Styrofoam: A Tightly Packed Coding Scheme for Camera-based Visible Light Communication

Robert LiKamWa*, David Ramirez*, and Jason Holloway*

Rice University, Dept. of Electrical and Computer Engineering, Houston, TX

(* Equal Contribution)

ABSTRACT

Screen-to-camera visible-light communication links are fundamentally limited by inter-symbol interference, in which the camera receives multiple overlapping symbols in a single capture exposure. By determining interference constraints, we are able to decode symbols with multi-bit depth across all three color channels. We present Styrofoam, a coding scheme which optimally satisfies the constraints by inserting blank frames into the transmission pattern. The coding scheme improves upon the state-of-the-art in camera-based visible-light communication by: (1) ensuring a decode with at least half-exposure of colored multi-bit symbols, (2) limiting decode latency to two transmission frames, and (3) transmitting 0.4 bytes per grid block at the slowest camera's frame rate. In doing so, we outperform peer unsynchronized VLC transmission schemes by 2.9x. Our implementation on smartphone displays and cameras achieves 69.1 kbps.

1. INTRODUCTION

Camera-based visible light communication (VLC) captures images to decode light source transmissions. While slower than photodiode-based VLC solutions, camera-based VLC can be established on existing display and camera devices, allowing for easy adoption, e.g. through the installation of an app on a smartphone. This makes VLC an appealing medium for transmitting large advertisements, menus, museum dialogues, or image, video, or audio media without an Internet infrastructure or any special sensors besides a camera. The advent of wearable devices in the form of wristwatches and eyeglasses also promotes communication scenarios that optimize ease-of-access with minimal setup.

Various camera-based VLC solutions have encoded light on LED arrays [7] and screen displays [8]. However, in all cases, the lack of synchronization between the screen and camera limits the rate of communication–if the light source transmits faster than the camera's frame rate, the camera will capture multiple code exposures in a single image. Previous works have avoided or mitigated the inter-symbol interference by slowing the frame rate [2], by sacrificing color channels [5], or by inserting additional coding frames [3].

As opposed to avoiding the interference of capturing multiple codes, we design our system, ***Styrofoam***, by characterizing and constraining inter-symbol interference through the introduction of a blank frame. By limiting the length of the blank frame, we maximize the transmission rate of symbols while guaranteeing the ability to decode multi-bit symbols over a half-exposure. Thus, Styrofoam inserts "air" (blank frames) in the "package" (transmission) to guarantee that it can be safely "unpacked" (decoded).

We show that enabling multi-bit codes far outweighs the cost of introducing a blank frame. Additionally, Styrofoam supports continuous streams; it decodes after receiving two frames, regardless of when the transmission began. Styrofoam supports any scenario with displays transmitting at up to twice the frame rate of the slowest supported camera.

Theoretically, Styrofoam codes can be transmitted on any display or LED array which transmits narrow-band RGB signals, as do most smartphone displays and hobby-oriented LED arrays, e.g. AdaFruit NeoPixel. We implement and characterize Styrofoam by transmitting a streaming barcode on a 60 fps LCD screen with a 30 fps smartphone camera receiver. On this setup, we achieve a 2.9x faster throughput than other mobile streaming barcodes, including LightSync [3], COBRA [2], or Unsynchronized Barcodes [5].

2. COMMUNICATION CHANNEL MODEL

In order to faithfully mimic communication scenarios encountered "in the wild", we design a system model for unsynchronized uni-directional communication between a light-emitting transmitter and a camera receiver. This section details the assumptions included in our device-to-device model.

2.1 Screen Transmitter Model

We adopt a transmission scheme similar to other VLC works, namely, streaming a barcode over a grid of blocks hosted by an LED array or display screen updating on a fixed frame rate period. We treat grid blocks as orthogonal channels and model each separately. Each grid block transmits one RGB color per frame period.

The act of code registration indicates where in a camera frame the code lies. This is well-studied, e.g., COBRA [2] sufficiently provides registration with the inclusion of tracking blocks on the borders of the barcode grid. Hence, we consider code registration outside of the scope of this work.

