# HiLight: Hiding Bits in Pixel Translucency Changes

Tianxing Li, Chuankai An, Andrew T. Campbell, and Xia Zhou
Department of Computer Science, Dartmouth College
{tianxing, chuankai, campbell, xia}@cs.dartmouth.edu

## ABSTRACT

We present HiLight, a new form of screen-camera communication without the need of any coded images (*e.g.* barcodes) for off-the-shelf smart devices. HiLight hides information underlying any images shown on a LED or an OLED screen, so that camera-equipped smart devices can fetch the information by turning their cameras to the screen. HiLight achieves this by leveraging the orthogonal transparency (alpha) channel, a well-known concept in computer graphics, to embed bits into pixel translucency changes without the need of modifying pixel color (RGB) values. We demonstrated HiLight's feasibility using smartphones. By offering an unobtrusive, flexible, and lightweight communication channel between screens and cameras, HiLight opens up opportunities for new HCI and context-aware applications to emerge, *e.g.* smart glass communicates with screens for additional personalized information to realize augmented reality.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

## Keywords

Screen-camera link; Visible light communication

## 1. INTRODUCTION

Screens (*e.g.* LED/OLED screens) and cameras are ubiquitous, not only in laptops, but also in today's smart devices such as smartphones, smart watches, and smart glass. Communication between screens and cameras has been attracting growing interest, where information is encoded into visual frames on the screen and cameras-equipped devices extract the information from the captured frames. Operating on the visible light spectrum, screen-camera links are highly directional and interference-free, offering a promising out-of-band communication approach for short-range information acquiring. The most popular example of screen-camera links is QR-Code [3], which embeds information (typically a URL) into 2D barcodes. Recent research endeavors lead to new modulation scheme to boost data rates [14], and innovative barcodes [8, 12, 20].

While exciting, most existing screen-camera proposals rely on specialized coded images on the screen to transmit data. These coded images (*e.g.* barcodes) are typically visible to users. They either occupy the whole screen, or interfere with the content currently shown on the screen, creating an unpleasant viewing experience. Recent designs aim to improve the user's viewing experience by designing unobtrusive barcodes [20], or integrating images/watermark into barcodes [12, 22]. However, by modifying pixel colors of the full image, these designs entail high complexity to transmit dynamic information in real time. Furthermore, existing designs require *absolute* color values or image features (*e.g.* edges) in the perceived frame to decode information, and hence are susceptible to problems of image blur and frame synchronization [10].

To overcome the above limitations, we propose HiLight, a flexible and robust form of screen-camera communication channel that hides information underlying any image currently shown on the screen. The key idea of HiLight is to embed information into pixel translucency changes on the LED or OLED screen, so that camera-equipped devices can decode the information based on the perceived color intensity changes. HiLight utilizes an orthogonal transparency channel (alpha channel), a well-known concept in computer graphics, to change pixel translucency and opacity. The level of pixel translucency change is configured so that it is perceivable to only cameras but not human eyes. Furthermore, by controlling the translucency changes of each pixel independently, HiLight enables multiple transmitter elements on the screen to transmit data simultaneously. This creates a MIMO communication channel between screen and cameras [4], which can be used to boost data rate or to improve transmission reliability.

Operating on a separate transparency (alpha) channel, HiLight provides three key benefits. *First*, without adding any specialized coded images, HiLight enables unobtrusive, on-demand data transmissions while preserving the user's viewing experience. *Second*, HiLight does not require any modifications to pixel color (RGB) values of the existing image on the screen. Hence HiLight entails low overhead to encode information and is applicable to transmit data in real time for off-the-shelf smart devices. *Third*, HiLight leverages the *relative changes* within a time window, rather than the absolute values, of perceived color intensity (*i.e.* the summation of RGB channel values) to decode information. This makes the system less susceptible to problems such as image blur and frame synchronization.

In the rest of this paper, we will present our initial system design of HiLight, and summarize preliminary results on its feasibility. Using the screen and camera of off-the-shelf smartphones (Samsung S5 and Note 3), we demonstrated that HiLight is able

*VLCS'14,* September 7, 2014, Maui, Hawaii, USA.

http://dx.doi.org/10.1145/2643164.2643171 .

to achieve 11Kbps throughput at normal smartphone viewing distance (30cm). Using a 50-inch TV screen, HiLight can support 3m viewing distance. Our initial results also point out remaining open challenges that we plan to tackle as ongoing work. By offering an unobtrusive, flexible, and lightweight hidden communication channel between screens and cameras, HiLight presents opportunities for new HCI and context-aware applications for smart devices. For example, a user wearing smart glasses can acquire additional personalized information from the screen (*e.g.* TV screen, smartphone or tablet screen) without affecting the content he/she and other users are currently viewing.

