Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Bufalino, Jacopo; Freire, Maria L.Montoya; Kannala, Juho; Di Francesco, Mario

MAMBA

Published in:
Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020

DOI:
10.1109/WoWMoM49955.2020.00059

Published: 01/08/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Bufalino, J., Freire, M. L. M., Kannala, J., & Di Francesco, M. (2020). MAMBA: Adaptive and Bi-directional Data Transfer for Reliable Camera-display Communication. In *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020* (pp. 307-316). Article 9217725 IEEE. https://doi.org/10.1109/WoWMoM49955.2020.00059

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

MAMBA: Adaptive and Bi-directional Data Transfer for Reliable Camera-display Communication

Jacopo Bufalino, Maria L. Montoya Freire, Juho Kannala, and Mario Di Francesco

Department of Computer Science, Aalto University

Abstract—**Camera-display communication uses visible light to transfer data wirelessly by using a screen as a transmitter and a camera as a receiver. Such an approach faces several challenges to be employed in practice, including unreliable decoding due to imperfect synchronization and channel impairments. This article introduces MAMBA, a mobile application framework for camera-display communications through adaptive and bi-dimensional color barcodes. MAMBA employs efficient computer vision techniques and scales with the number of blocks in the barcodes, rather than with the number of pixels in the captured image. As a consequence, it allows to take full advantage from the high-resolution cameras available on modern mobile devices. In addition, MAMBA realizes the first adaptive and bi-directional protocol for camera-display communications with fast feedback. Specifically, MAMBA jointly carries out dynamic adaptation of frame length based on both environmental conditions and the processing capabilities of the devices. Experimental results show that MAMBA is effective, thereby allowing reliable real-time communication in a variety of operating conditions.**

Index Terms—**Camera-display communication, color barcodes, smartphones, bi-directional data transfer, adaptive procol**

I. INTRODUCTION

Camera-display communication leverages visible light to exchange data wirelessly from a transmitting display to a receiving camera [1, 2]. Such a form of communication employs a machine-friendly representation of data in the visual domain, usually through bi-dimensional barcodes, for instance, Quick Response (QR) codes [3]. Instead of being printed on paper, these codes are shown on a screen or a display as a sequence of dynamically changing images [4]. As a result, camera-display communication enables novel applications and services involving device-to-device communication in different scenarios related to mobile and context-aware computing. For instance, it can be used as an alternative method for a self-payment system by using a smartwatch and a display. Indeed, we can leverage the availability of displays and cameras in current off-the-shelf devices to provide a more secure transmission when exchanging sensitive information (e.g., credit card numbers). This work targets an scenario of bi-directional camera-display communication involving two smartphones.

Furthermore, camera-display communication has several advantages over traditional radio-based wireless technologies. For instance, the visible light channel is more robust to interference due to noise and simultaneous access. As a consequence, it can be used when the radio spectrum is scarce, highly congested, or very unreliable (e.g., in industrial settings [5]). However, camera-display communication also faces several challenges in practice. In particular, decoding reliability is one of the most critical factors [6, 7]. First and foremost, there are fundamental constraints between the chance of successful decoding and environmental or operating parameters: the size of the code at the sender, channel conditions, the camera resolution at the receiver, and its field of view [2]. The primary cause of this issue is determined by *path loss* [8]. A solution to overcome this loss is to increase the frame resolution (e.g., from 1080p to 4K resolution), but at the cost of a higher computational time. Another crucial aspect is also represented by synchronization, which is hard to achieve in practice due to both hardware and software constraints [9]. Unidirectional communication scenarios are particularly critical in this regard, since the capture as well as display rates cannot be agreed upon beforehand. These issues have motivated the development of barcodes that can tolerate frame loss due to lack of synchronization [9, 10] and (or) operate on the frequency domain for increased reliability [8, 11]. However, these solutions incur in a substantial overhead that may prevent capturing images at high resolution in real-time, thereby not taking full advantage from modern embedded systems hardware.

This article proposes MAMBA – a framework for Medium-Aware Mobile Barcode Adaptation – to address these challenges. MAMBA consists of a custom color barcode, together with an efficient decoding pipeline based on computer vision techniques and suitable for modern mobile device hardware. Different from the state of the art, MAMBA leverages *bi-directional* camera-display communication and *adaptive* transmission to achieve *reliable* data transfer in spite of imperfect synchronization and corrupted or lost frames. In particular, the adaptive transmission scheme in MAMBA is able to quickly react to channel impairments and variations in the rate of the sender by dynamically estimating frame losses and recovering from errors through fast receiver feedback.

This work establishes the following contributions. First, MAMBA employs efficient computer vision techniques to scale with the number of *blocks* in a barcode, rather than with the number of *pixels* in the captured image. As a consequence, MAMBA allows to take full advantage from the high-resolution cameras available on modern mobile devices. Furthermore, MAMBA is the first solution – to the best of the authors' knowledge – employing *adaptive bi-directional communication* for reliable data transmission over screen-camera links. Such an approach overcomes message loss due to imperfect synchronization and channel impairments through dynamic setting of transmission parameters, as opposed to the existing solutions that are static. Finally, experimental results demonstrate that MAMBA is able to decode images beyond full HD resolution on off-the-shelf smartphones in *real-time*. Moreover, its adaptive communication protocol is *effective* in tailoring the transmission parameters to the environment, thereby allowing reliable communication in a variety of operating conditions.

