SoftLight: Adaptive Visible Light Communication over Screen-Camera Links

Wan Du, Jansen Christian Liando, and Mo Li
School of Computer Engineering, Nanyang Technological University, Singapore
{duwan, cjansen, limo}@ntu.edu.sg

***Abstract*—Screen-camera links for Visible Light Communication (VLC) are diverse, as the link quality varies according to many factors, such as ambient light and camera's performance. This paper presents SoftLight, a channel coding approach that considers the unique channel characteristics of VLC links and automatically adapts the transmission data rate to the link qualities of various scenarios. SoftLight incorporates two new ideas: (1) an expanded color modulation interface that provides soft hint about its confidence in each demodulated bit and establishes a bit-level VLC erasure channel, and (2) a rateless coding scheme that achieves bit-level rateless transmissions with low computation complexity and tolerates the false positive of bits provided by the soft hint enabled erasure channel. SoftLight is orthogonal to the visual coding schemes and can be applied atop any barcode layouts. We implement SoftLight on Android smartphones and evaluate its performance under a variety of environments. The experiment results show that SoftLight can correctly transmit a 22-KByte photo between two smartphones within 0.6 second and improves the average goodput of the state-of-the-art screen-camera VLC solution by 2.2×.**

I. Introduction

Visible Light Communication (VLC) over screen-camera links has become a promising technique for short-range communication on smartphones [12], [26]. When the sender screen displays a stream of barcodes, other camera-equipped devices can capture the frames and decode the information modulated by the colors of barcode symbols. Compared with conventional wireless Radio Frequency (RF) communication (e.g., cellular, bluetooth and WiFi), VLC provides unique advantages in security, spatial reuse and ease of usage, since the light propagation has strict direction and the communication normally happens between two users of short distance. Recent VLC studies focus on the throughput improvement of a single link by barcode designs and most of them consider relatively stable link quality [10], [11], [16], [18], [21], [27], [29]. The screen-camera links in practice, however, are highly diverse. The link quality varies according to many factors, including ambient light, camera's performance, relative distance and angles between the screen and camera, trembling of user hands, etc. VLC systems need to adapt to the link diversity of different scenarios. For example, different users may use different smartphone models or be in different environments (e.g., sitting statically or on moving buses).

Without efficient adaptation, high throughput (the correctly-decoded physical-layer bits per unit of time) may not result in high goodput (the correctly-recovered application-layer bits per unit of time), as a majority of physical-layer bits counted in the throughput may be duplicate and not able to contribute to the recovery of original data. A major challenge for providing adaptive VLC is that the screen-camera links have no feedback channel. The receiver cannot display any feedback information on the screen to the sender, since its screen must face to the user and display the captured frame for camera adjustment (e.g., zoom in/out, or adjust the phone orientation). As a result, conventional rate adaptation and retransmission schemes, like Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ), that rely on the feedback of channel conditions to recover from bit loss or errors are not suitable for the screen-camera VLC.

This paper presents SoftLight, a channel coding approach that provides adaptive and high-goodput VLC over screen-camera links. It leverages rateless coding to convert the original data into a stream of encoded bits. When a sender is displaying the barcode frames composed of rateless bits, receivers extract information from the captured frames, even those with error bits. Since every encoded bit contains innovative information of the original data, every correctly-received bit is useful for the link goodput. Once a receiver accumulates sufficient amount of clean bits, it can recover the original data by rateless decoding. Therefore, using SoftLight, a sender is able to automatically adapt the data rate to the dynamic channels of different link qualities.

Although a large body of rateless coding works have recently been done for wireless RF networks, they cannot be applied to screen-camera VLC. Conventional rateless erasure codes, like Luby Transform (LT) [19] and Raptor [23], are based on blocks of bits. The transmitted blocks (in packets) are assumed either correctly received or totally lost. A critical factor that affects the transmission performance is block size. If it is set too large, more blocks may be discarded even though they contain a large proportion of clean bits; otherwise, more overhead (e.g., checksum and error correction for every block) is required. In RF communication, block size is timely adjusted according to the channel coherence [3], [7], [14]. For screen-camera links, however, as we will demonstrate in Section II, it is difficult to set an appropriate block size due to the lack of channel coherence or feedbacks from receivers. Rateless codes for AWGN (Additive White Gaussian Noise) channels, like Spinal codes [22] and soft decoding [2], avoid the block size setting by bit-level coding. Their decoding process, however, involves intensive floating-point iteration operations which are computationally infeasible for VLC on smartphones [13].

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

Unlike conventional rateless codes, SoftLight achieves *light-weight rateless coding* on a *bit-level erasure channel*. We extract soft hint from the color of every received symbol to assess how likely each bit is correctly decoded during color demodulation. An erasure channel is established by discarding some low-confidence bits. Light-weight rateless erasure coding can be implemented over the bit-level erasure channel.

SoftLight incorporates a set of color modulation techniques to enable accurate soft hint estimation and establish a bit-level erasure channel with minimized false positive (the ratio between the wrongly-reserved bits and the total received bits) and false negative (the ratio between the wrongly-erased bits and the total received bits). In particular, SoftLight modulates bit frames with unique color components for better independence between the bits in the same symbol. It also exploits the independent components in YUV color space to reduce the color interference between adjacent symbols and adopts a special color palette for better color reservation.

SoftLight further adopts a new light-weight bit-level rateless coding scheme that tolerates the false positive of the bits provided by the erasure channel. SoftLight encodes the data frames based on a systematic rateless code and decodes the received frames by XOR operations at frame level. The uncertain bit positions are taken into account, which enables bit-level rateless coding. A majority vote decoding algorithm is used to eliminate the impact of false bits and guarantee the coding efficiency with light computation complexity.

