information even if the indicator light only occupies one pixel in the camera's frames. This design allows us to provide a greater working range without constraining the texture of the devices or the layout of retro-reflective or light-emitting points. Devices can maintain their existing visual design without modification to the hardware.

Researchers have also built systems that use natural features to track the position of a device. One way is to attach a camera onto the device itself [20]. This is commonly used in AR/VR headsets, such as HoloLens [6] and Oculus Quest [7], to track the position of the user's viewport. However, it's hard to use this method to track appliances around the user since it's hard to sync the coordinate system of the appliance with the user's viewport. Another approach is to use a camera that is fixed to the user's viewport to track the natural features of external devices, such as in [13]. While this approach

[1]The code for InfoLED is at https://github.com/infoled

does not require any hardware changes to the device that is being tracked, this solution cannot distinguish devices with similar looks since those devices will share the same natural features. This disadvantage makes these solutions less suitable for tracking mass-produced electronic devices, which look similar but hold different information and functionality that the user may need individual access to (e.g., multiple identical temperature sensors and lights).

Select-and-control methods

Researchers outside of the AR/VR field have also developed systems that ease interacting with electronic devices by simplifying how users select and control them. Beigl et. al. [10] first demonstrated using a laser beam to point at the desired device for selection. Later researchers implemented similar kinds of interaction using different kinds of sensors, such as a Kinect to track user pose and position [18], a projector to indicate the device being controlled [37], combining rough position from GPS and network and image processing to recognize objects [8], or combining rough position, image of the object, and pose data from smartphone sensors [15]. These solutions either require highly customized handheld clients and appliances [10, 37] or need all of the space and positions of the devices to be calibrated beforehand [18, 8, 15] and fixed during the period of usage.

Researchers have also shown ways of selecting devices by physical approximation, such as using RFID [40], personal area networks [25], and electromagnetic signatures [43]. These solutions require the user to spend the extra physical effort to get close to the desired device, as opposed to controlling them from a few meters away.

LED-to-camera communication methods

InfoLED is an LED-to-camera communication method. This is not a new area of research. Danakis et al.'s work [14] first showed the possibility to receive information using a CMOS camera by leveraging the rolling shutter effect. Other researchers have used this method to build a system for establishing a visible light landmark [34, 32] for an indoor positioning system. Many research efforts have also been put into enhancing the robustness of these systems [19], improving the rate of transmission [26], and realizing similar types of communication on networked light bulbs [36]. Researchers have also investigated using visible light communication (VLC) as one of the methods to communicate between smartphones and appliances [41]. A solution has also been proposed to use visible light generated by the flashlight on a smartphone to establish bidirectional communication [17]. These methods also encode information in the time domain by leveraging the rolling shutter effect to readout data from the LED. Thus, they require the light from the LED to be received by a significant area [17] of the CMOS (~600 pixels), which makes these methods only work with a large light source and within a short working distance (less than 1 meter) [32]. Therefore, they are unsuitable for receiving information from a small indicator light, and positioning appliances at a decent working range.

People have also tried to transmit information from LEDs to a camera without leveraging the rolling shutter effect. Cahyadi et al. [11] demonstrated using an array of LEDs to transmit data at a higher bit rate, while Novak et al. [33] explored using a single LED to track an Internet of Things (IoT) device. The former requires an array of LEDs, which is not always possible on commercial devices without significantly changing the appearance. The latter did not demonstrate a working tracking and decoding pipeline. There are also systems based on heavily-modified CMOS sensors with processing units attached directly to them, which achieve high-data-rates and long-range transmission of data [39, 38].

SYSTEM DESIGN

To provide a seamless AR experience with appliances, InfoLED can keep the original look and hardware design of appliances, display a set of human-readable patterns and colors, and transmit data that can be decoded by a smartphone client. To achieve this, we designed: 1) A data encoder that can modulate information onto an indicator light with existing patterns and colors, 2) A software pipeline that can efficiently recognize and track the position of the InfoLED, and 3) A decoding algorithm that can robustly extract information from an LED despite tracking inaccuracies, image noise, and misalignment of the clock between the sender and the receiver. We describe each of these parts of our system in turn.

Encoding information onto the indicator light

We had three design goals for our encoding scheme: 1) support human users in recognizing the original patterns and colors, 2) transmit information that is decodable by a common smartphone, and 3) provide packaging for the transmitted data.

Transmitting extra data in human-readable indicator lights

As suggested by previous work [44], we can leverage either chromatic flickering (quick changes in colors) or luminance flickering (quick changes in brightness) to achieve the goal of transmitting information that is not noticeable to a human being. To maintain compatibility, InfoLED uses luminance flickering since the majority of appliances are only equipped with a single color LED indicator. Therefore, we modulate the data into fast brightness (luminance) changes of the LED while matching the average brightness and color with a human-readable pattern and color.

Designing an encoding scheme for common smartphones

Another design consideration is that our encoding scheme has to stay within the detection limits of a cell phone camera. Nowadays, common phones are equipped with cameras that can achieve 240 frames per seconds (fps)[2], which corresponds to a maximum data rate of 120 bits per second from a single LED. Since we need to maintain the average luminance for human readable patterns, we chose Manchester codes as a modulation scheme that can transmit arbitrary data streams while accommodating any specified average brightness level. Manchester codes can encode raw bitstreams into symbols, and then we can transmit the encoded symbols by brightness levels of the LED.

[2] 8 of 15 top sellers on US Amazon support 240fps video recordings [4].

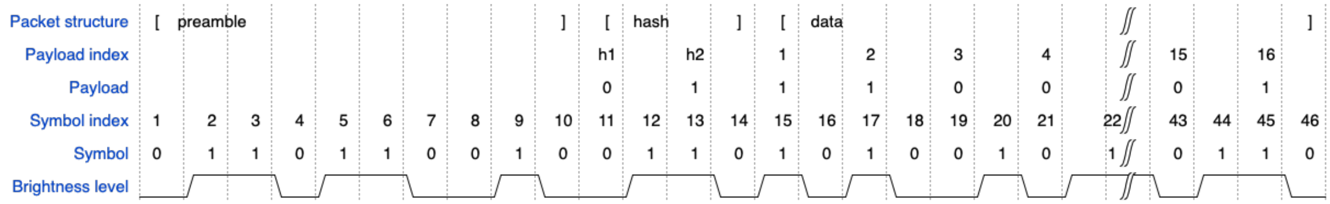


Figure 2: The data packet of the InfoLED
The data packet of the InfoLED is 46 symbols in length and can carry 16 bits of data. It consists of a 10-symbol preamble and a 36-symbol payload. Each payload contains a 2-bit hash and 16 bits of application-specified data which is Manchester encoded to symbols.
This packet structure provides the flexibility for the system to display a user-recognizable pattern and color along with the encoded data.