II. RELATED WORK

Several approaches for visible-light communication have been proposed in the literature [1, 12]. The following focuses on camera-display communication, as those are the most relevant for the solution proposed here. The section concludes with a comparison of MAMBA against the state of the art.

Bi-dimensional barcodes. Bi-dimensional (or 2D) barcodes encode data into a collection of blocks arranged onto a rectangular (or sometimes square) surface. For instance, Quick Response (QR) codes are widely used in the context of mobile and pervasive computing [3]. Their popularity has triggered the development of several solutions. For instance, merging an image with a QR code to encode information about the image itself, while preserving the image appearance [13, 14].

Several barcode designs have been explicitly proposed for camera-display communication as a sequence of barcodes over time. Among them, COBRA is one of the most well-known solutions targeting smartphones [15]. COBRA employs multiple colors to increase capacity and includes an adaptive mechanism to mitigate motion blur at the sender. RDCode [15] aims at increasing reliability of communication by applying error correction over a layered code structure, wherein frames are divided into macro-blocks in addition to regular blocks.

Other works have specifically addressed synchronization issues in camera-display communications. For instance, Zhan and Tong [16] employ inverse symbols shape and color shifting to deal with inter-frame blur caused by camera-display misalignment. Langlotz and Bimber [4] leveraged colors as multiple channels over which codes are repeated multiple times. Instead, LightSync [9] inserts special inter-frame erasure codes that are robust to frame mixing. Similarly, Styrofoam [17] explicitly adds blank frames among encoded data so as to shape multiple code exposures that occur in mixed frames. Moreover, Strata [10] employs hierarchical encoding in both the spatial and the temporal domains to seamlessly support receivers with heterogeneous capture rates and channel conditions.

Encoding data in the frequency domain. Data can also be encoded in the spectrum of a bi-dimensional code [18], instead of on its spatial layout with the approaches discussed above. In this context, PixNet [11] pioneered the use of Orthogonal Frequency Division Multiplexing (OFDM) for camera-display communication involving mobile devices. In particular, PixNet divides data into OFDM codes and leverages perspective distortion correction for decoding purposes. FOCUS [8] also builds on OFDM to define a layered code that can accommodate receivers with heterogeneous characteristics, similar to Strata. Both PixNet and FOCUS have been designed to target senders represented by large screen or displays, thereby allowing to show codes with large enough size (e.g., 100 square centimeters). They trade-off capacity and reliability in different ways to extend the communication range to several meters.

Bi-directional camera-display communication. All solutions mentioned so far target unidirectional screen-camera links. Indeed, camera-display communication have been applied to bi-directional scenarios too. Among them, CamTalk [19] was the first solution to securely exchange data between two smartphones. CamTalk leverages QR codes and supports a session-oriented communication scheme, inspired by the Transmission Control Protocol (TCP). Data in a session are retransmitted to ensure reliable data delivery at both ends. Montoya and Di Francesco [7] devised a similar solution for casual file exchange between two users through smartphones. The communication protocol defined therein is also based on QR codes and includes a selective retransmission scheme that improves communication efficiency.

Codes embedded into video streams. The works surveyed above encode data in a visible code that is shown on a screen or display [2]. A different option, instead, consists in embedding a data stream into visual content – generally, a video – in such a way that it is invisible to end users. HiLight [20] follows this approach by encoding data into the alpha channel that denotes the transparency value associated to individual pixels of an image, without the need to modify the color components of the source video streams [21]. InFrame++ [22] leverages complementary frames that are interdependent and displayed back-to-back. The physiological features of human eyes "blends" these frames, thereby making the encoded content unnoticeable. ImplicitCode [23] uses a similar approach but a different scheme to achieve smooth color transitions in the source video stream.

MAMBA versus the state of the art. While prior work has explored solutions including barcodes design and methods for data encoding, there is not much work about barcode solutions that are adaptive and scalable in the domain of the image quality. MAMBA leverages bi-dimensional barcodes, similar to COBRA [15] and the other solutions of the same class [6, 9, 17], but includes several features to improve the communication performance and to support adaptivity. MAMBA employs computer vision-based algorithms that scale decoding with the number of blocks in the source image as opposed to the pixels therein. This allows to take advantage from high-resolution cameras available on off-the-shelf mobile hardware to increase barcode capacity. These features make MAMBA the first barcode suited for real time high-quality VLC. In contrast with Strata [10] and FOCUS [8], MAMBA addresses synchronization through