2.1.1 *Transmitter Characteristics*

Styrofoam uses symbols spanning different color channels. This requires that the transmitter be a light source capable of modulating light intensity in non-overlapping red, green, and blue channels. We assume that the output intensity is spatially uniform across the display. The output must also bright enough that the minimum non-zero symbol intensity dominates ambient light to ensure a sufficient signal-to-noise ratio (SNR) for accurate decoding. This characteristic is satisfied with existing LCD and OLED displays indoors.

2.2 Camera Receiver Model

Smartphones provide a convenient camera and computation platform for VLC reception. However, smartphone imaging pipelines are optimized to produce perceptually pleasing images which entails purposefully altering color values, introducing non-linearity, and quantizing pixel intensities to 8 bits. While this results in visually appealing images, pixel intensities do not faithfully represent the measured light intensity. The correct representation of light intensity enables Styrofoam to use colored multi-bit symbols.

This subsection provides background to enumerate sources of error and uses standard correction methods to convert captured pixel values to symbol intensities.

2.2.1 *Gamma Correction*

While the image sensor captures values linearly proportional to light intensity, the human eye experiences a non-linear response to intensity. Device manufacturers compensate for this disparity by mapping input intensity (I) to a pixel value (P) using an exponential parameter, gamma (γ):

$$P = I^{1/\gamma}.$$

In the sRGB color space used by many consumer cameras, γ is set to 2.2 [1]. We undo the γ-correction and recover approximate linear intensity values, $\hat{I} \approx I$. The error between true and recovered intensities (due to quantization of P) is treated as noise during symbol decoding.

To unmap the gamma correction, the receiver must know the range of received intensity to properly scale the input intensities. The receiver obtains this range by observing fixed white, gray, and black values in the tracking code. The exposure and ISO of the capture may need to be controlled to allow the receiver to capture the full range of intensities.

2.2.2 *Color Cross-talk*

The spectral sensitivities of the red, green, and blue channels on an image sensor are not orthogonal and may overlap. Furthermore, white balancing and color enhancement processing shifts color combinations to optimize for visual appeal. Thus, a color channel may respond differently depending on the intensity in other color channels. These cross-talk effects are highly device-dependent. To avoid these effects, we ensure that our camera receives blocks containing strictly red, green, or blue, preventing color mixture interference.

2.2.3 *Rolling Shutter*

Most CMOS image sensors use a technique called *rolling shutter* to read out pixel rows while other rows are being exposed, allowing parallel readout and exposure. To maintain consistent exposure across the image, sensors begin exposing each row a set time before it is to be read out. As a consequence, each row of the image will be exposed during a different window of time. To compensate for this, we decode each row independently.

2.2.4 *Frame Timing*

The beginnings of each frame capture are separated by a constant time, the inverse of the capture frame rate. For a 30 fps camera, the capture period is $1/30$ Hz = 33 ms.

Pixels are exposed between readouts, but because of the rolling shutter, the exposure time of a row can be nearly as much as the capture separation time. However, bright light sources, such as LCD and OLED screens, may require the exposure time of a row to be significantly less than the frame separation to avoid saturating pixel intensities, e.g., a 10 ms exposure within a 33 ms capture period. Thus, there may be a sizable time gap between frame exposures during which no transmission is captured.

2.2.5 *Distinguishable Color Levels*

A major advantage of Styrofoam is the ability to encode information within a color channel by modulating the light intensity. The transmission scheme in Section 3.2 ensures that each symbol is captured with an exposure of no less than half of the full integration period. To enable accurate symbol decoding, levels must be assigned to be distinguishable at half-exposure in the presence of camera noise.

Spatial averaging over received blocks combined with aggressive denoising in the imaging pipeline greatly reduces the amount of image noise. We model the remaining camera noise–including quantization error–as additive white Gaussian noise (AWGN). To avoid ambiguity between neighboring levels, AWGN requires a minimum spacing of three times the standard deviation. The Lumia 1020 noise in the red channel is measured to have 1.55 standard deviation on a linearized 256 scale. Out of a half-exposure of 128, spacing rounding by six times the standard deviation gives a maximum number of 13 intensity levels.