Manchester codes encode each data bit into two symbols. '0' is encoded as '01', and '1' is encoded as '10'. For example, in Figure 2, the first and the second data bits '11' are encoded as '1010'. InfoLED then uses a low brightness level l to transmit a 0 symbol and a high brightness level h to transmit a 1 symbol. Manchester codes guarantee the number of '0' symbols and '1' symbols will always be the same, so we can produce any given non-zero brightness level b by assigning high and low brightness levels to average b. Also, the Manchester code ensures the number of consecutive symbol '1' or '0' will not exceed two, which allows InfoLED to maintain a flickering frequency between 30 Hz and 60 Hz. This frequency is near the human Critical Flicker Frequency (CFF) for luminance of about 50 Hz [28, 29].

Packaging for transmitted data

Since our camera can only receive a single stream of information from the LED, we also need a preamble that can mark the start of a data packet, and sync the start of each symbol. We want this preamble to 1) be as short as possible, 2) maintain the same number of '1' and '0' symbols (so that we can maintain the average brightness), 3) not have more than 2 consecutive off cycles so we keep the minimum flickering frequency of 30Hz (less visible flickering), and 4) not be a valid sequence in Manchester code so that we don't have to escape the data payload when we have a collision. Therefore we chose the 10-bit preamble "0110110010" that satisfies all of the above requirements.

We designed the data packet format for InfoLED as shown in Figure 2. In each packet, we first transmit a 10-symbol preamble to mark the beginning of the packet. Then, we transmit a 2-bit (4-symbol) Manchester encoded XOR hash of the data payload. Finally, we transmit a Manchester encoded 16-bit (32-symbol) payload that contains the identifier, debug information, and device status. The entire packet takes 46 cycles to transmit, which is 0.38 seconds, and contains 16-bits of data.

Note that packets with a shorter payload take less time to transmit and are less likely to fail. Packets with longer payloads take longer to transmit but waste fewer bits on the preamble and the hash. From Study 1, we measured the ratio of correct packets in an office environment at a distance of 7m to be 0.60. Since each packet is 46 bits, the bit error rate (BER) is 0.01104. The data rate at a certain payload bit length l is $\frac{(1-BER)^{14+l}}{14+l}l$. Therefore, the maximum data rate is reached when the payload bit length is $l = 29$. So we decided to keep the payload bit length as 32 (symbol length 16) since it's the closest power-of-two number to the optimal packet length.

Recognition and tracking of InfoLED

Recognition

Since the InfoLED is flickering in brightness, we leverage the pixel brightness change in consecutive frames captured by the camera to detect the location of the LED. However, due to the movement of the camera, just subtracting two consecutive frames ($\Delta_i = F_i - F_{i-1}$) will also result in the border of all objects being extracted, which produces many false positives (shown in Figure 3). To reliably recognize a potential candidate for the InfoLED, we divide the image into image blocks consisting of 8x8 pixels. For each block, we calculate the sum of the total difference between consecutive frames when different offsets are applied. Then we choose the offset that has the minimum difference within the block, which tells us which direction the objects in the block are moving between those frames. We then construct an RGB range for each pixel from the same pixel in the previous frame and the pixel with the offset in the subsequent frame. This shows the range of possible RGB values, caused by movement, for each pixel, if they keep a constant brightness. We can then tell which pixel has a change in brightness by calculating the RGB distance of each pixel between the RGB value in the current frame and the range we constructed from the previous frame. The resulting frame Δm_i should be an image with pixel values that indicate how much the color or brightness changed between consecutive frames, with compensation for camera motion.

Notably, this algorithm includes a fast and dense method to compute approximate optical flow to compensate for motion. Existing feature tracking algorithms can also be used to reduce the interference caused by motion, but are not nearly fast enough to compute at 240fps on current smartphone hardware.

Figure 3 shows an example of Δ_i and Δm_i. You can see that Δ_i produces undesirable bright edges on objects with high contrast with the background due to the motion of the camera. Δm_i suppresses those edges while keeping the position of the InfoLED visible.

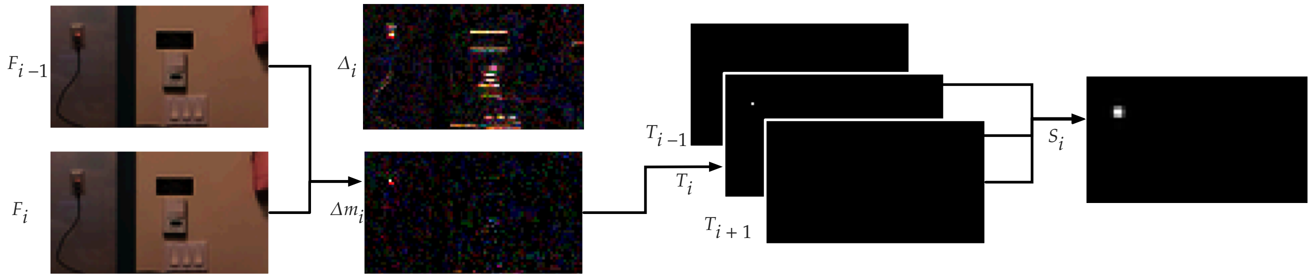


Figure 3: Tracking pipeline of InfoLED
F_i: ith captured frame, Δ_i: Difference between F_i and F_{i-1}, Δm_i: Motion compensated difference between F_i and F_{i-1}, T_i: Δm_i after appling a threshold (The LED is changing its brightness in ith frame, the difference is highlighted in T_i as a white dot. The LED is not changing brightness in $i-1$th and $i+1$th frames, so the result is black), S_i: Sum of a range of T_i (also downsampled to accelerate connected component processing).

However, Δm_i is not enough for the system to track the location of the InfoLED, since the InfoLED will only change its brightness once every few frames. So for each frame, we first apply a threshold τ onto Δm_i producing T_i to only highlight the position of a detected InfoLED. Then we produce S_i by summing a range of frames T_i from a few frames before the current frame and a few frames after: $S_i = \sum_{j=i-range}^{i+range} T_j$. As shown in Figure 3, S_i contains blobs that indicate the area and position of potential InfoLEDs. To extract this information from S_i, we run a connected components algorithm and call the detected connected components "candidates". We pass these "candidates" to the tracking pipeline for further processing.

The current implementation of InfoLED uses 720p video frames at 240fps as input and processes them at 4x decimation. We implemented compute shaders for the pixel processing on the GPU and we implemented the connected component algorithm on the CPU. Since the camera's field-of-view is 34°, the tracking accuracy is approximately 0.19°.