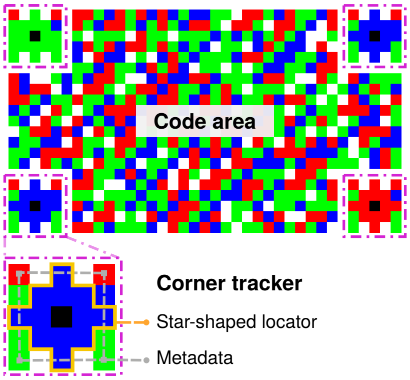


Fig. 1: A MAMBA barcode includes: corner trackers delimiting the barcode and specifying its orientation; a code area containing the source (raw) data.

frame rate and length adaptation, instead of using layered data encoding. Furthermore, MAMBA has a low decoding overhead because it does not have to perform the fast Fourier transform as solutions operating on the frequency domain of the code, e.g., PixNet [11] and FOCUS [8]. Finally, MAMBA performs adaptive bi-directional communication, as opposed to CamTalk [19] and [7], which instead use fixed transmission parameters.

III. MAMBA: ADAPTIVE BI-DIRECTIONAL CAMERA-DISPLAY COMMUNICATION

MAMBA consists of three components: a barcode encoding, an efficient decoding pipeline, and an adaptive communication protocol. Barcodes can be used for both unidirectional and bi-directional communication; either case leverages the same barcode encoding scheme and decoding process. On the one hand, the sender encodes the source data into a barcode that is shown on its display. On the other hand, the receiver captures the display of the sender with its built-in camera and decodes the corresponding barcode. In a bi-directional communication, barcodes are used at both the sender and the receiver simultaneously, and communication parameters are adjusted in real time through the MAMBA adaptive protocol.

This section first introduces MAMBA barcodes by explaining how they are encoded at the sender and decoded at the receiver. It then details the adaptive protocol used by MAMBA and the related implementation.

A. Encoding scheme

MAMBA is a color barcode formed by square blocks, each consisting of a set of pixels. Barcodes can be rectangular or square and use five different colors: black, white, red, green and blue. As it contains similar procedures of other barcodes, we will focus on the elements that make MAMBA barcode highly recommended for the easy identification of all the components with the minimum computational time. Figure 1 depicts the elements composing a MAMBA barcode.

The *corner trackers* delimit the barcode, thus allowing to distinguish it from the background. Each of the four corner trackers is a set of five by five blocks arranged as follows: one black block at the very center; a star-shaped locator as a collection of twelve blocks of the same base color around the black block; metadata encoded in the six remaining blocks around the corner locator. The base color used on each corner depends on its relative location with respect to the barcode, thereby allowing to read its content according to a well-specified block order. The particular shape of the blocks in the corner locator is meant to easily identify the corner tracker and quickly extract the metadata (as it will be apparent from the discussion in the next section). Such embedded metadata express the number of horizontal and vertical blocks (equivalently, the number of rows and columns) in the barcode. In particular, the number of rows is expressed by the six blocks on the top of the corner tracker, while the number of columns is expressed by the six blocks at the bottom of the corner tracker.

The *code area* contains the raw data encoded into blocks of four different colors: white, red, green, and blue; as a consequence, each block encodes two bits of information. MAMBA employs four colors as it has been shown that using eight is not beneficial, since the increased capacity is significantly impaired by the increased chance of quantization error in decoding [6]. Spacing separates corner trackers from the code area and the outer border of the barcode by means of white blocks. It also identifies the informative content of the barcode from its surrounding area. Unlike other designs [9, 15], MAMBA barcodes do not include any timing patterns but entirely rely on the corner trackers and the metadata encoded therein for control purposes.

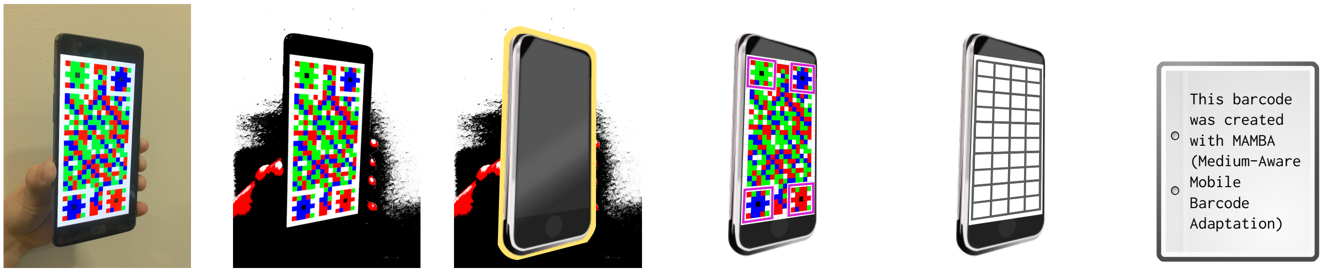
B. Decoding process

The decoding process of MAMBA consists of the different phases illustrated in Figure 2. The image captured by the receiver is pre-processed first to quantize colors and to remove the background. Corner trackers are then detected to perform perspective distortion erasure. Finally, the source data are obtained from the code area.