2.3 Communication Symbols

2.3.1 *Symbol Coding*

Due to the limits on distinguishability in Section 2.2.5, we assign levels in each color channel according to the noise spacing restriction. The number of levels in a channel is also limited by the maximum received intensity for each color. The relationship between the transmitted and received intensities for our testing system are shown in Figure 1. Notice that the blue and green channels do not fully span the intensity range of the camera, resulting in fewer levels in these

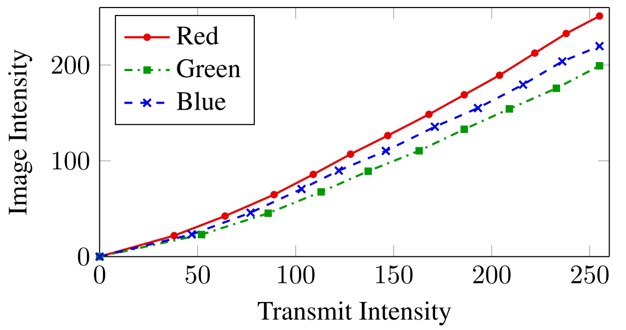


Figure 1: Mapping transmitted intensities to received values. The maximum number of levels per color channel is governed by the maximum intensity value in that channel and the 3σ spacing to deter noise interference.

channels. The maximum symbol depth is thus the sum of the number of unique intensities that can be accurately recovered within each channel: 13, 10, and 11, respectively for red, green, and blue. As all color channels share the black value, this gives each symbol 32 possible values.

2.3.2 *Inter-Symbol Interference*

As the camera and display are unsynchronized, the receiver's captured grid block r_j can capture multiple transmitted symbols s_k in a single exposure. The symbols are linearly weighted by the time spent in exposure. Thus, we model the received block r_j as a linear combination:

$$r_j = \sum \alpha_{jk} s_k, \tag{1}$$

where α_{jk} is the proportion of time the k^{th} transmitted symbol is exposed in the j^{th} camera frame.

2.3.3 *Timing Blocks*

Adjacent to each row of blocks, we use two timing blocks to allow the receiver to directly observe α_{jk} values. We set one timing block to be on during even-indexed frames with the other timing block on during odd-indexed frames. Both timing blocks transmit the maximum value of red. The values of the timing blocks, once linearized, are the α_{jk} exposure coefficients. Because the camera may use a rolling shutter, each barcode row may have different alpha values. Thus each row needs its own set of timing blocks. To allow the camera to decode in horizontal and vertical orientations, we place our exposure timing on the first and third rows and columns of the tracking pattern, as shown in Figure 3(c).

3. CODING SCHEME

We design the Styrofoam coding scheme to decode symbols in spite of inter-symbol interference by inserting blank frames between symbol transmissions. Introducing these gaps decreases the transmit frame rate, but enables each symbol to carry more depth, ultimately increasing the data rate.

3.1 Constraining Interference

While inter-symbol interference is unavoidable, we present the following constraints which limit the interference to guarantee decodability.

3.1.1 *Symbol Visibility*

To ensure that all symbols are captured by a frame in the capture stream, ***each symbol must be displayed for at least one capture period***. As the exposure time is always less than the capture separation time, this also ensures that inter-symbol interference is limited to two symbols.

3.1.2 *Symbol Resolvability*

We define a *pure* capture as having no interference, i.e., only one $\alpha_{jk} \neq 0$:

$$r_{jk}^{\text{pure}} = \alpha_{jk} s_k, \tag{2}$$

and a *mixed* capture as having captured a pair of symbols:

$$r_{jk}^{\text{mix}} = \alpha_{jk} s_k + \alpha_{j(k+1)} s_{(k+1)}. \tag{3}$$

Symbols received in pure captures are resolvable by dividing by the exposure coefficient:

$$s_k = \frac{r_{jk}^{\text{pure}}}{\alpha_{jk}}, \tag{4}$$

where α_{jk} is directly observed, as described in Section 2.3.3.