Tracking

Unlike other AR tracking methods, InfoLED uses the temporal resolution of the camera to receive information. Therefore, to decode the information we have to track the position of the LED over time. Since the recognition algorithm may pick up multiple candidates from multiple InfoLEDs and false positives in the same camera view, we need an algorithm that can determine which candidate in one frame corresponds to which in the next frame. Moreover, this algorithm should be able to handle false positives in the candidates and handle tracking of a real InfoLED even after it has been obstructed for a few frames.

We achieve this by maintaining a list of objects called "scope". Each scope represents a potential InfoLED. We keep two counters for each scope, a `historyFrames` counter and a `missingFrames` counter. `historyFrames` represents the number of frame that we have tracked this scope. Longer history means that it is less likely to be noise and should be kept around even if we missed it for a few seconds. `missingFrames` represents the number of consecutive frames in which we haven't found a matching candidate for this scope. We will discard the scope if it cannot be matched with a candidate for a long period of time or it has already been missing longer than it has been seen (possibly noise). Scopes that are missing for a few frames are also likely to reappear from farther away since their movements accumulate over time.

For each scope in every frame, we will try to find a candidate in each frame to update its location. We achieve this by running a matching algorithm for all existing scopes with the candidates detected for each frame. If a scope is not matched with any candidate, the scope will increase the `missingFrames` counter. If a scope is matched with a candidate, the scope will updates its position to that of the candidate, reset the `missingFrames` counter, and increase the `historyFrames` counter. At the end of each frame, we will create a scope for each candidate that is not matched with a scope and delete scopes with `missingFrames` up to a certain threshold or `missingFrames` larger than its `historyFrames`. This design makes scopes corresponding to noise in the image disappear quickly. Also, a scope tracking an actual InfoLED, which has a large `historyFrames`, can maintain the its location for a period of time, so that tracking can continue when the LED is in view again.

The matching algorithm also needs to be carefully designed to accommodate the following requirements: 1) A scope should be matched with a candidate with a close position and a similar size, 2) A scope with a long history should be prioritized in matching, and 3) A scope that has a large `missingFrames` should be able to match with candidates farther away.

We treat this matching problem as a minimum-cost bipartite matching problem (shown in Figure 4). In minimum-cost bipartite matching problems, there are two groups of nodes and a few possible matches between the two groups. Each match comes with a cost. Our goal is to find the solution with the maximum number of matches that has the lowest cost. We define the existing scopes $\{s\}$ as one party and all the candidates $\{c\}$ and a "shadow" copy $\{s\prime\}$ of those scopes as the other party. The reason we need the "shadow" copy is to represent the situation in which the scope is unmatched. As we stated above, an InfoLED may get obstructed for awhile before coming back to the camera view. In that situation, the scope should remain unmatched and wait for that LED to reappear.

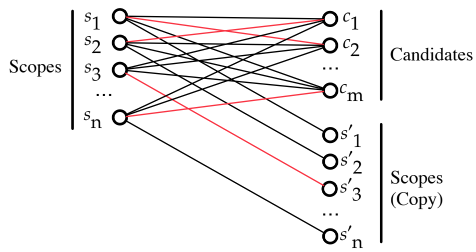


Figure 4: InfoLED scanner tracks InfoLEDs by finding the corresponding candidate for each scope using a minimum-cost bipartite matching algorithm. The red lines represent a possible set of matches.

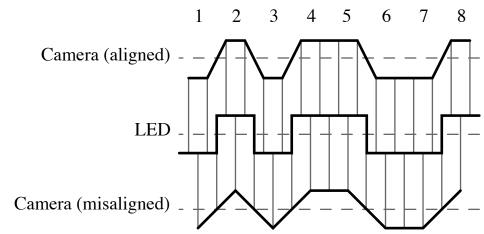


Figure 5: Misalignment between the clock of the camera and the micro-controller may cause decoding error.

Each scope s_i can match with any potential candidate c, and it can also match with its corresponding scope copy $s\prime_i$.

A possible set of matches is shown in Figure 4. s_1, s_2, s_n are matched with c_2, c_1, c_m respectively. s_3 is matched with $s\prime_3$, which means the third scope is not matched with any candidate in this frame.

We then assign the proper costs for each match between the scopes and the candidates, and each match between the scopes and their shadow scopes, as shown in the following equations.

$$weight(s) = \frac{log(2 * \texttt{historyFrames} + 1) + 1}{log(\texttt{missingFrames} + 1) + 1} \tag{1}$$

$$Cost(s,c) = weight(s) \frac{distance(s,c) + |s.\texttt{size} - c.\texttt{size}|/2}{log(\texttt{missingFrames} + 1) + 1} \tag{2}$$

$$Cost(s,s\prime) = weight(s) \times 300 \tag{3}$$

We assign a higher "weight" to a scope that has a long history and less missing frames so that it gets a larger penalty when matched with a bad candidate or assigned as unmatched. For the matching of scopes and candidates, the cost is proportional to the weight and the sum of the position and size differences and the cost is inversely proportional to the number of missing frames. For scopes that are assigned to no candidates, the cost is the product of a constant penalty and the weight of the scope. The constants and non-linear functions in the above equations were determined through trial-and-error.

To validate the effectiveness of our multi-InfoLED tracking algorithm, we performed a test that showed that our algorithm can track 5 InfoLEDs simultaneously while moving up to 33°/s.

Decoding information from an InfoLED

To decode the information from InfoLED, we use a three-step process: 1) we first apply rolling averages to extract the short term brightness level changes, 2) we then decode the transmitted symbols from these brightness level changes, and 3) we produce the transmitted data by unpacking the transmitted packet and dealing with multi-path effects.

Extracting brightness levels

To distinguish between the high and low brightness levels of the LED, we compute the rolling average of a range of frames with the current frame in the center. We then subtract the current pixel from this rolling average and call the result the brightness signal. Since a Manchester code guarantees the same number of '1's and '0's, and the human-readable pattern always runs at a much lower frequency, this signal will produce larger than zero values when the current pixel is transmitting a '1' (higher brightness) and smaller than zero values when the current pixel is transmitting a '0' (lower brightness).

Notably, we compute the rolling average over different frames on pixels in the same positions instead of pixels that are covered by the same scopes. We observe that the brightness of the pixels contained in a scope may be affected by the misalignment between the positions of the InfoLEDs and the pixels of the camera and the minor inaccuracies in the positional tracking. This decreases our signal-to-noise ratio when using pixels with different position according to each scope. Therefore, we compute the rolling averages across the entire frame and subtract this average from the current frame. We then decode the brightness by selecting the pixels covered by each scope at this frame and calculate the sum of pixels in the subtracted frame.

Decoding symbols

Since the clock on the InfoLED is not synced with the one on the InfoLED scanner, we are relying on the relative position of each brightness level change to tell whether this is a single symbol or consecutive symbols with the same brightness level.