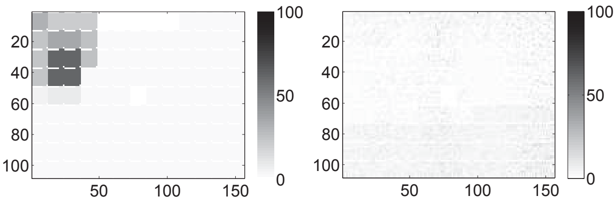
SoftLight does not impose any requirements on the barcode layout and is thus orthogonal to existing visual coding designs. We implement SoftLight atop two state-of-the-art VLC barcode layouts on Android smartphones of different phone models. The experiment results suggest that SoftLight can correctly transmit a 22-KByte photo between two smartphones within 0.6 second and achieve the maximum goodput of 317.3 kbps. In practical scenarios, on average, SoftLight provides 2.2× and 10.3× goodput improvement over RDCode [26] and COBRA [10] respectively.

II. Motivation

In this section, we study the diversity of screen-camera links by practical experiments and motivate the need of a new rateless coding scheme for VLC.

A. Diversity of screen-camera links

The screen-camera link diversity could be caused by many factors, such as the diverse camera performance of different smartphones and the spatiotemporal variation of environments. The experiments in [11] show that different cameras of current smartphones produce dramatically-varied performance for VLC transmissions. The authors in [10] demonstrate that due to the trembling of user hands, the transmission error rate augments when the acceleration of the receiver phone increases. In this section, we further study the impact of spatially-variant environments. We conduct experiments with RDCode [26], the latest high-throughput barcode design. We measure Bit Error Rate (BER), i.e., the ratio between the incorrectly-decoded bits and the total received bits.



(a) Outdoor. **(b)** Indoor.

Fig. 1: BER (%) of all symbol positions within the frame.

Environment diversity and intra-frame diversity. Figure 1 displays the BER experienced by the symbols at different positions of the barcode frame in indoor and outdoor environment. For outdoor environment, the BER is close to be uniform across all positions, probably because the sunlight interference is even. Some blocks experience high BER due to the blur effect caused by lens distortion or color inaccuracy of the screen or cameras. For indoor environment, however, non-uniform BER distribution is observed, which may be caused by the uneven interference from multiple fluorescent lights. The results suggest diversity among data bits in a same frame.

B. Why new rateless coding for VLC?

An adaptive VLC system is needed to work with varied link qualities and recover data from dynamic bit loss or errors. Rateless codes automatically adapt the transmission data rate according to channel conditions. Current rateless codes can be divided into two categories according to the channel model.

Rateless codes for erasure channels. Rateless erasure codes (e.g., LT [19] and Raptor [23]) work on block-level erasure channels in RF networks. Block-level LT code and Raptor code are used to enable fast data dissemination in wireless sensor networks [4] and improve the broadcast efficiency of led-camera communication [9] respectively. On screen-camera links, however, it is difficult to build an efficient erasure channel. VLC displays a barcode frame containing more than 32 kbits of data at one time. According to Figure 1, different positions on the frame experience varied BER which changes dramatically under different scenarios. It is hard to find a block size that efficiently extracts the correctly-received bits in one frame for different link qualities. Some FEC parity check bits may be added in each block [26] to enhance the block reception. The prefixed error correction capacity (sometimes also referred as coding rate) of FEC codes, nevertheless, cannot adapt to diverse link qualities.

Rateless codes for AWGN channels. Recent rateless codes, like Strider [5] and Spinal code [22], enable bit-level coding and approach the link capacity on AWGN channels, but the intensive decoding computation usually requires customized hardware and cannot be implemented on smartphones. Rateless erasure codes is also extended to AWGN channels by soft Belief Propagation (BP) decoding [2], which calculates the conditional probability of recovering one original bit based on the probability of some correctly-received bits. In VLC on

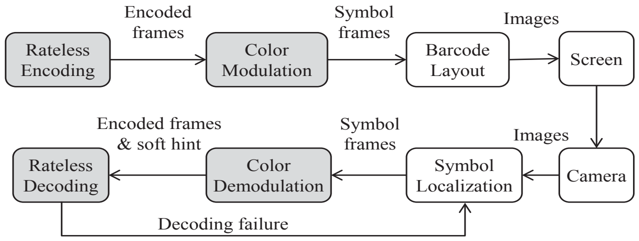


Fig. 2: System architecture of SoftLight. The main modules of SoftLight are highlighted in gray.

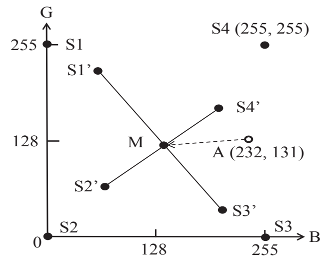


Fig. 3: Soft hint calculated for one symbol.

smartphones, a frame can be up to 32 kbits and the complex floating-point computation for updating the conditional probability of original bits needs to be performed 32k times. The intensive floating-point iteration operations of soft decoding cannot be completed on smartphones within the interval of two frames (e.g., normally less than 100 ms).

III. SoftLight design

In this section, we introduce the principle of SoftLight and describe its key modules in detail.

A. Overview of SoftLight

Figure 2 depicts SoftLight's architecture that is mainly composed of three modules, including rateless coding, special color modulation and soft hint extraction. The original data is first divided into a certain number of data frames that are then encoded into a series of rateless frames by *SoftLight rateless code*. The encoded data frames are further assigned with unique colors and modulated into a series of symbol frames. *Some unique color modulation techniques* are designed to mitigate the color interference between adjacent symbols. Finally, the symbol frames are rendered and displayed on the screen according to a specific barcode layout. As a channel coding scheme, SoftLight works atop all general barcode layouts, e.g., QR Code [1], COBRA [10] and RDCode [26]. The size of a data frame is determined by the number of symbols in one barcode frame.

At the receiver side, the symbols in the captured image frames are first localized by the barcode processing approach. The *color demodulation module* then extracts the data bits from the color of received symbols and also derives a *soft hint* for each bit from its color. Benefitting from the *unique color modulation techniques*, most of the wrongly-decoded bits can be identified according to their soft hint. A few of the error bits are still misclassified as correct bits (false positive). *SoftLight rateless module* decodes the received frames in consideration of the false positive problem and the computation complexity. If the decoding fails, the receiver adds more newly-received frames into the decoding incrementally. The original data is finally recovered when sufficient number of clean bits are received. The number of required frames is adapting to the channel quality.