In a mixed capture, a target symbol in a pair can be resolved as long as the other symbol is resolved. For example, if s_k is the target symbol and $s_{(k+1)}$ was previously resolved, then s_k can be solved by

$$s_k = \frac{r_{jk}^{\text{mix}} - \alpha_{j(k+1)} s_{(k+1)}}{\alpha_{jk}}. \tag{5}$$

Therefore, ***every target symbol in a mixed capture pair with a resolvable symbol***, to become itself resolvable.

3.1.3 *Symbol Discernability*

To decode the value of a target symbol in either a pure or mixed capture, α_{jk} must be sufficiently large to prevent numerical instability from the divide operation. For symmetry, we target that α_{jk}>0.5 for a symbol k to be decodable from capture j. Note that in a mixed capture, the mixed symbol's α_{jk} coefficient is only used to weight the symbol subtraction, and does not have to satisfy the constraint. Therefore, to decode properly, ***the target symbol must carry at least half of the frame exposure to be resolved***.

3.1.4 *Symbol Color Constancy*

As discussed in Section 2.2.2, color cross-talk causes captures with multiple color channels to misrepresent symbol values. Hence, only a single color channel must be exposed in a capture frame. While symbols are limited to single color channels, this additionally implies that ***decodable mixed captures must contain symbols with matching color channels***.

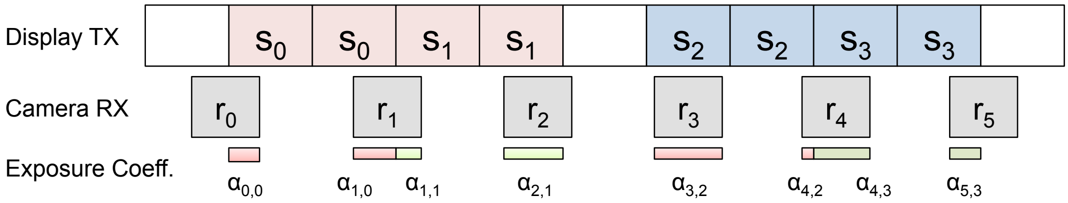


Figure 2: Styrofoam transmission scheme for a single block. Symbols are sent over two display transmitter frames. The camera capture frames with a period of no more than that of two transmitter frames. Due to inter-symbol interference, some received frames capture a weighted sum of symbols. r_1 and r_4 are mixed captures with target symbols of s_0 and s_3. r_2 and r_3 are pure captures of s_0 and s_3 with sufficient exposure, which allow the decode of the mixed symbols.

3.2 Scheduling Symbol Transmission

Following the constraints, we design Styrofoam to transmit and decode frames in spite of inter-symbol interference.

3.2.1 Capture Frame Rate vs. Symbol Time

To satisfy constraint 3.1.1, each symbol must be displayed for the maximum supported capture period, corresponding to the camera with the slowest frame rate. This naturally ensures that each symbol is seen as a target symbol with sufficient exposure in a pure capture or a mixed capture.

While pure captures can be decoded using Equation 4, the next section details how to decode mixed captures.

3.2.2 Mixed Capture Decode

Consider the mixed capture case where the target symbol A precedes the mixed symbol B. By the target symbol definition, A was exposed for $\alpha_{jk} \geq 0.5$. Because the visibility constraint from Section 3.1.1 enforces that the capture period is less than symbol spacing, the next frame's exposure will begin at least α_{jk} before the transmission of symbol B ends, guaranteeing that $\alpha_{(j+1)(k+1)} \geq \alpha_{jk} \geq 0.5$. Therefore, the mixed symbol B experiences a sufficient exposure as a target symbol in either a pure or mixed capture. Thus, A can be decoded if B can be decoded. We can thus inductively prove that all symbols in a mixed capture chain can eventually be decoded so long as one of the mixed symbols occurs in a pure capture. (The case when the target symbol follows the mixed symbol holds in the reverse direction.)

However, if the symbol spacing is equal to the capture period, the mixed symbol will never occur in a pure capture, thwarting resolvability. Furthermore, waiting for a pure capture induces latency before symbols can be decoded. Therefore, to reduce decode latency to N symbols, we transmit a blank frame after every N^{th} symbol. As we are guaranteed to have a target symbol of $\alpha_{jk} \geq 0.5$, the blank frame time simply needs to fill in the maximum possible remainder: half of the symbol displayed time.