One possible solution would be to count the number of consecutive frames with a negative or positive signal. However, this will likely produce an error when the clocks of the sender and the receiver are misaligned. For example, as shown in Figure 5, the LED transmitted a total number of six brightness levels. Let's inspect the second (one cycle of '1') and the fourth transmitted level (two cycles of '1'). When the clocks are aligned (image capture happens when the brightness level is not changing), we can get two samples of high brightness and four samples of high brightness respectively. In this case, we can decode the symbols transmitted just by counting the number of frames with positive signals and negative signals. However, when the clocks are misaligned, we will get one frame of high brightness and 2 frames of near-average brightness for the second transmitted level and three frames of high brightness and two frames of near-average brightness for the fourth transmitted symbol. In this situation, with a little noise, we may receive three frames of above-average brightness, for both one cycle and two cycles of '1' transmitted, which is problematic.

So instead of counting the number of frames with high and low brightness, we estimate the exact time when the brightness level goes across the average level by interpolating consecutive frames and recording it as the end time of the previous level and the start time of the next value. We then determine the number of symbols transmitted by comparing the time duration between the start time and the end time with the duration of one signal cycle (1/120 sec). In this way, the second symbol transmitted should get around 1/120 sec duration and the fourth symbol should get around 2/120 sec duration whether or not the capture clock and the signal clock are aligned.

Unpacking packets

With the symbols decoded, we just need to unpack the packets by stripping out the preamble, decoding the payload (the hash and the data) using the Manchester code, and verifying the data using the hash.

One last problem that we need to deal with in this step is the multi-path effect of the InfoLED, which means that the system can decode information not only from the position of the LED itself but also from the position of reflections or lens glow caused by the indicator light. We tackle this problem by storing a buffer of packets that are received around the same time and only keeping the one with the highest "energy" (the total variation in brightness signal). Since the reflection image is usually weaker in brightness than the original LED, we can keep only the true packet from the correct position.

EVALUATION

We have performed three evaluations of the InfoLED system. The first study measures the performance of the encoding and decoding pipeline with human users in different lighting conditions. The second study verifies that the design of InfoLED won't affect people's ability to gather pattern and color information from InfoLED. The third study evaluates the overall preference of users between InfoLED and a baseline system.

We built the InfoLED appliance using a Particle Photon microcontroller and the InfoLED scanner on an iPhone X.

Study 1: Performance of InfoLED

To show InfoLED is a reliable way of conveying information and tracking appliances, we wanted to measure how distance, ambient light, and human motion affect the performance of InfoLED. In study 1, we asked each participant to use a scanner app on the smartphone to scan an InfoLED for 10 consecutive seconds and see what the bit error rate was at different distances and under different ambient light conditions.

[3] The dark condition has slightly worse results than the office condition. We think this was due to the phone's automatic exposure not working well in the dark.

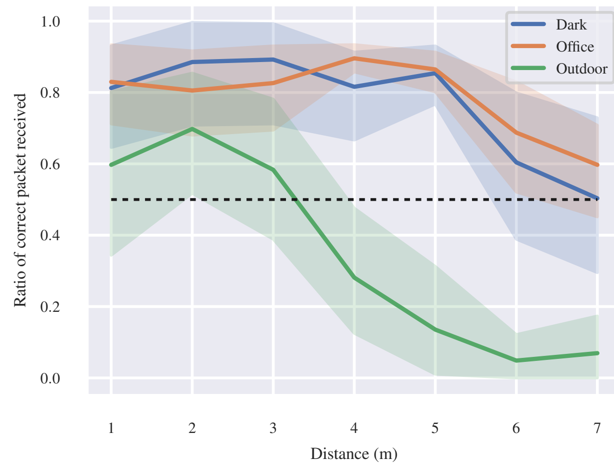


Figure 6: Ratio of packets received at different distances and in different lighting conditions. Dotted line shows the 50% correct packet received line. Error band shows 95% confidence interval.
Statistically significant results ($p < 0.05$ in proportion z test) were found up to 6m (dark), 7m (office), and 3m (outdoor).[3]

Participants

We recruited 12 participants (6 female, 1 non-binary), aged between 20 to 36 (median 25). Each participant was compensated $10 for their time.

Tasks

We asked each user to scan the InfoLED for 10 seconds at 7 different distances (1-7m) and with different lighting conditions (outdoor, office, and dark) in a counter-balanced order. For the outdoor condition, we used the real daylight at our experiment location, which measured around 10,000 nits. For the office condition, we controlled the ambient lighting to approximately 400 nits, while in the dark condition, we kept the ambient lighting at less than 10 nits.

Results

The results are shown in Figure 6. The x-axis shows different distances of the scan, and the y-axis shows the ratio of correct packets received at a given distance and lighting condition. The bands on the y-axis show the confidence interval of the distribution of the ratio of packets received. Since InfoLED is designed to transmit identification and status of the appliances, and each InfoLED packet only takes around 1/3 second to transmit, a ratio of correctly receiving 0.5 packet every packet transmitted is acceptable (shown in Figure 6 as the dotted line). At this ratio, the InfoLED system should be able to identify the position of appliances and receive status updates from them within 0.7 seconds on average. We found that in an indoor environment (office and dark), our system can work acceptably up to 7 meters. In an outdoor environment, it can work acceptably up to 3 meters.

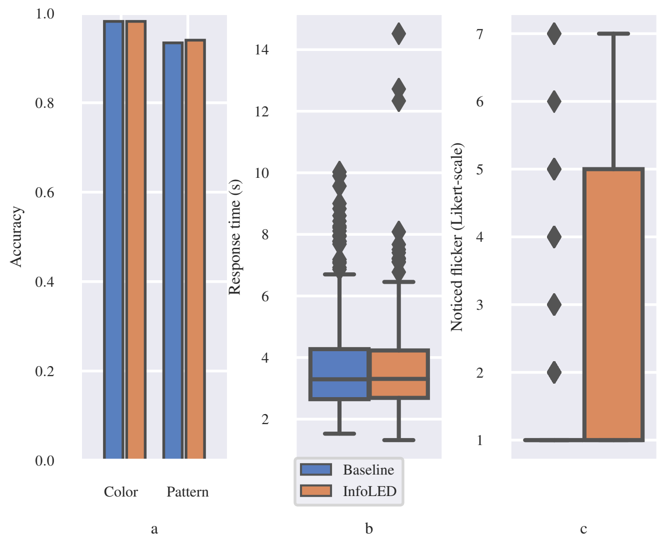


Figure 7: Study 2 shows that InfoLED does not have a significant effect on a user's accuracy and speed in recognizing the human-readable colors and patterns. Users are significantly more likely to notice flickering in the InfoLED condition, but in most cases, they still won't notice.