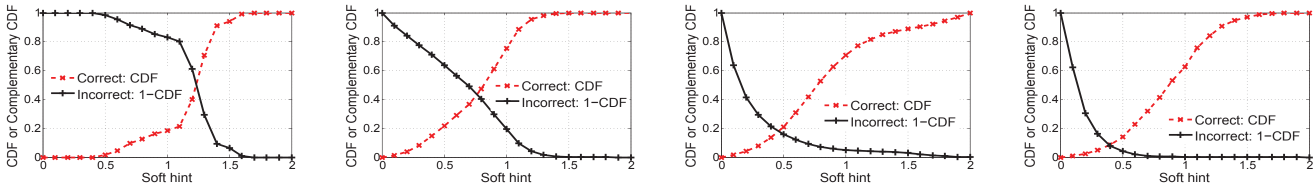
B. Soft hint

Current VLC systems [10], [16], [26] normally use 4 colors to modulate a symbol of two bits. When a barcode frame is captured, the data is recovered by comparing the color of each symbol with the standard colors. Figure 3 illustrates an example of one received symbol (i.e., point A) and four standard colors (i.e., S1$\sim$S4). A color palette of four symbols [26] may be inserted at some locations of the barcode frame to present the 4 standard colors that may be distorted to some other color values (e.g., S1'$\sim$S4') after transmission. Using the reserved standard colors in color palettes, the channel offset caused by some uniform interferences, like the color inaccuracy of the screen or cameras, can be eliminated. We will use the reserved standard colors in our calculation of soft hint.

The 4-color modulation is similar to QPSK of conventional RF communication. The color map of Figure 3 is comparable to the constellation map of QPSK. The soft hint of QPSK can be interpreted as the distance in signal space between the received symbol's constellation point and the decoded symbol's constellation point [15]. Similarly, we can estimate the soft hint of VLC links by the distance in color space between the received symbol color and the closest reserved standard color. The difference between the RF signal space and the color space is that the locations of standard symbols are fixed in RF communication but the reserved standard colors may change for different transmissions. Even in one transmission, four standard colors may experience different distortions. In Figure 3, four reserved standard colors (i.e., S1'$\sim$S4') have different distances to their middle point M.

If we refer the color distance of one symbol as the distance from that symbol to the middle point M of the reserved standard colors, *the soft hint of one received symbol is calculated as the color distance of that symbol normalized by the color distance of the closest standard color.* By normalization, the calculated soft hint is a relative value that is uniform for all standard colors across different transmissions. A low soft hint means that the color of the received symbol is close to M and the color is severely distorted. In contrast, when the soft hint is high, it is highly possible that the interference is low and the symbol is correctly demodulated.

An erasure channel can thus be established by discarding the symbols with a soft hint lower than a threshold τ. To evaluate the feasibility of such a soft hint enabled erasure channel, we conduct an experiment in a typical indoor environment with normal fluorescent light. We transmit 200 barcode frames from one smartphone to another using the RDCode layout. Figure 4a depicts the distributions of the soft hints calculated for all received symbols, which are separated by whether they



(a) Symbol level (BER: 4.5%; τ: 1.35; False positive: 1.0%; False negative: 76.4%).

(b) Bit level (BER: 2.8%; τ: 0.8; False positive: 1.0%; False negative: 58.3%).

(c) YUV color space (BER: 2.3%; τ: 0.2; False positive: 0.9%; False negative: 4.9%).

(d) Special color palette (BER: 2.2%; τ: 0.15; False positive: 0.8%; False negative: 1.8%).

Fig. 4: Distributions of soft hint for the correctly and incorrectly demodulated bits with the proposed techniques.

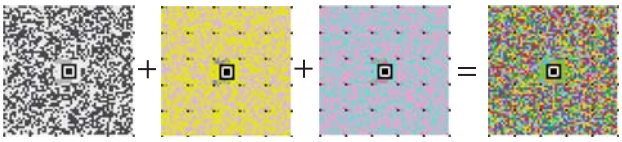


Fig. 5: Bit-level color modulation in YUV color space.

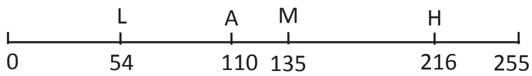


Fig. 6: Per-bit soft hint calculation.

are correctly or incorrectly demodulated. In Figure 4a, it is impossible to find a threshold that eliminates both the false positive and the false negative at the same time. With a small τ, the false negative is small, but the false positive is large. In contrast, with a large τ, the false positive is small, but the false negative is large. Since the wrongly-discarded symbols (false negative) only affect the efficiency of the erasure channel but the wrongly-reserved symbols (false positive) determine the correctness of the erasure channel, we set τ to a value that makes the false positive around 1%. In Figure 4a, when τ is 1.35, the false positive is 1.0% (considering that the overall BER is 4.5%) and the false negative is 76.4%.

Such high false negative prohibits the usage of derived soft hint to establish an effective erasure channel. The inaccurate estimation of soft hint is mainly due to the inherent intra-frame color interferences of current VLC color modulation.

C. Special color modulation

We modify the VLC color modulation and devise a set of techniques to achieve more accurate soft hint estimation.

Bit-level color modulation. Instead of assigning each possible value of one symbol with a unique color, our bit-level color modulation scheme allocates each bit of one symbol with an individual color component according to its position in the symbol. As shown in Figure 5, three bit frames are assigned with the three color components and are combined bit-wisely to construct a symbol frame. The color of one symbol, represented by the values (i.e., 0 or 255) of three color components, is determined by the value (i.e., 0 or 1) of the bits at the same position of the three bit frames.