As all transmissions are multiples of the blank frame time, we use it as the transmitter frame period. Thus, each of the N symbols is transmitted for 2 frames, while each blank frame occupies 1 frame. The inverse of the blank frame time is thus the transmitter frame rate. As the symbol period is at least equal to capture period, the transmitter sends frames slower than twice the rate of the slowest receiver.

3.2.3 Color Channels

Because of the color constancy requirement in Section 3.1.4, not only must symbols be single-channel, but adjacent symbols must also be within the same channel to prevent irrecoverable interference. When using a color-channel, the symbol depth L is set to the number of levels supported by the color channel, as discussed in Section 2.3.1. (For our setup, L = 13, 11, or 10 for red, green or blue channels.) Sending N symbols over a single color channel will allow each channel to transmit L levels of information.

However, the introduction of a blank frame introduces the opportunity to change color. The color change itself is used to encode information, as it can be one of a possible set of color channels of size $C = 3$ (for R, G, or B). Thus, the transmit rate (ρ) with relation to the choice of N is:

$$\rho = \frac{\log_2(CL^N)}{2N+1}[\text{bits/block}]. \tag{6}$$

With $C = 3$ colors and $L > 9$ levels, the transmit rate for $N + 1$ is higher than the transmit rate than for N; asymptotically more per-block information can be added by delaying the blank frame.

The choice of N also incurs a tradeoff between decode latency (N frames) and symbol frames per slowest capture frame rate ($\frac{N}{N+0.5}$). To rapidly decode, we set N=2, amounting to 3.6, 3.4, and 3.3 bits per block at the frame rate of the slowest receiver for red, green, and blue channels.

3.2.4 Multi-symbol Data Loading

Our channel levels enable 390 symbol pairs, allowing for 8-bit transmission over two symbols. We select only the 256 pairs with the lowest summed L-values to use in Styrofoam. This eases binary data loading and deters overexposure from interfering with the decode. Hence, we send 1 byte per block every 5 transmitter frames, or 0.4 bytes (3.2 bits) per block at the slowest receiver rate.

3.2.5 Styrofoam Coding Transmission

Combining the design parameters, the (N=2) Styrofoam transmission scheme, shown in Figure 2, is summarized as follows: After transmitting a blank frame, we transmit a symbol s_k over two frames, followed by an s_{k+1}, followed by another blank frame. Therefore, the Styrofoam display transmits the sequence:

$$\{\text{blank}\}, s_0, s_0, s_1, s_1,$$
$$\{\text{blank}\}, s_2, s_2, s_3, s_3,$$
$$\ldots$$

As the symbol display time (twice the display frame period) must be greater than any supported capture period, this requires that the display transmit at twice the frame rate of the slowest camera. Thus, on each block, Styrofoam transmits symbols at 0.8 the frame rate of the slowest receiver.

This is a slower symbol rate than related schemes; LightSync [3] transmits 1 symbol per block at the frame rate of the slowest receiver. However, while their system is limited to 1-bit per symbol, our packing scheme provides a guaranteed exposure of each symbol so that color levels can be robustly determined, allowing Styrofoam to transmit multiple bits per block, ultimately delivering a superior transmission rate.

3.3 Unpacking Symbols from Camera Frames

Styrofoam unpacks a block's symbols by observing the timing blocks associated with its row. Pure captures will have either the odd or the even exposure timing blocks illuminated. Mixed captures will have both values active.

For pure capture blocks, Styrofoam checks that $\alpha_{jk} > 0.5$. If so, then Styrofoam divides it by α_{jk}, quantizes to the color level, and stores the symbol in the reception stream.