Study 2: Human readability of InfoLED

We designed the InfoLED to carry extra information for AR devices along with the original information intended for human eyes. We wanted to verify that humans can correctly recognize the original information carried by the indicator light with the extra modulation required by InfoLED. In this study, we asked users to identify the different patterns and colors displayed by an indicator light with or without information carried by InfoLED, and compared the difference between the success rates and the time for the participants to respond to the prompt.

Participants

We recruited 12 participants (5 female), aged between 20 to 36 (median 26.5). Each participant was compensated $10 for their participation. The user pool partially overlapped with those in study 1.

Tasks

We designed 16 different tasks that consisted of 4 common patterns displayed on indicator lights [22] (constant, flashing quickly, flashing slowly, and breathing/pulsing) and 4 different colors (red, green, blue, and white). In each task, we asked the user to tell us what the pattern and color that they recognized is. We also asked participants about whether they see "bright flashes" or "flickering" apart from the shown pattern in a Likert scale from 1 (strongly disagree) to 7 (strongly agree). We also recorded the amount of time that it took for users to respond to each question. We asked the user to accomplish these tasks in a random order in two different lighting conditions (dark and office lighting).

Results

We computed the average correct recognition rate for patterns and colors in both conditions (shown in Figure 7a). For colors, we get 0.982 and 0.982 for with and without the InfoLED condition, respectively. For patterns, we get 0.940 and 0.934 for with and without the InfoLED condition, respectively. Fisher exacts cannot find significant differences between the two conditions ($p = 1.00$ for colors and $p = 0.88$ for patterns).

As shown in Figure 7b, the time it takes for users to recognize the patterns and colors in the InfoLED condition averaged 3.63s (standard deviation $\sigma = 1.49$), and for the condition with normal indicator lights, it was 3.65s (standard deviation $\sigma = 1.49$). We performed a t-test and cannot find a statistically significant difference between the two groups ($p = 0.82$).

For the Likert-scale question about "bright flashes" and "flickering", both conditions have a median of 1 which suggests that users strongly disagree that they see flickering in both conditions. However, they are more likely to notice flickering with InfoLED (shown in Figure 7c). A Mann-Whitney U test suggests a significant difference between the two groups ($p = 2.56 * 10^{-20}$).

In conclusion, when recognizing the pattern and color the results do not show any difference between with and without InfoLED for the user's accuracy and time. The user can sometimes tell whether an indicator LED is transmitting extra information, but in most cases, they have not noticed any bright flashes or flickering (median is 1).

Study 3: Overall preference between InfoLED and AR-Tag

Finally, we wanted to evaluate users' general preference and experience in using the InfoLED compared with the state-of-the-art systems. We built two InfoLED enabled devices, a smart light switch and a smart button, and asked users to try to debug, control, and configure them in an AR environment.

Tasks

We asked the users to perform three tasks for both the InfoLED and the baseline conditions.

The first task was to diagnose and solve a problem with the smart button. We designed two possible failure cases for the smart button: network connection lost and low battery. Since the InfoLED can carry this information with the embedded indicator light, in the experimental condition the user is asked to use the app to check what's wrong with the device and follow the instructions embedded in the app to fix it. In the control condition, we asked the user to read a short paper-based manual that listed all of the possible indicator light patterns and colors, what problem each corresponds to, and how to fix each problem. The user then followed the instructions in the manual to fix the problem.

The second task was to control the light switch using an AR interface in an experimental app. In the InfoLED condition, we asked the user to use our experimental app built using our InfoLED tracking pipeline. They first pointed the camera of the smartphone towards the switch, waited for the icon to appear, and tapped on the icon to turn the light switch on and off. In the control condition, we attached a 15mmx15mm ArUco marker [35, 21] to the switch and asked the user to do the same thing with another app that is built with a similar interface but uses the ArUco tracking pipeline.

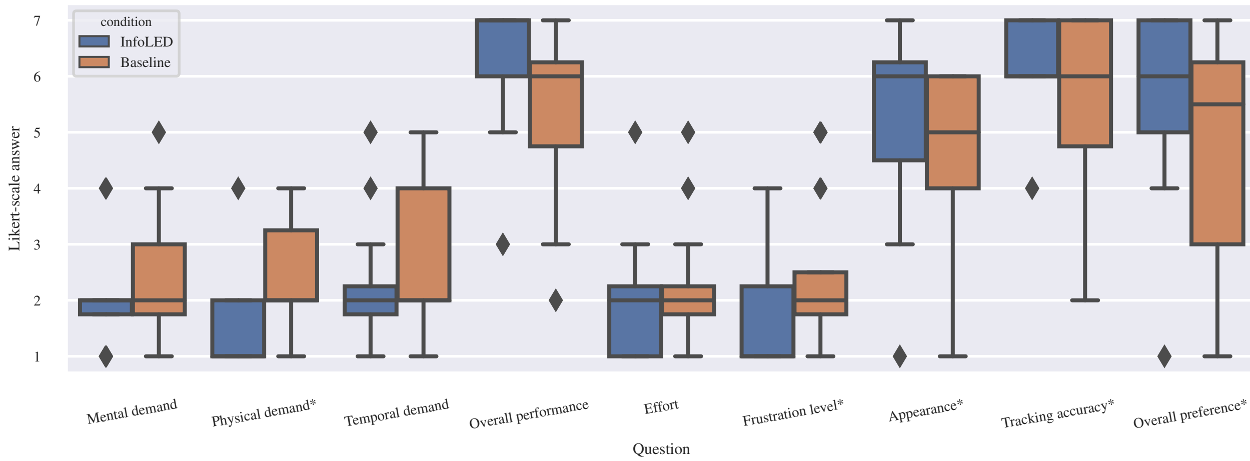


Figure 8: Questionnaire result for study 3. *: statistically significant (p<0.05).

The third task was to configure the smart button so that the physical button can control the light switch without needing to use the app. In the InfoLED condition, we asked the user to first point the smartphone camera to the switch to register that device in the system, point the camera to the button to do the same, and then draw a connection between the button to the switch in the app to establish a connection between them. In the control condition, we asked the user to do the same thing with an ArUco marker attached to both devices.

Each user tried both conditions (control and InfoLED) in a counter-balanced order. We also ensured that the problem with the smart button that they encountered was different within subject and the combination of problems and condition orders was counterbalanced between subjects. After they tried each condition, we asked them to fill in a questionnaire consisting of 6 questions from the NASA-TLX [23] and 3 questions we designed about the appearance of the appliances, tracking accuracy, and their overall preference[4]. In both conditions, the participant was approximately 3 meters away from the smart switch and 1 meter away from the smart button.

Participants

We recruited 12 participants (8 female), aged between 19 to 51 (median 28). Each participant was compensated $10 for their participation. We made sure that all participants hadn't participated in the first two studies.