Figure 6 presents an example of one received bit and two standard values in one color component. The color distance of one bit refers to the distance from the received color (e.g., point A) to the middle point M of the standard L (Low) and H (High) color values. By normalizing the color distance of one bit with the color distance of the standard colors, we obtain the soft hint of individual bits, calculated as follows.

$$SH_a = \frac{|C_a - (C_H + C_L)/2|}{(C_H - C_L)/2} \tag{1}$$

where C_a presents the color component value of point A. Different color components may have different color distances, e.g., the G and B color components in the RGB color space have different color distance to the middle point M in Figure 3. Based on the normalization in Equation 1, soft hint is a relative value that is uniform for different color components. Figure 4b plots the distributions of the soft hints calculated by the bit-level color modulation. The data is modulated in commonly-used RGB color space. The BER of the link is 2.8%. If we set τ to 0.8, the false positive and false negative are 1.0% and 58.3% respectively. While the bit-level soft hint is more accurate than the symbol level, its false negative is still high.

Independent color space. We further exploit the independence between color components to reduce their mutual interference. One color in RGB color space is composed of three components (red, green, and blue). All three components, however, contain the brightness ingredient that is sensitive to ambient light and may easily leak to the other color components of a same symbol or neighboring symbols due to blur effect. Instead of RGB, SoftLight modulates data in YUV color space. Brightness is extracted as an independent component Y (i.e., luma, the first color frame in Figure 5), and the other two chrominance components U and V (the second and third color frame in Figure 5) are also independent. By modulating bit frames with the independent color components of YUV color space, the interferences between different color components of one symbol or adjacent symbols are minimized. Now, the interference only occurs between the same color component of two adjacent symbols that have opposite values (i.e., 0 and 255). It can be accurately captured by soft hint.

Figure 4c presents the distributions of the soft hints calculated by the bit-level YUV-based color modulation. If we set τ to 0.2, the false positive and false negative are 0.9% and 4.9% respectively. The soft hint derived from the bit-level color modulation in YUV color space has much smaller false negative than in RGB color space.

Special color palette. We also devise a special color palette to better calibrate color distortions. Since SoftLight modulates bits independently in YUV color space, we only need to present the L and H values of each color component in color palettes. We use six symbols to form the color palette. The first three symbols are all set to the L value (i.e., "0,0,0") for all the three color components and the last three symbols are set to the H value (i.e., "255,255,255"). As a result, two middle symbols (the second and fourth symbols), representing the standard values of individual color components, are well protected by the adjacent symbols of same color and thus immune to the blur interference from the adjacent symbols.

Figure 4d shows the distributions of the soft hints calculated by the bit-level YUV-based color modulation with the special color palette. If we set τ to 0.15, the false positive is 0.8% and the false negative is 1.8%. Less than 1% of the bits are wrongly classified to correct bits for screen-camera VLC. Similar false positive rate is observed in Zigbee networks [15]. Such a small proportion of false positive and false negative allows us to establish an efficient bit-level erasure channel.

Discussions. All experiments in Figure 4 are conducted with the same setting, i.e., using the same phones and at the same location in an office. We will conduct more experiments in many other scenarios in Section IV-C to demonstrate that the soft hint extracted by our color modulation techniques is accurate for constructing a bit-level erasure channel and a constant τ can keep the false positive and false negative low in a variety of environments.

Like the previous works (e.g., RDCode and COBRA), one symbol in the barcode frame contains two bits. The chrominance components U and V are used to modulate two bits independently, since the Y color component is sensitive to the external interference like ambient light.

D. SoftLight rateless coding

Every bit provided by our color demodulation module could have one of three possible values, i.e., '0', '1' and 'x'. The value 'x' means that the bit is erased due to its low soft hint. The result of XORing 'x' with '0' or '1' is 'x'. Moreover, a few bits of '0' or '1' may be wrongly-demodulated but not be erased by the soft hint (false positive). Based on a bit-level erasure channel, a simple way to provide rateless coding is to encode individual bits as the blocks in conventional LT or Raptor codes. However, the binary BP decoding algorithm is developed for perfect erasure channels and cannot tolerate any false positive of the bits. The errors caused by one incorrect bit may propagate to many other original bits and jeopardize the whole decoding process.

SoftLight incorporates a new rateless coding scheme that considers the unique features of screen-camera VLC. It efficiently processes the 'x' bit and handles the false positive problem. At the same time, the SoftLight decoding algorithm has low computation complexity that avoids the coding process hindering the link goodput.

Encoding. SoftLight encodes data by a systematic rateless code at frame level. The sender segments the original data into a sequence of bit frames, which are first encoded by a FEC code, like Reed-Solomon (RS) code or Low-Density Parity-Check (LDPC) code, to generate some parity check frames for error correction. The intermediate frames (the original bit frames and the parity check frames) are further encoded to produce a stream of rateless frames, each of which is calculated by XORing a certain number of randomly chosen intermediate frames (the number is denoted as ρ). The encoded frames including both the intermediate frames and the XORed frames are sent to the TX buffer in sequence for transmitting.

The systematic design allows SoftLight to approach the channel capacity when the link quality is high, since the receiver can recover the original data by a few parity check frames without any rateless encoded frames. Moreover, the FEC parity checking enables the fast convergence of SoftLight decoding, which will be introduced in the decoding part.

In every transmitted frame, we insert the coding information on the number of original frames and the seed of random number generator, which allows receivers to reproduce the generation equations (coefficient matrix) of encoded frames.

Decoding. Upon the reception of its first frame, a receiver reproduces the generation equations, in which the number of encoded frames are normally $20\times$ larger than the number of original frames. It is able to adapt the data rate to be $20\times$ smaller than the bandwidth and is thus sufficient for a large range of link quality, even some extremely-lossy links. By solving the linear generation equations (the number of equations is $20\times$ larger than the number of variables), every intermediate frame has multiple instances, each of which is expressed as XORing several encoded frames.