# 2. RELATED WORK

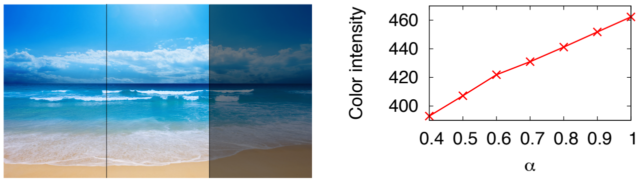
As a particular form of visible light communication, screen-camera links have received increasing interests in recent years. Here we categorize existing research on screen-camera communication into two main categories. The first category of work focuses on building high data rate, reliable screen-camera links. PixNet [14] applies orthogonal frequency division multiplexing (OFDM) to boost the communication data rate between a LCD display and cameras. COBRA [8] and SBVLC [23] propose innovative barcodes for large data transmissions in real-time systems. To improve link reliability, [10, 17] tackle the frame synchronization problem using erasure coding or new preamble detection methods. All these designs, however, require obtrusive coded images shown on the screen, creating unpleasant user viewing experiences. The second category of work aims to improve the user's viewing experience while enabling screen-camera communication. In [22], Yuan *et al* propose a watermarking approach that represents dynamic messages as coded images and embeds them into the original screen image. VRCode [20] designs unobtrusive barcodes by switching some complementary hue-based barcode at 60Hz on the display. PiCode and ViCode [12] integrate barcodes with existing images for better viewing experiences. IVC [6] inserts encoded messages into video and plays the video at a high frame rate to transmit data. While these methods effectively reduce the impact of visual code on human eyes, they require modifications of pixel RGB values, and lead to high encoding overhead. In our work, we aim to enable unobtrusive screen-camera communication without the need of any coded images while promoting low encoding overhead. HiLight is similar in spirit to prior efforts that modulate bits by changing the screen's backlight [7, 11, 13]. However, unlike these designs that require special hardware support (*e.g.* shutter plane [7]) to enable MIMO communication, HiLight uses off-the-shelf smart devices to create a hidden MIMO channel.

# 3. HiLight DESIGN

In this section, we introduce the system design of HiLight. We seek to achieve two goals: 1) unobtrusive data transmissions between screen and cameras without using any coded images; and 2) low encoding overhead to enable on-demand, real-time data transmission for off-the-shelf smart devices. HiLight achieves these goals by leveraging a separate alpha channel to control each pixel's translucency and opacity changes. This represents a dramatic departure from existing screen-camera link designs, which encode data into each pixel's absolute color value. Next, we start with the background knowledge of alpha channel, followed by design challenges of HiLight and our initial solution.

## 3.1 Background and Design Rationale

**Alpha Channel.** As a standard technique in computer graphics, the alpha channel has been widely used to form a composite image with partial or full transparency [16]. While the image element stores the color intensity values in red, green, and blue (RGB) diodes for each pixel of the foreground image [19], the alpha channel stores a value ($\alpha$) between 0 and 1 to indicate the pixel translucency. An $\alpha$ value of 0 means that the foreground image pixel is fully transparent, and 1 (the default value) means that the pixel is fully opaque. Since today's LED and OLED screens use a black matte as the background image, $\alpha$ of 0 means dimming the pixel to black, and $\alpha$ of 1 means displaying the original color intensity of the pixel. Therefore, decreasing the $\alpha$ value of a pixel essentially dims the pixel. As an example, Figure 1(a) shows the screen of a Samsung Note 3, where we uniformly set the $\alpha$ values of the right part of the screen to 0.5, resulting into a darker appearance.



(a) Changing $\alpha$ values of the screen from 1, 0.99, to 0.5

(b) Color intensity perceived by the receiver (camera)

**Figure 1: The impact of changing pixels' $\alpha$ values. (a) shows an image with different $\alpha$ values: the default value 1 (the left), 0.99 (the middle), and 0.5 (the right), using a Samsung Note 3; (b) shows the color intensity of the frame captured by Samsung S5 when varying the $\alpha$ values on the transmitter's screen.**

This dimming effect is perceived as color intensity (summation of RGB channel values) changes by the camera (receiver). To examine the resulting color intensity perceived by off-the-shelf cameras, we set up a Samsung Note 3 phone as the transmitter, and a Samsung S5 phone 15cm apart as the receiver. We adjust the $\alpha$ values of all pixels on the Note 3's screen, and calculate the color intensity at each pixel using the frame captured by S5's camera. As shown in Figure 1(b), increasing $\alpha$ value leads to a linear increase of the color intensity perceived by the receiver's camera. This is because $\alpha$ value is used to multiply the RGB color values. Most importantly, even for the $\alpha$ value change of 0.01 ($\alpha = 0.99$), which is not perceivable by human eyes (Figure 1(a)), the camera of S5 is able to detect the color intensity change. The results demonstrate the potential for off-the-shelf, camera-equipped smart devices to decode information embedded in pixel translucency changes.