Results

The result of the questionnaire are shown in Figure 8. Wilcoxon Signed-rank tests show that InfoLED has less physical demand ($p = 0.02$) and frustration level ($p = 0.03$) than the baseline. It also suggests that appliances equipped with InfoLED have a better appearance than those with the ArUco marker, InfoLED also appears to have better tracking accuracy, and users preferred InfoLED over the baseline condition.

Task completion time is shown in Figure 9. Welch's t-tests show that users in the InfoLED condition take less time to complete Task 2 ($p = 1.2 * 10^{-5}$) and Task 3 ($p = 0.0063$) than in the baseline. Although users completed Task 1 faster in the baseline condition (not statistically significant $p = 0.19$), we observed that the increase in time for InfoLED in Task 1 is due to users spending more time watching an optional instructional video on how to fix the problem. We think the reduction in time in Task 2 and Task 3 was due to InfoLED working better at long range and in low light conditions. Users have less trouble when using InfoLED to control the switch and InfoLED won't lose track of the appliance after users turn the lights off.

[4]See our questionnaire in supplementary materials

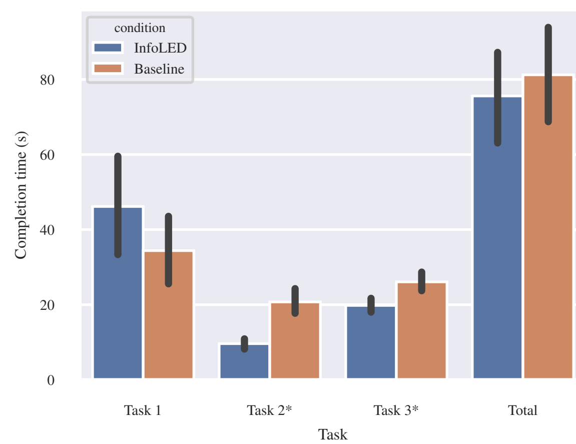
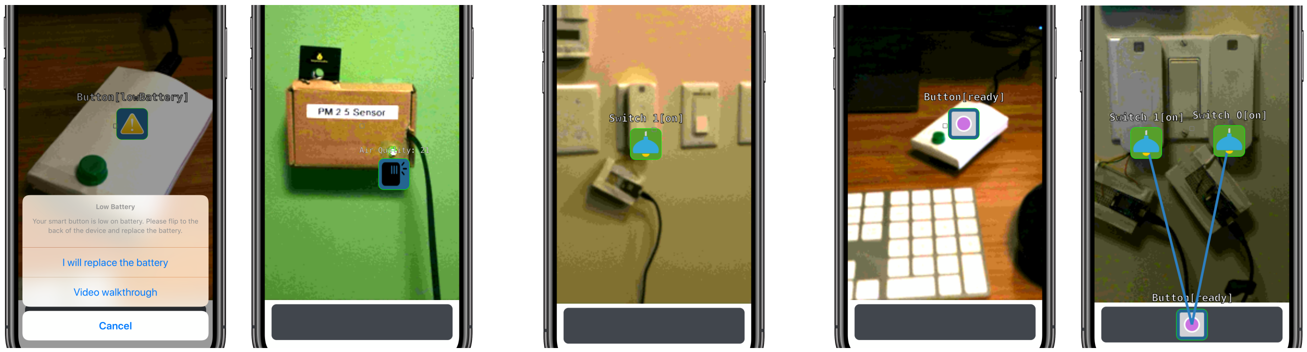


Figure 9: Task completion time in the InfoLED and baseline conditions. *:statistically significant differences between conditions

We collected subjective feedback during and after the user study: Users said that InfoLED seems more responsive (P5) and fluid (P12). They thought that the interaction in the InfoLED condition is self-explanatory (P5). Users had mixed feelings about the walkthrough video on how to change the



(a) Reading information from appliances (b) Controlling appliances (c) configuring appliances

Figure 10: Applications of InfoLED

battery. P6, P7, and P11 thought the walkthrough video was helpful while P4 thought it was not as good as the written instructions. P4 said that he/she was surprised that InfoLED can work from that far away. 5 out of 12 users commented that in the baseline condition they needed to stand up and get closer to the switch to toggle it, especially when the light was turned off. P6 told us that the AR sticker is "ugly" and does not provide more accurate results. Overall, P2 thought using AR to interact with appliances is easy and P11 thought it's "kind of fun". Users also suggested a few future applications, including controlling kitchen appliances (P2), locking and unlocking doors (P2, P10), and using the system as an accessibility tool (P2, P9).

DISCUSSION

In this section, we discuss the applications and limitations of the InfoLED system.

Applications

We have built demonstrations of three categories of applications for the InfoLED system: 1) diagnosing and reading information from appliances, 2) controlling appliances, and 3) configuring appliances.

Diagnosing and reading information from appliances

InfoLED can transmit 16-bits of information for the AR system to identify the position and status of an appliance. This can be used to transmit essential device information, such as diagnostic information, device status, and sensor data.

We built three devices leveraging this feature. The smart button uses two bits in the InfoLED data packet to indicate diagnostic information about its current status (lost internet connection, low battery, or fully operational). This information allows the user to find out and solve the problem with their appliances in a fluid interface. As shown in Figure 10a, by scanning the InfoLED, the system recognizes what's wrong with the device, informs the user about the solution, and even shows a link to a walkthrough video. The smart air quality sensor (Figure 10a) transmits the current air quality in 256 levels by using 8-bits in the InfoLED packet, so that the user can know the rough air quality through the color of the indicator light (green is good, red is bad), and if they want detailed data, they can view the actual air quality using InfoLED scanner. The smart switch in Figure 10b also uses one bit in the InfoLED data packet to indicate whether it's on or off.

Controlling appliances

Although InfoLED only supports unidirectional communication, when implemented on a connected appliance, InfoLED can be used to indicate the ID of the appliance and help the AR device establish a bi-directional channel with the appliance.

In the case of the smart switch (Figure 10b), it can be controlled via the network if the user knows the ID of the appliance and selects it from a list. With InfoLED, the user can just point the smartphone at the switch. InfoLED will recognize the position and the ID of the switch, establish a connection with the device with that ID on the network, and show an icon on the InfoLED Scanner app so the user can interact with the switch directly.

Configuring appliances

The third application of the InfoLED is that it can be used to configure multiple appliances equipped with InfoLED and configure the relationship between them. If the user owns multiple devices that support the InfoLED, the system can show all devices in an AR interface to ease their configuration.

As shown in Figure 10c, InfoLED can be used to establish a connection between an input and output device. For example, the user can first scan a smart button to let the system know a button is nearby, then the user can drag a link from the button to the other appliance to establish a connection. The figure shows two smart switches linked with the button so that the button can turn both of switches on and off at the same time.