When an encoded frame is received, it is plugged into the expression formula of the related instances. As more encoded frames are received, more instances are calculated. For one bit position, all calculated instances of an intermediate frame may have different results. If the bit in an encoded frame is 'x', the corresponding bit in the relative instances would be 'x'. If the bit in an encoded frame is erroneous, the corresponding bit in the relative instances will be wrong. A majority vote is performed to find the correct value of each bit. We record the occurrence frequency of '1' and '0' at every bit position in all calculated instances. The bits are set to the value with the highest occurrence frequency. Therefore, for every bit position, the error or 'x' bits in one or few instances can be overweighed by the other correct instances.

Figure 7 presents an example of SoftLight rateless decoding. Every frame contains four bits. The intermediate frames include two original frames (i.e., X1 and X2) and one FEC parity check frame (i.e., P1). Each intermediate frame has three instances that are expressed by XORing two encoded frames. When the encoded frame Y6 is received, all three instances are calculated. Some received frames contain several erased bits, e.g., the second and third bit in Y2, which cause the related bits in some instances to be 'x' (e.g., the second instance of X1 and the first instance of X2). Some bits in the received frames are erroneous but not erased by soft hint. They result in errors at the related bit positions of some instances, as highlighted

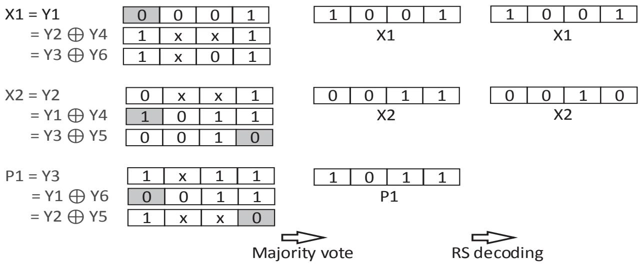


Fig. 7: SoftLight rateless decoding. The erased bits are presented as 'x' and the error bits are highlighted in gray.

in gray in Figure 7. By majority vote, the error and uncertain bits are corrected and all intermediate frames are recovered. The original frames X1 and X2 are finally recovered by parity checking. Upon the reception of every barcode frame, the above decoding process is performed. The transmission stops when the original data is correctly recovered.

Interleaving. Since different positions in one frame may experience varied link qualities due to the intra-frame diversity (as shown in Figure 1), it is possible that some positions in the frame have high BER and are difficult to be recovered. Therefore, in SoftLight, the symbols in one frame are interleaved before transmission, and the received frames are deinterleaved after demodulation. By interleaving, the bits at different positions experience a similar bit reception rate on average. All bit positions succeed in decoding almost at the same time. Even if few bits of the intermediate frames still have errors after rateless decoding, the original bit frames can be recovered by the parity check frame.

Discussions. To control the error propagation of false positive bits and provide a large volume of encoded frames., the parameter ρ is set to 3. At the receiver side, every instance of the intermediate frames is calculated by three received encoded frames. The error bits in all relative encoded frames may add in one instance. With a small ρ, we limit the total number of error bits in one instance. The limited error bits can be corrected by the other instances during majority vote. The decoding with a small ρ also involves less XOR operations and imposes light computation complexity. On the other hand, with $\rho=3$, the encoded rateless stream contains sufficiently large number of encoded frames (e.g., for 100 intermediate frames, 161700 encoded frames could be produced) that provides an effective data rate ranging from 0.2 kbps to 329.1 kbps. Our experiments in Section IV show that the setting of $\rho=3$ can efficiently control the impact of false positive bits and provide significant goodput gain in a variety of environments.

IV. Evaluation

We conduct a series of experiments to evaluate the performance of SoftLight in a variety of environments.

A. Implementation

We implement SoftLight on top of two typical barcode layouts, COBRA [10] and RDCode [26], with Android 4.2 platform. COBRA is the first VLC system running on smartphones. Based on it, many works [11], [18] have been developed. RDCode is the latest VLC design which supports the reception of partial frames. The source codes of COBRA and RDCode are from their authors. We modify the color modulation of COBRA and RDCode to add the bit frame modulation in YUV color space and the special color palette. We develop both online and offline decoders in C and C# respectively. The offline decoder takes video as input, and adopts the same rateless decoding engine as the online decoder.

Computation efficiency of decoding. SoftLight decoding includes de-interleaving, soft hint calculation, majority voting, and RS parity checking. The experiment results on LG Nexus 5 reveal that the first three components take a short time, i.e., 5.6ms for for the frame of 84*60 symbols (including 0.1ms for de-interleaving, 1.2ms for soft hint calculation, and 4.3ms for majority voting) and 17.1ms for for the frame of 156*108 symbols (0.4ms for de-interleaving, 4.0ms for soft hint calculation, and 12.7ms for majority voting). The RS parity checking in SoftLight is time consuming as it requires a large number of RS decoding operations. Every byte at the same position of all recovered intermediate frames forms a byte array. We perform RS decoding for every byte array independently. Although one RS decoding operation is generally less than 0.05ms, there are 544 byte arrays for the frame of 84*60 symbols and 1856 byte arrays for the frame of 156*108 symbols. However, we do not need to perform RS parity checking for all the byte arrays after the reception of every frame. When the decoding of one byte array succeeds, we exclude it from the subsequent computation. Thus, the required number of RS decoding operations decreases sharply as more frames are received. The experiment shows that SoftLight can complete all the computation without introducing any additional transmission overhead.

B. Experiment setting

Except the experiments investigating the performance of different smartphones, we use LG Nexus 5 for all experiments. It has a camera of 8M pixel resolution and 1080p@30FPS video capturing rate. The brightness of the screen is set to 50% of its maximum value. Unless otherwise stated, our experiments are conducted in an indoor environment with normal fluorescent light ($1 \sim 100$ lx). The distance between the sender and the receiver is about 20 cm. The sender and the receiver devices are kept still for most experiments. The frame rate of the sender is 10 FPS (the maximum frame rate the RDCode online encoder supports) and the capture rate of the receiver is 30 FPS. We also test the impact of these environment factors in some individual experiments, e.g., increasing communication distance and frame rate, and trembling of user hands .