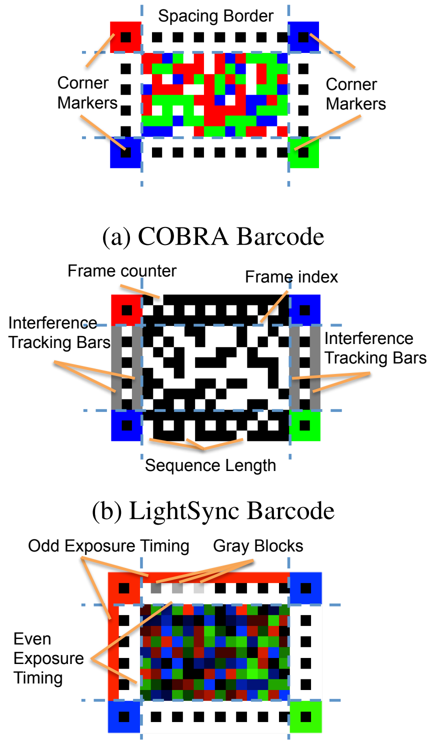
For mixed capture blocks, if the resolved pure symbol is unavailable, we delay processing the frame until the pure symbol is captured, which is guaranteed to be the next receiver frame. When the mixed symbol is resolved, we use Equation 5 to decode and quantize the target symbol.

The unpacking is thus very lightweight, as at most 3 arithmetic operations are required to decode a block. Furthermore, for unresolved mixtures, the unpacking only needs to store a block and its timing values for one frame. This minimizes the memory overhead as well as the latency of the decode. Each symbol is decoded within two transmit frames.

4. EVALUATION AND COMPARISONS

To characterize the practicality of the coding scheme, we decode streams transmitted and received on commodity smartphones. To receive, we sample camera frames at 30 fps on a Nokia Lumia 1020. To transmit codes, we use a Blackberry Z30, transmitting a 60 fps video generated by MATLAB. To compare against LightSync and COBRA works, we use a 36x20 code block grid for 720 code blocks. We analyze our captured codes offline with MATLAB. We ensured the full range of our transmitter was visible by manually setting ISO and exposure. This setup decodes with 100% accuracy.

We transmit using a 60 fps display and receiving with a 30 fps camera. Sending 720 blocks per frame and transmitting 8



(a) COBRA Barcode

(b) LightSync Barcode

(c) Styrofoam Barcode

Figure 3: Styrofoam barcodes compared to related works' barcodes. Styrofoam slightly adapts the COBRA tracking markers, using the first and third rows and columns to denote the per-row symbol exposure times.

bits per 5 transmitter frames yields an effective transmission rate of 69.1 kbps and a decode latency of 30 ms.

To support capture rates as low as 15 fps, Styrofoam can reduce the display frame rate to 30 fps, reducing its transmission rate to 34.6 kbps. This is 2.9x faster than LightSync's cited 12 kbps under identical conditions, while being able to begin decoding a stream with only 60 ms latency.

4.1 Related work: alternative coding schemes

We survey mobile camera-based VLC, with attention to impact of inter-symbol interference in the coding schemes.

COBRA [2] enables streaming color barcodes between smartphones with robustness to image blur from movement. Leveraging the transmitter accelerometer, COBRA adaptively sacrifices rate to gain robustness against blur. They also reorder colors in the barcode, clustering similar colors to further protect against blur. To mitigate inter-frame interference, COBRA slows the transmit rate to half the capture rate, guaranteeing multiple captures of a symbol. It then uses a blur-related metric to decide which capture to decode. Styrofoam addresses interference without a specific barcode design, thus allowing for compatibility with COBRA color reordering. Also, our symbols use intensities in each color channel for 32 symbol values, while COBRA uses five colors (black, white, red, green and blue).

LightSync [3] is a screen-to-camera streaming barcode system that transmits a fixed sequence at a maximum frame rate that is twice that of the slowest receiver's frame rate. They use four vertical tracking bars on the code to track inter-symbol interference. They append extra "erasure code" frames to allow the receiver to decode interfered symbols. This requires N extra frames to decode a code sequence of N frames. They cease reception once a sequence can be decoded, allowing faster decode for fast receivers.

However, considering the entire frame sequence of codes and extra coding frames, the LightSync effective code transmission rate is equal to the rate of the slowest receiver. Styrofoam is slower, transmitting codes at 0.8 times the rate of the slowest receiver. However, LightSync limits transmissions to be either black or white, thus a 1-bit code block, while Styrofoam enables symbols to be encoded on three dimensions of color level values, thus higher bit code blocks. Finally, the introduction of a blank frame allows Styrofoam to have a much shorter decoding latency.