**Design Rationale of HiLight.** Motivated by the above observations, HiLight encodes information into pixel's translucency ($\alpha$) changes on transmitter's LED or OLED screen. To decode the information, camera-equipped smart devices examine the changes of the perceived color intensity in captured frames. To make the translucency changes imperceptible by human eyes, HiLight changes the $\alpha$ value by 0.01 ($\alpha = 0.99$) to dim a pixel, and the translucency changes as the screen refreshes. The LED or OLED screens in today's smartphones support 60Hz refresh rate, which is sufficient to avoid the direct flicker effect when a user views a static image [5]. Furthermore, we can divide the screen into grids, and change the $\alpha$ values of all pixels in each grid independently. *Hence each grid functions as an independent transmitter element, resulting into a MIMO channel between screen and cameras.* Overall, operating on the alpha channel, HiLight provides a communication channel unobtrusive to users with low encoding overhead. Also, when dividing screen into small grids, HiLight reduces the impact of frame vertical synchronization issue [10], because each grid operates independently and not all grids contain unsynchronized frame pixels.

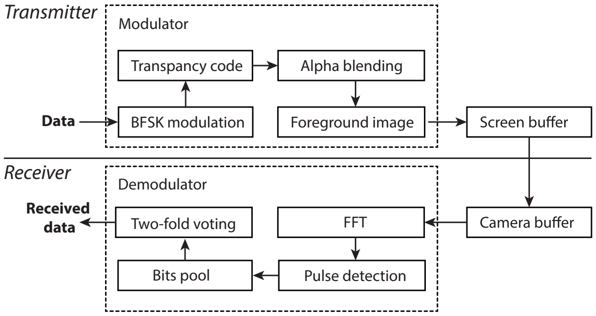
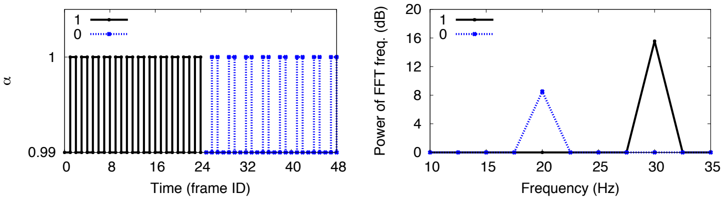
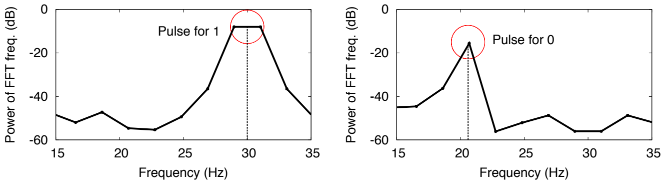


**Figure 2: HiLight system overview.**



(a) Time domain (b) Frequency domain

**Figure 3: Embedding bits into the frequency of pixel translucency ($\alpha$ value) changes. (a) shows the sequence of $\alpha$ values for symbols '01' with 60Hz sampling rate. (b) shows the translucency changes in frequency domain for bit 0 and 1 after applying FFT.**



(a) Frequency pulse for bit '1' (b) Frequency pulse for bit '0'

**Figure 4: Applying FFT to the perceived color intensity values of a single grid over 24 frames at the receiver side. 1% change of the $\alpha$ values at the transmitter side can generate distinguishable pulses in the frequency domain for the receiver to decode data.**

## 3.2 Design Challenges

In a nutshell, the efficacy of HiLight depends on its ability of detecting small changes in pixel translucency and reliably decoding information under various practical settings. This is non-trivial even for a static image because of the following challenges. *First*, illumination changes from ambient light and noises of camera sensors can interfere with the coded translucency changes. Since existing smart devices have a rolling shutter speed up to 120fps and LED display refreshing rate of 60Hz, the available frequency to transmit data is close to DC component (0Hz), making the data decoding vulnerable to noises. *Second*, detecting translucency changes is harder for darker images with lower RGB values because the pixel's RGB value is multiplied by an $\alpha$ value. Hence for a given $\alpha$ change, the larger the pixel's RGB value, the larger the absolute RGB value change, and hence the more easily the camera can detect the changes. It requires us to seek solutions that reduce the system's sensitivity to image colors. *Third*, to support on-demand data transmission, the receiver needs to synchronize with the transmitter and detect the starting point of a data stream. This is challenging given that screen-camera communication is a one-way channel.