Benchmarks. We compare SoftLight with COBRA and RDCode. COBRA does not apply any error correction in its default design. We add some RS parity check bytes in each frame for COBRA and two error correction levels are tested. RDCode adopts a three-tier error correction scheme.

TABLE I: The False Positive (FP) and False Negative (FP) of the bits filtered by different soft hint thresholds.

Average BER(%)	BER(%) Range	$\tau = 0.2$		$\tau = 0.5$		$\tau = 0.8$	
		FP(%)	FN(%)	FP	FN	FP	FN
0.6	0-10	0.1	2.98	0.007	12.9	0	39.7
14.5	10-20	5.8	12.8	1.0	34.3	0.29	51.9
24.6	20-30	12.3	12.8	2.6	30.9	0.2	46.7

The original data is divided into several frames, each of which includes a certain number of blocks. Every block contains n (i.e., 12*12) symbols, including k original bits and *(n-k)* RS redundant bits (that can correct *(n-k)/2* error bits). In each frame, p parity check blocks are added to recover from the lost blocks. All data frames followed by q parity check frames form a packet, which is transmitted repeatedly until the receiver successfully recovers the original data. For every experiment, different settings of RDCode parameters are tested and the best setting is used for evaluation.

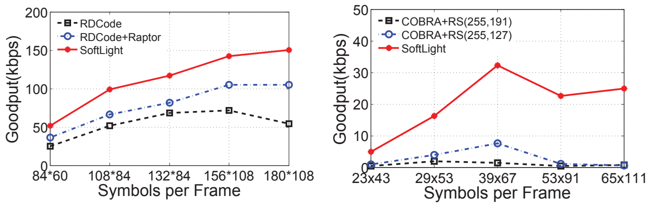
We also implement block-level Raptor coding on top of RDCode, denoted as RDCode+Raptor. The sender encodes blocks by Raptor code. The receiver recovers the original data by binary BP decoding algorithm when sufficient amount of clean blocks are received. The intra-block RS coding of RDCode is enabled to improve the block reception rate. Since encoded blocks are independent of each other, the loss of some blocks at any particular positions does not impact the recovery of the original data. Therefore, block-level interleaving does not improve the goodput of RDCode+Raptor.

C. Parameter setting of SoftLight

SoftLight has two parameters, the soft hint threshold and the size of RS parity checking used in the rateless coding, which may need to be set in different environments. Through a series of experiments, we demonstrate that these two parameters can be set as constant for a large range of link quality without impact to the adaptability of SoftLight.

Soft hint threshold. Figure 4 has shown that the soft hint threshold τ determines the tradeoff between the false positive and the false negative of the bits provided by the erasure channel. In this section, we show that a constant τ is sufficient to establish an efficient erasure channel in a variety of environments. We collect frame traces in our experiments of various scenarios, including indoor and outdoor environments.

Table I presents the false positive and false negative of the bits filtered by different soft hint thresholds under different link qualities. We classify the links into three quality levels. When τ increases, the false positive decreases and the false negative increases. With a moderate τ, both the false positive and false negative are small for a large range of link quality. Even for the lossy links with 20%-30% BER, the false positive is only 2.6% and the false negative is 30.9%. Although 30.9% bits are discarded, they do not cause further decoding overhead in SoftLight, since the transmitted bits in SoftLight are independent and the key determinant of successful decoding is the amount of correctly-received bits but not any particular bits.



(a) SoftLight and RDCode. **(b)** SoftLight and COBRA.

Fig. 8: Goodput with different symbol sizes.

In the following evaluation experiments, we will show that based on the constant τ (i.e., 0.5), SoftLight can significantly improve the goodput of VLC systems in all the environments.

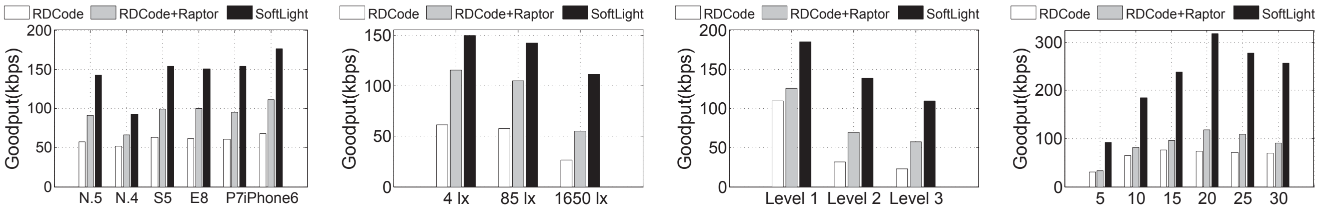
RS parity checking. In SoftLight, the error correction capacity of RS parity checking does not need to change for different environments, since SoftLight adapts to different link qualities by rateless coding but not RS parity checking. In both indoor and outdoor environments, we set the RS redundancy of SoftLight to different levels (i.e., 20%, 30%, 40%, and 50% of the original data). The deviation of the goodputs achieved in all experiments is within 5.6%. Therefore, we set the RS error correction capacity of SoftLight to 20%.

D. Goodput of a single receiver

We investigate the adaptability of SoftLight to dynamic channel conditions and in different environments. For every experiment, 10 tests are conducted and a photo of 22 KBytes is transmitted in each test. To guarantee the transmission accuracy, we measure the goodput when the VLC systems correctly recover the photo. It is necessary to use the VLC system to transmit photo, since it can deliver the complete information of the photo. If we display the photo directly on the sender's screen and capture the photo by the receiver's camera, the captured picture may lose some detailed information about the original photo due to camera constraint or blur effect.