Unsynchronized 4D Barcodes [5] repeat symbols within a color channel to ensure that a code is received. Codes are transmitted 3 times in each color channel with the transmission frame rate matching the receiver frame rate. This ensures that each symbol will be received over a full exposure of the frame. They send different symbols over each color channel, giving an effective symbol rate that matches receiver rate. However, the use of colors in the repetition scheme precludes the use of colored symbols for transmissions. They constrain block symbols to 1-bit.

5. DISCUSSION

In this work, our contribution is limited to symbol scheduling to recover from interference. Our scheme leaves open avenues of research to enhance Styrofoam's transmission.

Code registration in time and space: For this iteration of Styrofoam, we assume the adoption of previous works' spatial timing blocks and corner markers to register our code. However, we note that the spatial timing reduces a 42x26 grid block by 34% to a final code space of a 36x20 grid of blocks. We will consider improvement by minimizing sacrificed code space in registering codes in time and space.

Color Coding: We designed Styrofoam color levels with device-dependent noise measurements. We should instead characterize a noise model across a range of devices. To optimize the communication scheme, a generalizable investigation should tightly characterize the noise and linearity in the transmission and reception in a multi-color channel. For example, while the blue channel receives interference from other color channels, adequate coding could ensure that the blue channel could transmit at higher depth when the interference from other channels can be removed.

Steganography: Steganography hides codes in images or videos in a way that does not deter from the user experience. Steganography pairs well with other VLC works to transmit information alongside other media [4, 9]. We plan to consider how Styrofoam codes could hide in video streams.

Real-Time Computation Evaluation: In this work, we simulate the decode operations by performing them offline from smartphone-captured data. Transferring our system to a mobile platform will allow us to investigate systems issues, including computation and memory complexity of our decode operations, and energy efficiency tradeoffs with regards to the image capture [6]. We speculate that the system overhead is low, as the arithmetic recovery operations are simple and unresolved frames only need to be held for one frame.

Fine Camera Control: For our experiments, we manually forced the exposure and ISO settings of the camera to capture a full range of light for the barcode. Future implementations should automatically set parameters to capture the range of intensities before decoding the transmission.

6. CONCLUSION

We design Styrofoam, a camera-based VLC coding scheme that ensures that symbols can be decoded despite inter-symbol interference due to lack of synchronization. We do this by inserting a short blank frame in our transmission pattern. Our scheme allows for code symbols to be represented by multi-bit R, G, or B values, leading to high code density. This results in a 2.9x faster data transmission rate than peer algorithms, such as LightSync or Unsynchronized 4D Barcodes.

REFERENCES

[1] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet–sRGB. In *Color and Imaging Conference*, number 1, pages 238–245. Society for Imaging Science and Technology, 1996.

[2] T. Hao, R. Zhou, and G. Xing. Cobra: color barcode streaming for smartphone systems. In *Proc. Mobisys*, pages 85–98. ACM, 2012.

[3] W. Hu, H. Gu, and Q. Pu. Lightsync: unsynchronized visual communication over screen-camera links. In *Proc. Mobicom*, pages 15–26. ACM, 2013.

[4] W. Huang and W. H. Mow. Picode: 2d barcode with embedded picture and vicode: 3d barcode with embedded video. In *Proc. Mobicom*, pages 139–142. ACM, 2013.

[5] T. Langlotz and O. Bimber. Unsynchronized 4d barcodes. *Advances in Visual Computing*, pages 363–374, 2007.

[6] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proc. MobiSys*. ACM, 2013.

[7] D. O'Brien, L. Zeng, H. Le-Minh, G. Faulkner, J. W. Walewski, and S. Randel. Visible light communications: Challenges and possibilities. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5. IEEE, 2008.

[8] S. Perli, N. Ahmed, and D. Katabi. Pixnet: interference-free wireless links using lcd-camera pairs. In *Proc. Mobicom*. ACM, 2010.

[9] D. Reilly, H. Chen, and G. Smolyn. Toward fluid, mobile and ubiquitous interaction with paper using recursive 2d barcodes. *Pervasive Mobile Interaction Devices*, 2007.