## 3.3 Initial Solution

We now describe our initial solution to tackle the above challenges. At a high level, HiLight encodes data using Frequency Shift Keys (FSK). We judiciously design the modulation keys to avoid low frequency components close to the frequency of environmental noise. This reduces the impact of noise on data decoding. To detect the start and end of a data stream, HiLight aggregates the perceived bits of all pixels on both the time and spatial domain to reduce the impact of frame synchronization and environmental noise. Finally, we explore the use of erasure coding [18] to add data redundancy and recover information in dark areas. Figure 2 illustrates the flowchart of HiLight. We next introduce each step in detail, assuming that the receiver records video at 60fps. We introduce HiLight with erasure coding in Section 3.4.

**Modulating Data.** For a given bit stream and pre-defined grids on the screen, the transmitter first modulates bits using Binary Frequency Shift Keying (BFSK). We select BFSK because of its simplicity. We are interested in exploring more advanced modulation schemes as part of future work. Here bit 0 and 1 are represented by different frequency of translucency changes over a certain number of frames (24 frames in our implementation). Figure 3(a) shows the sequence of $\alpha$ values of a grid on the transmitter's screen to represent bit 0 and 1. Unlike Pharos [9] that keeps the duty cycle to 50%, HiLight adjusts the duty cycle to remove the low frequency components ($<$ 15Hz). Hence the frequency power of the data bit will be less affected by environmental noise.

To transmit each bit, the transmitter generates the sequence of $\alpha$ (translucency) values of each pixel for 24 frames. Pixels within a grid have the same $\alpha$ value. It then applies alpha blending method [19], a standard technique in computer graphics, using OpenGL API to combine the foreground image with background image (black matte), and outputs the frames into the screen buffer.

**Demodulating Data.** Once the receiver captures frames in its camera buffer, it applies Fast Fourier Transform (FFT) to project the translucency changes in a decoding window (24 frames) into the frequency domain. As shown in Figure 3(b), the translucency changes for bit 0 and 1 lead to frequency power pulses centered on 20Hz and 30Hz respectively. Hence the receiver filters the frequency components below 15Hz, and checks the power of 20Hz and 30Hz frequency components. If either component has the highest power across all frequency components above 15Hz, the receiver then applies an edge detection method [21] to calculate the first-order derivative of this frequency component. If the absolute derivative value is above a threshold (3 in our implementation), the receiver then outputs the corresponding bit into a candidate pool. The receiver slides the decoding window by one frame and repeats the above procedure.

Given all candidate bits in the bit pool, HiLight determines the received bit for every 24 frames as the bit with the most appearances. To detect the first frame of a data transmission, HiLight checks the number of detected bits for a given frame. If more than 60% of grids detect a bit, HiLight treats this frame as the first frame. Clearly, using more frames to transmit a bit helps reduce bit errors caused by frame synchronization and environmental noises (*e.g.*, ambient light and camera sensor noise), and thus leads to higher decoding accuracy at the receiver side. From our experiments, we observe that using 24 frames can remove most bit errors. Figure 4 plots the frequency components after applying FFT on the perceived color intensity values of a single grid over 24 frames. We observe that even with 1% change of the $\alpha$ value, a window of 24 frames is able to generate distinguishable pulses in the frequency domain. The downside of using 24 frames per bit is the associated delay (24/60=0.4 second) to transmit a single bit, which limits the data rate achieved by a single grid. An interesting open

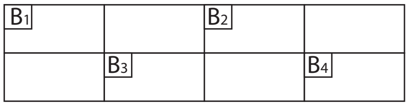


**Figure 5: An optimal block placement for an example block group when $K = 3$. The screen is divided into 8 regions, and the 4 elements of this block group are placed at $B_1, B_2, B_3$ and $B_4$.**

problem is to design more sophisticated modulation and demodulation schemes that use fewer frames to transmit and decode each bit reliably. We plan it as a future work (Section 5).

## 3.4 HiLight with Erasure Coding

The basic design of HiLight directly distributes data bits to grids. Clearly for grids located in the image's dark area, decoding the translucency changes is hard and leads to high bit errors. To improve the reliability of data transmission, we explore the use of erasure coding to add redundancy [18]. The key idea is to add $n$ redundant bits for every $m$ data bits. By the property of erasure coding, the receiver can recover the $m$ data bits if it successfully decodes any $m$ bits of the $(m + n)$ bits.