Comparison with RDCode. Figure 8a depicts the goodput achieved by RDCode and SoftLight with different symbol sizes. When the number of symbols in one frame is large (the bandwidth is large), the symbol size is small and the link quality is low due to severe blur effect. The default parameter setting of RDCode is: *n-k=6*, *p=3*, and *q=1*. We have also tested three settings for RDCode and RDCode+Raptor with different capacities of intra-block error correction, i.e., *n-k=12*, *n-k=18* and *n-k=24*. For most symbol sizes, *n-k=6* is sufficient for error correction, except the last scenario of *180*108* symbols in which *n-k=12* performs the best. RDCode cannot automatically change the parameter setting for different links. We report the best result of these four settings for RDCode and RDCode+Raptor in Figure 8a.

Figure 8a reveals that the maximum goodput of SoftLight is 150.1 kbps when every frame contains 180*108 symbols. In contrast, the maximum goodput of RDCode and RDCode+Raptor is 68.2 kbps (symbols per frame: 132*84) and 105.1 kbps (symbols per frame: 180*108). SoftLight improves the goodput of RDCode and RDCode+Raptor by 2.2× and 1.4× respectively. Moreover, for every symbol



(a) Smartphones models. **(b)** Illuminance. **(c)** Trembling levels. **(d)** Frame rates (FPS)

Fig. 9: Adaptability of SoftLight and RDCode to different scenarios.

size, the goodput of SoftLight is higher than the other two approaches. SoftLight can best use every correctly-received bit with rateless coding. In RDCode, however, many blocks are discarded because their BER exceeds the capacity of intra-block RS coding, and some blocks are repeatedly received in multiple transmission rounds. Although the block-level rateless coding introduces the independence between blocks and mitigates the duplicate reception problem, due to the prefixed intra-block error correction, RDCode+Raptor still wastes many blocks that contain a large amount of clean bits.

Comparison with COBRA. Figure 8b presents the goodput achieved by COBRA and SoftLight with different symbol sizes. We add two different levels of error correction for COBRA, denoted as COBRA+RS(255,191) and COBRA+RS(255,127). In RS(255,191), 191 data bytes are appended with 64 RS parity check bytes, which can correct 32 error bytes. SoftLight improves the average goodput (for all symbol sizes) of COBRA+RS (255,191) by 20.5× and COBRA+RS(255,127) by 10.3× respectively. As the symbol count per frame increases, the goodput of SoftLight augments. The default symbol size of SoftLight is set to the smallest symbol size (65*111 symbol/frame), which provides the highest bandwidth. Due to the prefixed error correction of RS code, COBRA is sensitive to the parameter setting. Assuming we did a site survey and set COBRA to the best setting (39*67 symbol/frame and RS(255,127)), SoftLight still provides a goodput of 3.3× higher than COBRA.

According to the experiment results in Figure 8, the goodput of RDCode is much higher than COBRA. We will only present the results of RDCode in the following experiments.

Adaptability to the diversity of cameras. Figure 9a shows the goodput achieved by RDCode and SoftLight with different smartphones. Most of these smartphones have a camera of 8M pixel resolution and 1080p@30FPS video capturing rate. The camera of Apple iPhone6 has higher video capturing rate (60FPS), and Huawei Ascend P7 has higher pixel resolution (13M). With these new-released smartphones, SoftLight improves the average goodput of RDCode and RDCode+Raptor by 2.4× and 1.5× respectively. This experiment reveals that SoftLight is orthogonal to the hardware progress.

Adaptability to the diversity of environments. Figure 1 has demonstrated that in indoor environments, the BER of different positions in the barcode frame varies dramatically and no pattern can be found. The links in outdoor environments possess almost opposite characteristics. Therefore, the VLC systems must adapt to the unique link qualities of indoor and outdoor environments. Figure 9b plots the goodput achieved by RDCode and SoftLight in different environments, including indoor without light (4 lx illuminance), indoor with normal fluorescent light (85 lx illuminance) and outdoor with shadow (1650 lx illuminance). They are distinguished by the illuminance strength. SoftLight is able to automatically accommodate itself to different environments and improve RDCode and RDCode+Raptor by an average goodput gain of 2.8× and 1.5× respectively. The relatively-smooth performance of SoftLight in different environments comes from its adaptivity that achieves the best performance in different environments without any modifications of parameters.

Adaptability to the temporal diversity. We evaluate the adaptivity of the benchmark approaches to the temporal variation of link quality by trembling the receiver phone. We measure the trembling strength as the average acceleration value during the transmissions via the accelerometer on the phone. We have conducted a series of tests and the results are categorized by their trembling strength levels, i.e., Level 1 (0.2∼0.5m/s^2), Level 2 (0.5∼1.2m/s^2) and Level 3 (1.2∼1.9m/s^2). Figure 9c demonstrates the goodput achieved by RDCode and SoftLight under different trembling levels. For all trembling levels, on average, SoftLight achieves a goodput of 144.4 kbps that is 2.6× and 1.7× higher than RDCode and RDCode+Raptor respectively. In level 1, the relative-movement between the sender and the receiver is small. Benefitting from the systematic rateless coding, SoftLight corrects the small link loss efficiently and improves the performance of RDCode by 1.7×. In level 3, the receiver's phone is trembled slightly throughout the experiment, which causes severe channel variation. Based on the independence of rateless encoded bits, SoftLight can recover from any bit loss and achieves a goodput gain 5.1× higher than RDCode.

Adaptability to the diversity of frame rates. The CMOS-based cameras on current smartphones capture an image by scanning across the scene horizontally (i.e., rolling shutter). If the frame rate of the sender is high, some received frames may contain the contents of multiple displayed frames and the lines captured during the transition of displayed frames experience severe blur effect [11]. Figure 9d presents the goodput for different frame rates. The capture rate of the receiver, LG NEXUS 5, is 30 FPS. Higher frame rate means higher bandwidth but also higher blur interference due to rolling shutter. SoftLight achieves better goodput with higher frame rate (i.e., 20 FPS) than RDCode (i.e., 15 FPS). SoftLight can automatically recover from the bit loss of blurred symbols by accumulating more rateless encoded bits. The best goodput achieved by SoftLight is up to 317.3 kbps, which is even larger than the bandwidth of ZigBee networks (i.e., 250 kbps).