Color quantization. The color quantization process takes the captured image as input and reduces the number of colors to those allowed by MAMBA barcodes – namely, white, black, blue, red and green. This pre-processing step is needed to transform the source image in a form that can easily be manipulated in terms of the actual barcode colors, irrespective of the environmental conditions affecting the quality of the captured image (including ambient light).

Indeed, accurate color mapping is necessary to identify the different components in any color barcode. Different solutions exist for color quantization, with different processing overhead. For instance, each color can be identified by using Gaussian filtering and by comparing its distance to those in the allowed color palette, as in RDCode [6]. However, this solution incurs in a significant computation time. In contrast, MAMBA operates in the HSV colorspace through an appropriate definition of quantization thresholds, similar to COBRA [15].

Screen identification. Identifying a MAMBA barcode (namely, its corner trackers in the first place) operates



Source image **Color quantization** **Screen identification** **Corner detection** **Perspective distortion erasure** **Code scan**

Fig. 2: The decoding process in MAMBA. An image captured by the received is pre-processed first to quantize colors and to remove the background. Corner trackers are then detected to identify the barcode boundaries, derive the number of horizontal and vertical blocks, and perform perspective distortion erasure. Finally, the source data are obtained from the code area.

directly on the captured image after color quantization. In general, the boundary of the barcode may not be close to the corners of the captured image. This happens in practice as the transmitting and the receiving smartphones may not be perfectly aligned. In this respect, screen identification is the process that filters out the portion of the captured image that does not include the barcode. This is an optional pre-processing step meant to speed-up the subsequent phases of the decoding process, which is only executed once during transmission. MAMBA employs object detection, i.e., it identifies the screen of the transmitter in terms of the related boundaries instead of the source barcode directly. This is made easier by the presence of the outer spacing in the barcode – otherwise there would be no reliable way to distinguish between the barcode and unrelated background content purely based on the elements in the barcode itself. This specific detection is very precise as it recognizes each side of the barcode with pixel accuracy.

Corner detection. Corner detection in MAMBA starts from a black block in the source image and checks if all surrounding non-black blocks have the same color (i.e., one of blue, green or red). A corner tracker is successfully detected when a black block surrounded by neighboring colored blocks is found. After that, the boundaries of the star-shaped locators (i.e., the blocks corresponding to the vertices of the star) are identified. Their midpoint coordinates are then employed to find the blocks containing the metadata through an affine transformation [24]. This approach works in practice because perspective distortion is limited in the corner trackers, as their size is small compared to that of the source image [6]. The number of rows and columns in the barcode (each expressed over 12 bits) is finally extracted from the metadata.

Perspective distortion erasure. Depending on the relative position between the sender and the receiver, the captured image can be subject to perspective distortion[1] which may be significant. To overcome this issue, MAMBA applies projective mapping, which is the most general linear transformation perspective distortion erasure [25].

Using homogeneous coordinates [24], a projective mapping is described by a 3 by 3 transformation matrix $\mathbf{H}$ which can be determined on the basis of eight points: four in the original (i.e., distorted) space and the four corresponding points in the transformed (i.e., non-distorted) space. The first group of points is represented by the indices of the black blocks in the source (straight) barcode; whereas the second group of points is represented by the coordinates of the midpoint pixels in the corner trackers as determined during decoding.

Code scan. As a result of the previous phases, the number of horizontal and vertical blocks in the barcode are known, as well as the midpoint location of the corner trackers. It is then easy to identify the individual blocks in the code area by accounting for perspective distortion. In this respect, MAMBA applies projective mapping in the distorted space. Specifically, a matrix of indices is constructed first to describe the relative coordinates of the blocks, then an offset of half block size is added. Projective mapping is finally applied to such a matrix so as to obtain the coordinates of the blocks in the source image. Each block is directly indexed by such coordinates to derive the corresponding bit values. These values are then concatenated to form the raw message that can be used by the higher layers.

C. *Adaptive transmission protocol*

Before proceeding further, it is worth recalling here the reference scenario considered for adaptive bi-directional communication. Two smartphones communicate with each other, namely, each device employs its own display to show MAMBA barcodes; at the same time, the device employs its front-facing camera to receive the MAMBA barcodes displayed by the other end . Operationally, the protocol distinguishes between the roles of sender and receiver, even though each device may simultaneously act as both.

After a brief overview of the general communication process, the rest of this section details the mechanism used to

---

[1] Lens distortion is a different aspect due to the characteristics intrinsic to camera optics. Preliminary experiments have demonstrated that lens distortion is not significant in the considered scenario; accordingly, MAMBA only applies perspective distortion to decode barcodes.

establish a connection between the sender and the receiver, as well as the adaptive features of the proposed protocol.