In particular, we apply Vandermonde Reed-Solomon coding [15] to encode data bits. That is, we multiply every $K$ data bits by a distribution matrix that consists of an identity matrix and a Vandermonde matrix. Take $K = 3$ as an example and let $d_i, 0 < i \leq K$ denote each data bit, we calculate the code as the following:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ r \end{bmatrix}, \tag{1}$$

where the redundant data $r = \sum_{i=1}^{K} d_i \leq K$. Since each grid can transmit only 0 or 1, we need a block of $\lceil \log(K + 1) \rceil$ grids to represent $r$. The $K$ data grids and 1 redundant data block form a block group, where successfully receiving any $K$ of them allows the recovery of $K$ data bits. The redundancy ratio of each block group is $\frac{\lceil \log(K+1) \rceil}{K + \lceil \log(K+1) \rceil}$. When $K = 3$, each redundant block has 2 grids, thus each block group has $\frac{2}{3+2} = 40\%$ redundancy.

Now given the block groups, the next question is how to place them on the screen. To reduce the impact of dark areas, we maximally spread out the $(K + 1)$ elements ($K$ data grids and 1 redundant data block) of each block group on the screen. This minimizes the possibility of having multiple elements in dark areas and thus maximizes the probability of data recovery. Specifically, we first divide the screen into $2(K + 1)$ regions. We then distribute the $(K + 1)$ elements of each block group into these regions, so that we maximize the minimal distance between any two elements in the same group. Figure 5 shows an example placement for a block group with data grids $B_1, B_2, B_3$ and a redundant data block $B_4$, assuming $K = 3$. We have proved the optimality of our placement solution, and omit the proof in the interest of space.

# 4. HiLight EXPERIMENTS

We now examine the feasibility of HiLight using off-the-shelf smart phones. We aim to understand the throughput and accuracy HiLight achieves under different practical settings. We define the throughput as the number of bits that the receiver receives correctly per second, and the decoding accuracy is the percentage of bits received successfully over all bits transmitted.

## 4.1 Experimental Setup

We implemented HiLight's modulation and demodulation in a MATLAB emulator. The modulator takes the random bits we generated and a static image as input, and outputs a video where the data bits are embedded into pixel translucency changes. We play the video on a Samsung Note 3 Android phone as the transmitter, and use a Samsung S5 phone to capture the frames. These captured frames are the input for the demodulator to decode transmitted bits. As the next step, we plan to implement the modulation and demodulation on existing smart device for real-time encoding and decoding (Section 5).

Figure 6 shows our experimental setup, where we fix the receiver (left) and the transmitter (right) on two phone holders. This avoids the problems of image alignment and image blur (we plan it for a ongoing work), and allows us to focus on examining HiLight feasibility. Specifically, the transmitter is a Samsung Note 3 with 5.7-inch OLED display (1920 x 1080 pixels). The display is 12.6cm x 7.2cm in size and its refreshing rate is 60Hz. The receiver is a Samsung S5 with a rear camera that has a resolution of 16 megapixels and supports 120fps video recording. To match the refreshing rate of the display, we used 60fps at the receiver side. For all experiments, we used the default auto-focus function on S5's camera to record videos. We ignore the first 2 seconds of the video because of the high noise during initializing the auto-focus function.

We evaluate HiLight in terms of its accuracy and throughput. By default, we used HiLight basic design without erasure coding. We next present our preliminary results to understand the impact of image colors, grid sizes, environmental factors such as ambient light and distance, as well as the benefits of erasure coding scheme.

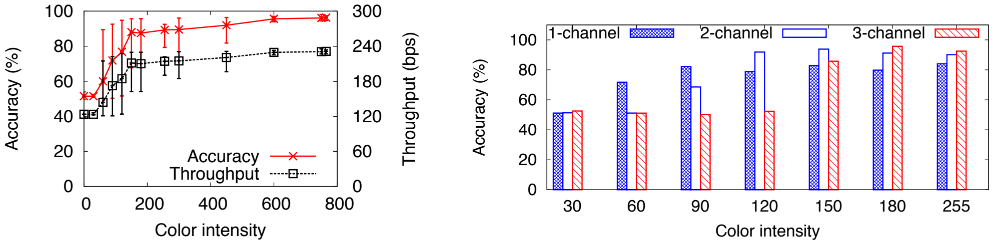
## 4.2 Preliminary Results

**Impact of Foreground Image Color.** As we discussed in Section 3.2, the color of screen's foreground image affects camera's ability to detect color intensity changes. To understand the impact of image color on HiLight system, we tested 61 single-color images with colors varying from black (0, 0, 0) to white (255, 255, 255). The transmitter and receiver are 15cm away, which maximizes the area of the transmitter's screen in the viewfinder of the receiver's camera and prevents the out-of-focus problem. We divided the transmitter's screen into grids, where each grid is $94.5mm^2$ in size and has 160x135 pixels. Using the basic HiLight design without erasure coding, we encoded bits using each single-color image as the foreground image, played the video on Note 3 phone's screen, and decoded bits using the frames captured by S5's camera. We repeated the experiment for 5 rounds for each image. All experiments were performed under fluorescent lamp with illuminance ≈14lux.