V. Related works

In this section, we study the previous works on screen-camera VLC and soft hint of conventional RF communication. Rateless codes have been analyzed in Section II-B.

Screen-Camera Visible Light Communication. Many barcodes, like HCCB [20] and QR code [1], are widely used in our daily life to retrieve information scanned by smartphones. COBRA [10] is the first VLC system running on smartphones. Its barcode design is optimized for blur resilience and light-weight computation. The synchronization problem between sender's frame rate and receiver's capture rate is addressed in [11], [16], [18]. Some recent works, like HiLight [17] and InFrame++ [25], embed information into images unobtrusively; however, the supported throughput is low. RDCode [26] develops a barcode layout to handle the problems of locality and partial availability. Current screen-camera VLC works focus on the barcode design for higher throughput. SoftLight targets at improving the adaptability of these works under a variety of link qualities by channel coding.

Soft hint of conventional RF communication. Prior works in information theory have discussed the application of PHY-layer soft information provided in RF decoding [6], [8]. SoftPHY [15] develops an interface that provides per-symbol soft hint in Zigbee PHY and retransmits the symbols with low confidence. SoftRate [24] adapts the data rate according to the soft hint of wireless PHYs, like 802.11a/b/g and WiMax. SOFT [28] recovers a packet from its instances received by multiple APs based on soft hint. FlexCast [2] encodes video data using Raptor code and decodes the encoded bits by soft BP decoding. SoftLight extends soft hint to color modulation of screen-camera links and applies the per-bit soft hint in a novel VLC rateless coding.

VI. Conclusion

This paper presents SoftLight, a channel coding design for screen-camera VLC systems. SoftLight exploits the confidence in each demodulated bit of VLC and develops a set of color modulation techniques to provide accurate soft hint estimation. SoftLight also devises a novel rateless coding scheme that tolerates the false positive of per-bit soft hints and imposes light computation complexity. The results of experiments on Android smartphones show that SoftLight is able to automatically adapt to different VLC environments and significantly improve the goodput over existing VLC systems.

Acknowledgments

We thank Hao Feng and Rentro Lui Jian Ting for their help on the experiments. We acknowledge the support from Singapore MOE AcRF grant MOE2012-T2-1-070, MOE2013-T1-002-005, and NTU NAP grant M4080738.020.

References

[1] Automatic identification and data capture techniques - qr code 2005 bar code symbology specification.

[2] S. Aditya and S. Katti. Flexcast: graceful wireless video streaming. In *ACM MobiCom*, pages 277–288, 2011.

[3] W. Du, Z. Li, J. C. Liando, and M. Li. From rateless to distanceless: Enabling sparse sensor network deployment in large areas. In *ACM SenSys*, pages 134–147, 2014.

[4] W. Du, J. C. Liando, H. Zhang, and M. Li. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *ACM SenSys*, pages 365–378, 2015.

[5] A. Gudipati and S. Katti. Strider: automatic rate adaptation and collision handling. In *ACM SIGCOMM*, pages 158–169, 2011.

[6] V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *IEEE FOCS*, pages 28–37, 1998.

[7] A. Hagedorn, S. Agarwal, D. Starobinski, and A. Trachtenberg. Rateless coding with feedback. In *IEEE INFOCOM*, pages 1791–1799, 2009.

[8] J. Hagenauer and P. Hoeher. A viterbi algorithm with soft-decision outputs and its applications. In *IEEE GLOBECOM*, pages 1680–1686, 1989.

[9] J. Hao, Y. Yang, and J. Luo. CeilingCast: Energy efficient and location-bound broadcast through led-camera communication. In *IEEE INFOCOM*, 2016.

[10] T. Hao, R. Zhou, and G. Xing. COBRA: color barcode streaming for smartphone systems. In *ACM MobiSys*, pages 85–98, 2012.

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

[12] W. Hu, J. Mao, Z. Huang, Y. Xue, J. She, K. Bian, and G. Shen. Strata: layered coding for scalable visual communication. In *ACM MobiCom*, pages 79–90, 2014.

[13] P. A. Iannucci, K. E. Fleming, J. Perry, H. Balakrishnan, and D. Shah. A hardware spinal decoder. In *ACM/IEEE ANCS*, pages 151–162, 2012.

[14] P. A. Iannucci, J. Perry, H. Balakrishnan, and D. Shah. No symbol left behind: a link-layer protocol for rateless codes. In *ACM MobiCom*, pages 17–28, 2012.

[15] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, pages 409–420, 2007.

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

[17] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou. Real-time screen-camera communication behind any scene. In *ACM MobiSys*, pages 197–211, 2015.

[18] R. LiKamWa, D. Ramirez, and J. Holloway. Styrofoam: A tightly packed coding scheme for camera-based visible light communication. In *the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, pages 27–32, 2014.

[19] M. Luby. LT codes. In *IEEE FOCS*, pages 271 – 280, 2002.

[20] D. Parikh and G. Jancke. Localization and segmentation of a 2D high capacity color barcode. In *IEEE Workshop on Applications of Computer Vision*, pages 1–6, 2008.

[21] S. D. Perli, N. Ahmed, and D. Katabi. PixNet: interference-free wireless links using lcd-camera pairs. In *ACM MobiCom*, pages 137–148, 2010.

[22] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah. Spinal codes. In *ACM SIGCOMM*, pages 49–60, 2012.

[23] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, 2006.

[24] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM*, pages 3–14, 2009.

[25] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng. Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication. In *ACM MobiSys*, pages 181–195, 2015.

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

[27] Q. Wang, M. Zhou, K. Ren, T. Lei, J. Li, and Z. Wang. Rain bar: Robust application-driven visual communication using color barcodes. In *IEEE ICDCS*, pages 537–546, 2015.

[28] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi. Beyond the bits: cooperative packet recovery using physical layer information. In *ACM MobiCom*, pages 147–158, 2007.

[29] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang. SBVLC: Secure barcode-based visible light communication for smartphones. In *IEEE INFOCOM*, pages 2661–2669, 2014.