1) Overview: The communication process starts by establishing a session, namely, with the parties advertising their availability to transfer data to the other endpoint. Afterwards, communication proceeds as follows. Application-level (source) data are divided into different segments with a length of N bytes. Each segment is then arranged into a window of W messages, each of size $S = N/W$. Messages contain a header (illustrated in Figure 3) with the following fields: an 8-byte checksum as a cyclic redundancy check value; one byte of binary flags, including three bits used in the connection establishment phase and 5 bits for reporting error messages (currently unused); 2 bytes for the sequence number; 2 bytes for the acknowledge number; and 2 bytes for the window size. The payload size ranges from 130 to 2,120 bytes for typical values of block sizes (i.e., ranging from 17 to 43 pixels). Note that the receiver may also send its own source data at the same time, since the channel allows for full-duplex operations. In this case, control messages are piggybacked into the header.

2) Session establishment: The session establishment phase follows a three-way handshake inspired by the TCP, similar to [7, 19]. However, session establishment in MAMBA relies on a *peer-to-peer* mode, in the sense that the two communicating ends do not have to agree beforehand on being sender (i.e., client) or receiver (i.e., server). This is especially useful for practical user interactions, as otherwise the roles of sender and receiver (from the protocol perspective) would have to be defined through some out-of-band mechanism (e.g., by users explicitly agreeing on the roles of their smartphones).

Before initiating the connection, each device creates a unique identifier (`UID`) of 32 bits, obtained by applying a hash function on some information specific to the phone (such as the MAC address or IMEI). After that, the device displays a special header (with the `SYN` flag set) that embeds such a `UID` into the `ACK_NO` and `SEQ_NO` fields so as to define the roles of sender and receiver. After reading the barcode of the other peer, the device compares its own `UID` with the other one: if its own `UID` is greater than that of the other end, the device takes the role of sender. In case the two values are equal, other 32 bits are sent with the same header. If the two codes are different, the device starts displaying a new frame, thereby entering the data transfer phase. To counter possible frame losses, the protocol continues including the special header in the first frame of the data transfer phase. This ensures that the smartphone which missed a frame can overcome the loss due to the error recovery mechanism of the protocol. A connection is terminated by setting the `FIN` flag.

3) Adaptive communication: The protocol employs a retransmission scheme to recover from errors affecting messages within a certain window. The `ACK_NO` and `SEQ_NO` fields of the header are used for this purpose after a connection has been successfully established. Retransmitted messages are used as a basis to *dynamically* adapt communication parameters.

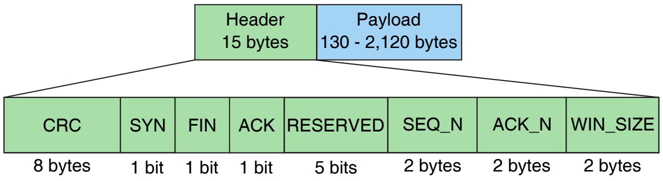


Fig. 3: Protocol header.

Algorithm 1: Frame rate and block size adaptation

```
input: CircularBuffer packets_to_send, display_time,
       block_size
buffer_size = sizeof(packets_to_send),
loop_counter = 0
while sizeof(packets_to_send) $>$ 0 do
    foreach element $\in$ packets_to_send do
        $Send$(element);
    $RemovePacketsWithACK$(packets_to_send);
    if sizeof(packets_to_send) $>$ 2 and
     loop_counter $>$ 1 then
        display_time++
    else if sizeof(packets_to_send) $<$ 2 and
     loop_counter $<$ 1 then
        display_time$--$
    if sizeof(packets_to_send) == buffer_size then
        loop_counter$++$
    buffer_size = sizeof(packets_to_send)
if counter $< R_{lo}$ then
    block_size$--$
else if counter $> R_{hi}$ then
    block_size++
```

MAMBA applies two forms of adaptivity on top of the window-based transmission scheme. The pseudo-code of the adaptive protocol is outlined in Algorithm 1. The first is *frame rate adaptation*, which allows to overcome lack of precise synchronization between the two phones. In this respect, MAMBA evaluates the number of successfully decoded frames every W transmitted messages or whenever other operating parameters (i.e., the block size) change. The obtained value is used to derive the alter the display time of the barcode by adding delays so as to achieve a desired (target) frame rate in steps of t_a.

The second form of adaptivity is *frame length adaptation* which adjusts the block size after a window is successfully transmitted. In particular, the protocol considers two thresholds to define the number of retransmissions within a window: a lower threshold R_{lo} under which the block size is decreased; and an upper threshold R_{hi} over which the block size is increased in steps of B_a pixels. The block size remains the same as long as the number of retransmissions is within the two thresholds.

D. Implementation

We implemented MAMBA as a mobile application for the Android mobile operating system. The related implementation entirely relies on the Android API level 25, supported by Android version 7.1 and above, without requiring any third-party libraries. OpenGL ES 2.0 was employed to render barcodes and camera views, as well as to speed up certain parts of the processing.