Figure 7(a) shows the accuracy and throughput of HiLight as a function of image color intensity. Larger color intensity indicates a brighter color. As expected, as the color becomes brighter, HiLight's performance gradually stabilizes. This is because the pixel translucency changes for brighter colors lead to larger absolute color intensity changes, and hence are easier for the camera to detect. Overall we observe that once the color intensity is above 200, HiLight achieves 90%+ accuracy and 230-240bps throughput.

We further examine the impact of color intensity distribution in three RGB channels (red, green, and blue), by testing colors with the same color intensity. Taking a color intensity of 30 as an example, we examined 1-channel colors such as (30, 0, 0), 2-channel colors such as (15, 15, 0), and 3-channel colors such as (10, 10, 10). Figure 7(b) shows the resulting accuracy for different RGB distributions for each given color intensity. We observe that for 2-channel and 3-channel colors, HiLight requires a higher color intensity (120 - 150) to achieve 80%+ accuracy and stabilize. Compared to 1-channel colors, 2-channel and 3-channel colors spread the RGB value across more channels and each color channel has a

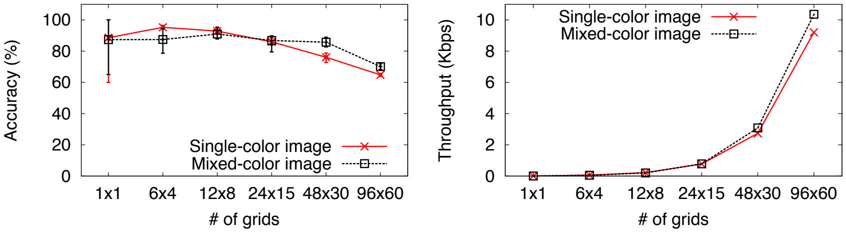
**Figure 6:** **Experimental setup with Samsung Note 3 (right) as the transmitter and Samsung S5 as the receiver (left).**



(a) Impact of RGB sum

(b) Impact of RGB value distribution

**Figure 7:** **Impact of foreground image color on HiLight accuracy and throughput, using HiLight basic design without erasure coding, grid size of 94.5$mm^2$ (*i.e.* 12x8 grids), and 61 single-color foreground images.**



(a) Accuracy

(b) Throughput

**Figure 8:** **Impact of grid size on HiLight accuracy and throughput.**

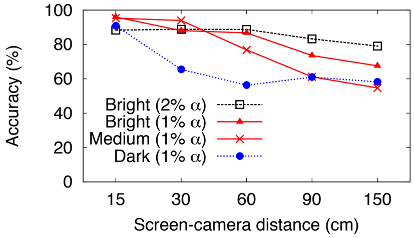


**Figure 9:** **Impact of screen-camera distance under three indoor ambient light conditions when changing $\alpha$ values by 1% and 2%.**

**Table 1:** **Grid settings.**

| **# of grids** | 1x1 | 6x4 | 12x8 | 24x15 | 48x30 | 96x60 |
|---|---|---|---|---|---|---|
| **Size** ($mm^2$) | 9072 | 378 | 94.5 | 25.44 | 6.24 | 1.56 |
| **Grid pixels** | 1920x1080 | 320x270 | 136x135 | 80x72 | 40x36 | 20x18 |

lower value. Hence HiLight performs better when the RGB value at non-zero color channel is above a threshold (50 - 60).

**Impact of Grid Size.** HiLight divides the transmitter's screen into grids to enable multiple independent transmitter elements. Next, we examine the impact of grid size on HiLight performance. Using the experiment setting in Section 4.1, we tested 6 grid settings summarized in Table 1, using both single-color images with color intensity values from 120 to 600 and real images with mixed colors.

Figure 8 shows the accuracy and throughput as the number of grids increases. Overall, HiLight's accuracy drops as we increase grids. This is because a larger number of grids lead to a smaller grid size, making the system more susceptible to the inter-symbol interference [14]. On the other hand, a larger number of grids allow the transmitter to transmit more bits simultaneously. The increase in transmitted bits outweighs the accuracy drop. Therefore, the system throughput grows as the number of grids increases. In particular, with a grid size of 1.56$mm^2$, HiLight achieves 9-11Kbps. The minimal grid size is ultimately limited by the resolution of camera sensor and screen-camera distance.

**Environmental Factors.** We now move on to evaluating the impact of ambient light and screen-camera distance. We tested three indoor ambient light conditions: 1) *dark* (1lux), 2) *medium* (9lux), and 3) *bright* (14lux), where we measured the light illuminance in each condition using the Light Sensor app [1] in Google Play. For each illuminance condition, we increased the screen-camera distance from 15cm to 150cm. Since the size of the transmitter (Samsung Note 3) screen is 5.7 inches, 30cm is the comfortable viewing distance. We used grid size of 94.5$mm^2$ and a real foreground image with mixed colors. We measured the decoding accuracy in each setting, and repeated all experiments for 5 rounds.