Limitations

One limitation of InfoLED is the data rate. Currently, InfoLED can transmit at a data rate of 60-bits per second. Although this is not fast compared to other wireless communication methods, such as Bluetooth or Wi-Fi, it should be enough for just transmitting the ID and the status of the device. InfoLED can be used to distinguish up to 65,536 devices, compared to a state-of-the-art AR solution, such as ArUco [35, 21] that can typically distinguish between 4,000 devices. Combined with the method used in Snap-to-it [15], we can support even more devices by leveraging other sensor data from a smartphone, such as the location and orientation of the user, to distinguish between multiple devices that display the same ID but are located in different buildings or facing different directions. Also, some smartphones (e.g., the Sony Xperia 1 and the Samsung Galaxy S9) have been equipped with a camera with even higher frame rates. This can also be used to increase the bit rate of the InfoLED system.

Another limitation is that when the LED is turned off, we can no longer track the position of the device. In the future, we can potentially implement activating the system using other communication channels. For example, we can send a Bluetooth or network broadcast to nearby devices that are equipped with InfoLED to turn the LED on temporally so that InfoLED can track and display information for them.

Conclusions

In this paper, we present InfoLED, an AR positioning and communication system that leverages the existing indicator lights on appliances. We designed a specialized encoding method and processing pipeline so that we can encode extra information without disturbing the original human-readable patterns and colors on indicator lights or changing the appearance of the appliances while allowing positioning from up to 7 meters indoors. We evaluated the system in three user studies and discovered that InfoLED can shorten the user's task completion time and impose less physical demand and frustration on the user. It also has a better appearance and overall performance compared to 2D barcodes. With InfoLED, future smart appliances can be easier to interact with and future augmented reality applications can be even more exciting with the ability to interact with other electronics in the physical environment.

Acknowledgments

The authors would like to acknowledge Tianshi Li for her help on the early prototypes and writing. In addition, the authors would like to thank Xieyang Xu, Parastoo Abtahi, and Dae Hyun Kim for feedback on system design, user study, and writing. Finally, the authors would also like to thank the reviewers for their constructive feedback.

REFERENCES

[1] 2007. Razer Hydra | Official Razer Support. https://support.razer.com/console/razer-hydra/. (2007). (Internet Archive) https://web.archive.org/web/20190703215016/https://support.razer.com/console/razer-hydra/.

[2] 2016. VIVE™ | Discover Virtual Reality Beyond Imagination. https://www.vive.com/us/. (2016). (Internet Archive) https://web.archive.org/web/20190701114334/https://www.vive.com/us/.

[3] 2017. IKEA Place augmented reality app. https://highlights.ikea.com/2017/ikea-place/. (2017). (Internet Archive) https://web.archive.org/web/20190622121032/https://highlights.ikea.com/2017/ikea-place/.

[4] 2019. Amazon Best Sellers: Best Cell Phones. https://www.amazon.com/Best-Sellers-Cell-Phones-Accessories/zgbs/wireless/7072561011. (2019). (Internet Archive) https://web.archive.org/web/20190703211521/https://www.amazon.com/Best-Sellers-Cell-Phones-Accessories/zgbs/wireless/7072561011.

[5] 2019. Google is letting some users test its AR navigation feature for Google Maps - The Verge. https://www.theverge.com/2019/2/10/18219325/google-maps-augmented-reality-ar-feature-app-prototype-test. (2019). (Internet Archive) https://web.archive.org/web/20190606135557/https://www.theverge.com/2019/2/10/18219325/google-maps-augmented-reality-ar-feature-app-prototype-test.

[6] 2019. Microsoft HoloLens | Mixed Reality Technology for Business. https://www.microsoft.com/en-US/hololens. (2019). (Internet Archive) https://web.archive.org/web/20190630040958/https://www.microsoft.com/en-us/hololens/.

[7] 2019. Oculus Quest | Oculus. https://www.oculus.com/quest/. (2019). (Internet Archive) https://web.archive.org/web/20190703215222/https://www.oculus.com/quest/.

[8] Mark Adler and Franklin Reynolds. 2007. Vision-Guided "Point and Click" for Smart Rooms. In *2007 Second International Conference on Systems and Networks Communications (ICSNC 2007)*. IEEE. DOI: http://dx.doi.org/10.1109/icsnc.2007.84

[9] Andreas M. Ali, Kung Yao, Travis C. Collier, Charles E. Taylor, Daniel T. Blumstein, and Lewis Girod. 2007. Acoustic source localization using the acoustic ENSBox. In *Proceedings of the 6th international conference on Information processing in sensor networks - IPSN '07*. ACM Press. DOI: http://dx.doi.org/10.1145/1236360.1236437

[10] Michael Beigl. 1999. Point & Click-Interaction in Smart Environments. In *Handheld and Ubiquitous Computing*. Springer Berlin Heidelberg, 311–313. DOI: http://dx.doi.org/10.1007/3-540-48157-5_31

[11] Willy Anugrah Cahyadi, Yong Hyeon Kim, Yeon Ho Chung, and Chang-Jun Ahn. 2016. Mobile Phone Camera-Based Indoor Visible Light Communications With Rotation Compensation. *IEEE Photonics Journal* 8, 2 (apr 2016), 1–8. DOI: http://dx.doi.org/10.1109/jphot.2016.2545643

[12] C. Celozzi, G. Paravati, A. Sanna, and F. Lamberti. 2010. A 6-DOF ARTag-based tracking system. In *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*. IEEE. DOI: http://dx.doi.org/10.1109/icce.2010.5418706

[13] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. 2006. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (jul 2006), 615–628. DOI: http://dx.doi.org/10.1109/tvcg.2006.78

[14] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. 2012. Using a CMOS camera sensor for visible light communication. In *2012 IEEE Globecom Workshops*. IEEE. DOI: http://dx.doi.org/10.1109/glocomw.2012.6477759

[15] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. ACM Press. DOI: http://dx.doi.org/10.1145/2858036.2858177

[16] Klaus DorfmÜller. 1999. Robust tracking for augmented reality using retroreflective markers. *Computers & Graphics* 23, 6 (dec 1999), 795–800. DOI: http://dx.doi.org/10.1016/s0097-8493(99)00105-3

[17] Alexis Duquel, Razvan Stanica, Herve Rivano, and Adrien Desportes. 2018. Decoding methods in LED-to-smartphone bidirectional communication for the IoT. In *2018 Global LIFI Congress (GLC)*. IEEE. DOI: http://dx.doi.org/10.23919/glc.2018.8319118

[18] Ana Fernandez, Luca Bergesio, Ana M. Bernardos, Juan A. Besada, and Jose R. Casar. 2015. A Kinect-based system to enable interaction by pointing in smart spaces. In *2015 IEEE Sensors Applications Symposium (SAS)*. IEEE. DOI: http://dx.doi.org/10.1109/sas.2015.7133613