Optimizations. MAMBA's code was profiled first to identify the most critical functions in terms of decoding time. The first one was represented by color quantization, in particular, color space conversion. Software-based options were determined to have a significant computation time. Consequently, MAMBA carries out hardware-accelerated color quantization through a custom fragment shader. In particular, the shader performs RGB to HSV color space conversion and applies the thresholds for color quantization. Another optimization was parallelization. In the decoding phase, MAMBA performs the detection of each corner in parallel. Indeed, the corner detection algorithm adopted here takes advantage from the availability of multiple cores by running multiple threads concurrently, i.e., one for each of the corners (which allows to cut down the processing time by a factor of four). As for the bi-directional protocol, MAMBA performs screen updates and camera captures in parallel; it also uses a circular buffer for messages in a window to reduce the memory access and management overhead.

Camera access. MAMBA leverages the capabilities of the `Camera2` [2] Android API that offers full customization of camera capture. There are two different modes to capture images: as a burst of images or as a video stream of frames. A preliminary evaluation has shown that capturing burst of images is a more flexible and powerful option than taking frames from a video stream. In fact, such an approach obtains higher resolutions and (or) frame rates, especially when high-FPS modes are available or when specific video capture modes are not supported. Capturing individual images also allows fine-grained control of the parameters related to the camera, including exposure and focus. Indeed, an appropriate setting of these parameters allows to significantly reduce the capture time, especially with reference to auto-focus.

IV. PERFORMANCE EVALUATION

In the following, we present an experimental evaluation of MAMBA's performance. After introducing the experimental setup, the rest of the section details the obtained results and contrasts them with the state of the art for scenarios involving both unidirectional and bi-directional communication.

A. Experimental setup

Testbed. A testbed was setup by using two smartphones as a sender-receiver pair. The smartphones were connected to a laptop (acting as controller for the experiments) through a USB cable. The two smartphones were attached to supports during the experiments, similar to the state of the art [7, 8, 19], so as to accurately set the parameters for the evaluation and ensure consistent results. Unless otherwise specified, experiments were conducted in an office under uncontrolled light settings, with both natural light from a large window and artificial light generated by fluorescent lamps: the actual illuminance varied from 150 to 350 lux as reported by the LMT pocket lux 2 meter[3]; half of the maximum brightness level was used for the display of the transmitter.

Parameter	**Description**
Mobile device	OnePlus 3T
Camera resolution	16 MP
Display rate at the sender	15 FPS
Capture rate at the receiver	30 FPS
Image capture resolution	2K (1,728×2,304 px)
Distance between devices	20 cm

TABLE I: Experimental parameters for the evaluation.

Experiments were carried out by using two identical One Plus 3T smartphones[4] for the sake of simplicity. This specific model was chosen because of the features of the built-in hardware, including its screen and cameras. In particular, the smartphone has a display resolution of $1,920 \times 1,080$ pixels as well as a 16 MP sensor for both the front-facing and the rear-facing cameras. The rear-facing camera has auto-focus and can capture 720p video at 120 FPS, while the front-facing camera has a fixed focus and a capture rate of only 30 FPS; they both can take pictures at a 4K resolution. The two smartphones run OxygenOS 4.1.6, an Android-based operating system corresponding to Android version 7.1.1.

Methodology. To reduce the bias in the evaluation, the operating system was reset to factory settings before installing MAMBA. All wireless connections and power management features were also disabled during the experiments to avoid the impact of factors unrelated to the evaluation. Each experiment was conducted according to the independent replication method; it consisted of at least five iterations and a minimum number of 1,500 frames exchanged. The results shown next report the average values as well as the corresponding standard deviations as error bars, when they can be distinguished in the plots. The key experimental parameters are summarized by Table I.

B. Experimental results

The results obtained from the experiments are divided in two parts. The former part reports MAMBA's performance for a unidirectional communication, in terms of processing overhead and the reliability of decoding. The latter part describes the results obtained for a bi-directional communication with focus on the throughput and the dynamic adaption of operating parameters.

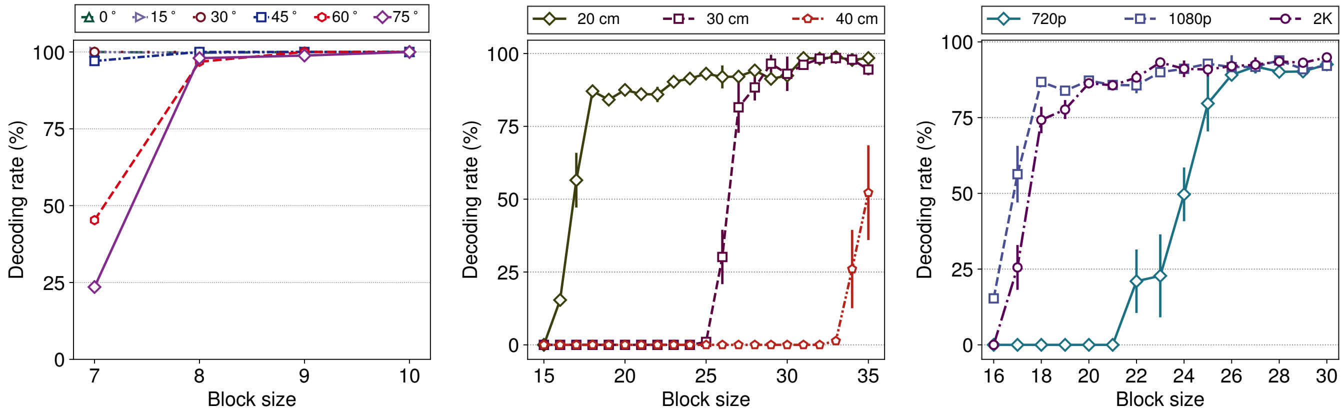
1) Unidirectional communication: As the first step to evaluate MAMBA, we consider the scenario of an unidirectional communication. The scenario consists in a smartphone used as a transmitter and another one as a receiver, similar to [15]. The receiver used the rear-facing