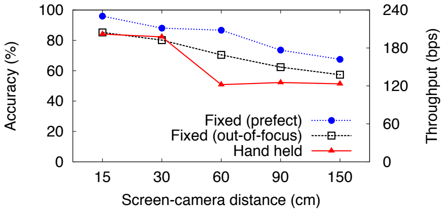


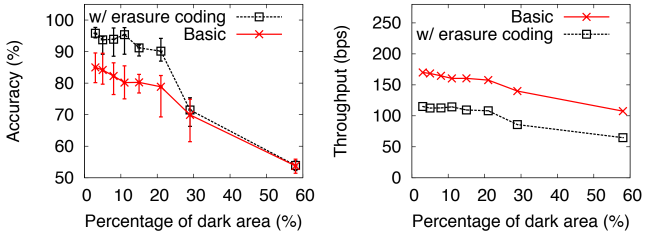
**Figure 10:** **Impact of hand motion on HiLight accuracy.**

Figure 9 shows HiLight accuracy as the screen-camera distance increases in each ambient light condition. While our current implementation changes the $\alpha$ value by 1% (changing $\alpha$ from 1 to 0.99) to ensure changes invisible to users, we also tested larger (2%) $\alpha$ value changes (changing $\alpha$ from 1 to 0.98) to understand the impact of $\alpha$ changes. Our key observations are as follows. *First*, HiLight maintains high accuracy (90%+) within the normal viewing distance (30cm) for the 5.7in screen. We also tested HiLight using a 50-inch TV screen, and we observed that HiLight can support up to 3m viewing distance in bright or medium light condition. As the screen-camera distance further increases, it becomes harder for the camera to detect the color intensity changes on the transmitter's screen, and thus both the accuracy and throughput of HiLight drop.

*Second*, HiLight works best in bright environment (*i.e.*, the bright and medium settings), because the dark setting introduces high camera noise. As a result, HiLight supports a shorter distance (15cm) to maintain 90%+ accuracy in the dark setting.

*Third*, larger change of pixels' $\alpha$ values leads to higher accuracy, because it generates more significant changes in color intensity and the receiver can detect the intensity changes more accurately. This can serve as a potential solution to supporting longer viewing distance or darker environment. The key challenge, though, is to ensure that the changes are unobtrusive to human eyes.

**Impact of Hand Motion.** So far we have assumed that the transmitter and receiver are perfectly aligned by fixing them on phone holders (Figure 6). In real settings, users hold phones in their hands, and their hand motion can lead to image blur by being out-of-focus, and cause screen misalignment at the receiver side. To



(a) Accuracy (b) Throughput

**Figure 11: HiLight with erasure coding.**

examine the impact, we held the transmitter phone in the air facing a fixed receiver (camera), and measured the decoding accuracy. We also tested the case with only image blur, where we fixed both phones on phone holders and set the transmitter screen outside of the focus area in the camera preview window. This allows us to understand the impact of screen misalignment caused by hand motion. Figure 10 compares the accuracy in both cases to that with perfect alignment. We see that within the normal viewing distance (30cm), motion blur is the dominating factor, leading to less than 10% decrease in accuracy. As the viewing distance further increases, the receiver is more sensitive to screen misalignment, and fails to correctly identify the grids on transmitter screen. Thus the accuracy in the hand-held case drops quickly. To solve the problem, we can apply image tracking and alignment algorithms in computer vision to better track transmitter screen. We plan it as our future work.

**Benefits of Erasure Coding.** Finally, we evaluate HiLight with erasure coding to understand its gain when dealing with images with dark areas. We selected 8 images with different percentages of dark areas, where a dark area refers to an area with pixels whose color intensity values are less than 200 (Figure 7(a)). Figure 11 compares HiLight's basic design to HiLight with erasure coding in terms of accuracy and throughput, using screen-camera distance of 30cm and grid size of 113.4$mm^2$. Overall by adding redundancy, erasure coding brings moderate gain (up to 15%) in accuracy while sacrificing end-to-end throughput. Furthermore, since our coding adds 1 redundant data block for every 3 data bits (Section 3.4), the system is unable to recover bits in dark areas once the dark areas occupy more than 20%+ of the image. This indicates the need of adapting coding redundancy to the channel condition (*e.g.*, the percentage of dark areas in an image and the ambient light). We plan this study as part of ongoing work (Section 5).

## 5. CONCLUSION AND FUTURE WORK

We presented HiLight, a new form of screen-camera communication that hides information underlying any image on the screen. We introduced our initial HiLight system design, and demonstrated its feasibility using off-the-shelf smartphones.