[19] Julia Ferrandiz-Lahuerta, Daniel Camps-Mur, and Josep Paradells-Aspas. 2014. A Reliable Asynchronous Protocol for VLC Communications Based on the Rolling Shutter Effect. In *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE. DOI: http://dx.doi.org/10.1109/glocom.2014.7417229

[20] Frank J. Ferrin. 1991. Survey of helmet tracking technologies. In *Large Screen Projection, Avionic, and Helmet-Mounted Displays*, Harry M. Assenheim, Richard A. Flasck, Thomas M. Lippert, and Jerry Bentz (Eds.). SPIE. DOI: http://dx.doi.org/10.1117/12.45422

[21] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (jun 2014), 2280–2292. DOI: http://dx.doi.org/10.1016/j.patcog.2014.01.005

[22] Chris Harrison, John Horstman, Gary Hsieh, and Scott Hudson. 2012. Unlocking the expressivity of point lights. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. ACM Press. DOI: http://dx.doi.org/10.1145/2207676.2208296

[23] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (Oct 2006), 904–908. DOI: http://dx.doi.org/10.1177/154193120605000909

[24] Valentin Heun, Shunichi Kasahara, and Pattie Maes. 2013. Smarter objects. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*. ACM Press. DOI: http://dx.doi.org/10.1145/2468356.2468528

[25] Todd D. Hodes, Randy H. Katz, Edouard Servan-Schreiber, and Lawrence Rowe. 1997. Composable ad-hoc mobile services for universal interaction. In *Proceedings of the 3rd annual ACM/IEEE international Conference on Mobile Computing and Networking - MobiCom '97*. ACM Press. DOI: http://dx.doi.org/10.1145/262116.262121

[26] Pengfei Hu, Parth H. Pathak, Xiaotao Feng, Hao Fu, and Prasant Mohapatra. 2015. ColorBars. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies - CoNEXT '15*. ACM Press. DOI: http://dx.doi.org/10.1145/2716281.2836097

[27] Ke Huo, Tianyi Wang, Luis Paredes, Ana M. Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018. SynchronizAR. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI: http://dx.doi.org/10.1145/3242587.3242595

[28] Yi Jiang, Ke Zhou, and Sheng He. 2007. Human visual cortex responds to invisible chromatic flicker. *Nature Neuroscience* 10, 5 (apr 2007), 657–662. DOI: http://dx.doi.org/10.1038/nn1879

[29] Ravi Karkar, Rafal Kocielnik, Xiaoyi Zhang, Paul G. Allen, Jasmine Zia, George N. Ioannou, Sean A. Munson, and James Fogarty. 2018. Beacon. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (Sep 2018), 1–27. DOI: http://dx.doi.org/10.1145/3264927

[30] Fahim Kawsar, Enrico Rukzio, and Gerd Kortuem. 2010. An explorative comparison of magic lens and personal projection for interacting with smart objects. *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10* (2010). DOI: http://dx.doi.org/10.1145/1851600.1851627

[31] Josef Kulmer, Stefan Hinteregger, Bernhard Grosswindhager, Michael Rath, Mustafa S. Bakr, Erik Leitinger, and Klaus Witrisal. 2017. Using DecaWave UWB transceivers for high-accuracy multipath-assisted indoor positioning. *2017 IEEE International Conference on Communications Workshops (ICC Workshops)* (May 2017). DOI: http://dx.doi.org/10.1109/iccw.2017.7962828

[32] Zening Li. 2016. *Indoor Positioning System Using Visible Light Communication and Smartphone with Rolling Shutter Camera*. Ph.D. Dissertation. Department of Electrical and Computer Engineering, UC Riverside. https://escholarship.org/uc/item/4280x06f

[33] Marek Novak, Otakar Wilfert, and Tomas Simicek. 2017. Visible light communication beacon system for internet of things. In *2017 Conference on Microwave Techniques (COMITE)*. IEEE. DOI: http://dx.doi.org/10.1109/comite.2017.7932304

[34] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. 2014. Visual light landmarks for mobile devices. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE. DOI: http://dx.doi.org/10.1109/ipsn.2014.6846757

[35] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. 2018. Speeded up detection of squared fiducial markers. *Image and Vision Computing* 76 (aug 2018), 38–47. DOI: http://dx.doi.org/10.1016/j.imavis.2018.05.004

[36] Stefan Schmid, Thomas Richner, Stefan Mangold, and Thomas R. Gross. 2016. EnLighting: An Indoor Visible Light Communication System Based on Networked Light Bulbs. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. DOI: http://dx.doi.org/10.1109/sahcn.2016.7732989

[37] Dominik Schmidt, David Molyneaux, and Xiang Cao. 2012. PICOntrol. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. ACM Press. DOI: http://dx.doi.org/10.1145/2380116.2380166

[38] Isamu Takai, Tomohisa Harada, Michinori Andoh, Keita Yasutomi, Keiichiro Kagawa, and Shoji Kawahito. 2014. Optical Vehicle-to-Vehicle Communication System Using LED Transmitter and Camera Receiver. *IEEE Photonics Journal* 6, 5 (oct 2014), 1–14. DOI: http://dx.doi.org/10.1109/jphot.2014.2352620

[39] Isamu Takai, Shinya Ito, Keita Yasutomi, Keiichiro Kagawa, Michinori Andoh, and Shoji Kawahito. 2013. LED and CMOS Image Sensor Based Optical Wireless Communication System for Automotive Applications. *IEEE Photonics Journal* 5, 5 (oct 2013), 6801418–6801418. DOI: http://dx.doi.org/10.1109/jphot.2013.2277881

[40] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. 1999. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '99*. ACM Press. DOI: http://dx.doi.org/10.1145/302979.303111

[41] Hanno Wirtz, Jan Ruth, Martin Serror, Torsten Zimmermann, and Klaus Wehrle. 2015. Enabling ubiquitous interaction with smart things. *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)* (Jun 2015). DOI: http://dx.doi.org/10.1109/sahcn.2015.7338324

[42] Robert Xiao, Chris Harrison, Karl D.D. Willis, Ivan Poupyrev, and Scott E. Hudson. 2013. Lumitrack. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*. ACM Press. DOI: http://dx.doi.org/10.1145/2501988.2502022

[43] Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Deus EM Machina. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press. DOI: http://dx.doi.org/10.1145/3025453.3025828

[44] Lan Zhang, Cheng Bo, Jiahui Hou, Xiang-Yang Li, Yu Wang, Kebin Liu, and Yunhao Liu. 2015. Kaleido. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*. ACM Press. DOI: http://dx.doi.org/10.1145/2789168.2790106

[45] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. 2008. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE. DOI: http://dx.doi.org/10.1109/ismar.2008.4637362