---

[2] `https://developer.android.com/reference/android/hardware/camera2/package-summary`

[3] `http://www.lmt-berlin.de/en/plux2.html`

[4] `https://oneplus.net/3t`



(a) (b) (c)

Fig. 4: Decoding rate as a function of the block size for different (a) capture angles, (b) distances between the smartphones, and (c) image capture resolutions.

Phase	**Time** (ms)	σ (ms)
Color quantization	Negligible	n/a
Screen identification	206.5659	24.4079
Corner detection	6.5368	5.6174
Projective mapping	0.0142	0.0625
Code scan	0.9113	0.6295
All phases	210.1416	22.8500
No screen identification	6.0921	2.0298

TABLE II: Breakdown of the execution time for the different decoding phases in MAMBA.

camera to capture the screen of the sender.

Processing overhead. This set of experiments characterizes the performance of MAMBA in terms of the *execution time*, as the time (in milliseconds) taken by each of the different phases in the decoding process.

In particular, Table II shows the breakdown of the execution time for the different phases in the decoding process. As color quantization is hardware-accelerated, its processing time is negligible (i.e., with respect to the time needed to copy the image from the hardware buffer of the camera to the main memory). As apparent from the numbers, screen identification is the most computationally expensive phase. This is because it needs to scan the whole captured image to identify the boundaries of the barcode. However, screen identification does not need to be performed more than once for a sequence of images if the smartphone does not move or moves slowly. In fact, corner detection can operate on the source image by using an arbitrary partition of the areas where corners are expected to be found, for instance, by dividing the source image into four rectangles of the same size. Projective mapping and code scan have both a very low processing time, in practice almost negligible with the rest. The total processing time with no screen identification is about 6 ms, which achieves a frame rate over 120 FPS.

Reliability. The last set of experiments characterizes the impact of capture angle, distance, and picture resolution on the reliability of decoding. In particular, we measure the *decoding rate*, expressed as the percentage of correctly decoded frames.

Figure 4a illustrates the decoding rate as a function of the block size for different capture angles. Here, capture angle refers to the rotation of the sender smartphone with respect to its vertical axis while the receiver remains in a fixed position. Capture angles range from 15 to 75 degrees in steps of 15 degrees. The results in Figure 4a show a decoding rate above 95% when the block size is greater than 8 pixels, for all the considered values of capture angle. This is because of the perspective distortion erasure method used in the decoding process. Clearly, using such a method ensures higher decoding rates even if the device is not at all aligned (including angles over 60 degrees) with respect to the one receiving data.

Figure 4b depicts the decoding rate as a function of the block size for different values of the distance between the two smartphones. The figure clearly shows that MAMBA is only able to decode barcodes with higher block size when the distance increases, reducing its overall throughput. However, it is worth noting that MAMBA achieves a decoding rate close to 100% in many cases, especially when the block size is above 15 pixels and the distance is short. The general trend of the curves corresponding to different distances is consistent: there is a cut-off value of block size under which the decoding rate is low. In contrast, the decoding rate has higher fluctuations for high values of the block size. Nevertheless, the results are comparable irrespective of the distance when the block size is higher than 34 pixels.

Finally, Figure 4c shows the decoding rate as a function of the block size for different resolutions of the image captured by the receiver. Increasing resolution clearly improves the decoding rate, as expected. The rate increases rapidly for 2K and 1080p resolutions by reaching values above 80% for values of the block sizes greater than 17 and 18 pixels, respectively. In contrast, a 720p resolution allows reliable decoding only for block sizes of 25 pixels and above. These

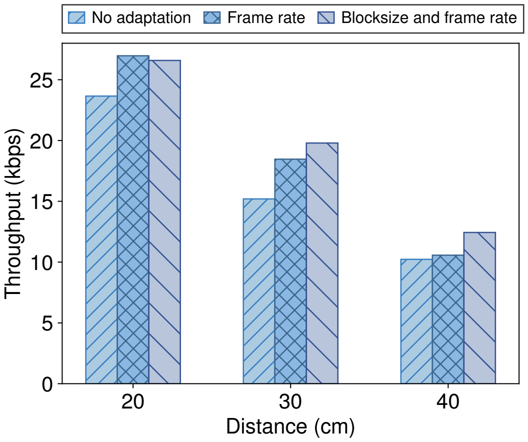


Fig. 5: Throughput of MAMBA with different adaptivity methods as a function of the distance between the smartphones.

results demonstrate the importance of scalable decoding, as higher resolutions incur in higher processing overhead.