Moving forward, we plan to address the following challenges. *First*, HiLight currently supports only static foreground images. Supporting video brings new challenges, because a video naturally contains fast image changes and the resulting color intensity changes interfere with decoding HiLight data. We are studying new modulation schemes to extract color intensity changes associated with the transmitted data. *Second*, HiLight's throughput is currently limited by the speed of pixel translucency changes on an OLED screen, which is bounded by the screen's refreshing rate (60Hz in most existing smart devices). Theoretically each pixel on an OLED screen has a response time of 0.01ms - 0.1ms [2], and thus physically supports frequency range of 10KHz - 100KHz . We are investigating techniques to approach this physical limit and further boost HiLight's data rate. *Third*, we are exploring schemes to adapt transmitter's configuration (*e.g.* coding redundancy, the level of pixel translucency changes, grid size), based on the current link condition (*e.g.* ambient light level, image darkness) and receiver-side feedback (*e.g.* link distance). Such adaptation is essential for HiLight to support mobile receivers and diverse environmental settings. We also plan to study vision algorithms (*e.g.*, image tracking and alignment) to improve HiLight's robustness against image blur in the mobile scenario. *Fourth*, while our current modulation design uses BFSK over 24 frames to demonstrate our idea, we are investigating more advanced modulation schemes that can reliably encode data using fewer frames. This can boost HiLight data rate and reduce transmission delay. *Finally*, we plan to implement our design on existing smart devices to realize real-time data transmissions between OLED screens and smart devices.

## 6. REFERENCES

[1] `https://play.google.com/store/apps/details?id=com.app.lightsensor`.

[2] `http://en.wikipedia.org/wiki/OLED`.

[3] Automatic identification and data capture techniques - QR code 2005 bar code symbology specification. ISO/IEC 18004:2006.

[4] Ashok, A., et al. Challenge: Mobile optical networks through visual MIMO. In *Proc. of MobiCom* (2010).

[5] Bullough, J., et al. Effects of flicker characteristics from solid-state lighting on detection, acceptability and comfort. *Lighting Research and Technology 43*, 3 (2011), 337–348.

[6] Carvalho, R., Chu, C.-H., and Chen, L.-J. IVC: Imperceptible video communication.

[7] Fattal, D., et al. A multi-directional backlight for a wide-angle, glasses-free three-dimensional display. *Nature 495*, 7441 (2013), 348–351.

[8] Hao, T., Zhou, R., and Xing, G. COBRA: Color barcode streaming for smartphone systems. In *Proc. of MobiSys* (2012).

[9] Hu, P., et al. Pharos: Enable physical analytics through visible light based indoor localization. In *Proc. of HotNets* (2013).

[10] Hu, W., Gu, H., and Pu, Q. LightSync: Unsynchronized visual communication over screen-camera links. In *Proc. of MobiCom* (2013).

[11] Huang, S. Backlight modulation circuit having rough and fine illumination signal processing circuit, Mar. 27 2012. US Patent 8,144,112.

[12] Huang, W., and Mow, W. H. PiCode: 2D barcode with embedded picture and ViCode: 3D barcode with embedded video (poster). In *Proc. of MobiCom* (2013).

[13] Kimura, K., Masuda, S., and Hayashi, M. Display apparatus and method for controlling a backlight with multiple light sources of a display unit, Sept. 11 2012. US Patent 8,264,447.

[14] Perli, S. D., Ahmed, N., and Katabi, D. PixNet: Interference-free wireless links using LCD-camera pairs. In *Proc. of MobiCom* (2010).

[15] Plank, J. S., et al. A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. *Softw., Pract. Exper.* (1997).

[16] Porter, T., and Duff, T. Compositing digital images. In *ACM Siggraph Computer Graphics* (1984).

[17] Rajagopal, N., Lazik, P., and Rowe, A. Visual light landmarks for mobile devices. In *Proc. of IPSN* (2014).

[18] Rizzo, L. Effective erasure codes for reliable computer communication protocols. *ACM SIGCOMM CCR* (1997).

[19] Tan, K. W., et al. FOCUS: a usable & effective approach to OLED display power management. In *UbiComp* (2013).

[20] Woo, G., Lippman, A., and Raskar, R. VRCodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *Proc. of ISMAR* (2012).

[21] Yang, L., et al. Supporting demanding wireless applications with frequency-agile radios. In *Proc. of NSDI* (2010).

[22] Yuan, W., et al. Dynamic and invisible messaging for visual mimo. In *IEEE Workshop on Applications of Computer Vision (WACV)* (2012).

[23] Zhang, B., et al. SBVLC: Secure barcode-based visible light communication for smartphones. In *Proc. of INFOCOM* (2014).