2) Bi-directional communication: The results presented next were obtained in a scenario involving bi-directional communication. For this purpose, the experimental setup follows the same scenario introduced at the beginning of Section III-C: the two smartphones are facing each other in such a way that each camera can fully capture the screen of the other device [7, 19]. Accordingly, both smartphones use their front-facing cameras in this case, as opposed to the unidirectional scenario.

As a representative use-case, the evaluation considers a file exchange application [7]. Specifically, a file is selected as input, which is transferred from one smartphone to the other one; the same applies to the two smartphones at once. At the time communication starts, the two smartphones simultaneously transmit and receive their data. The files exchanged have the same size of 10.5 Kbytes. Unless otherwise specified, the following protocol-specific parameters were used: a window size of $W = 5$ messages; retransmission thresholds of $R_{lo} = 3$ and $R_{hi} = 6$; increment (decrement) steps for the operating parameters equal to $B_a = 1$ pixel and $t_a = 2$ ms. The initial block size was set to $B = 30$ pixels, while the initial frame rate was set to 15 FPS.

The experimental results are divided in two parts: the first evaluates the impact of adaptation on throughput while the second provides a comparison with the state of the art.

Impact of adaptation on throughput. The first set of experiments evaluates the impact of adaptation on the *throughput*, defined as the amount of raw data transferred from the sender to the receiver in terms of non-duplicate frames correctly received divided by the total transfer time. To this regard, bi-directional communication was carried out with different options: no adaptation (as a reference), frame rate adaptation only, as well as joint frame rate and length adaptation. Figure 5 illustrates the corresponding results as a function of the distance between the two devices.

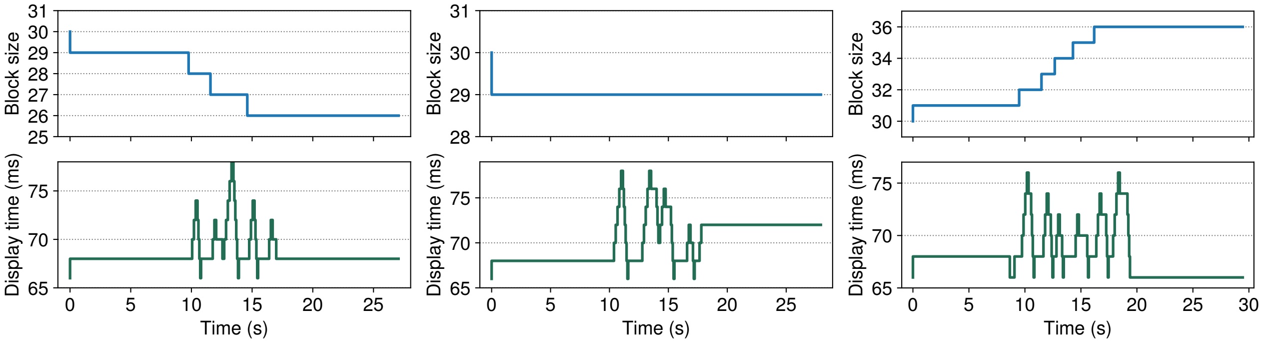
The figure clearly shows how the lack of adaptation results in a lower throughput for all considered values of distance. This happens because the data transfer protocol cannot increase the capacity of the barcode when devices are close; conversely, higher distances produce higher frame losses and retransmissions. The schemes including adaptation perform consistently better, instead, with values of throughput ranging from 11 to 28 Kbps. These also correspond in an increase between 5% and 20% over the case with no adaptation. Joint frame rate and length adaptation achieves the highest increase in the throughput irrespective of the distance. Frame rate adaptation only, instead, performs better than the other schemes at a short distance. However, the throughput is almost the same as with no adaptation for the highest distance of 40 cm. This happens as communication is significantly affected by frame losses caused by smaller blocks. In contrast, the joint frame rate and length adaptation has some overhead that results in a slightly lower throughput for the smallest distance of 20 cm. The advantage of using frame length adaptation is apparent for higher values of the distance, instead, demonstrating that the adaptive approach taken by MAMBA is practical and effective.

Scenario	Vertical illuminance	Horizontal illuminance	Description
Bright	1,277 lux	637 lux	Devices and external light sources at their nominal brightness
Medium dark	227 lux	127 lux	Devices and external light sources at 20% of their nominal brightness
Dark	25 lux	55 lux	Devices at 10% of their nominal brightness without external light sources

TABLE III: Configurations used for the measurements about adaptation with respect to light conditions.

Dynamic parameter adaptation. The second set of experiments characterizes dynamic adaptation of the communication parameters under light-controlled conditions in terms of the display time at the sender (i.e., the reciprocal of the frame rate) and the value of the block size as a function of time. In detail, experiments were conducted by placing the smartphones at a distance of 25 cm in a darkroom. The actual configurations employed correspond to the bright, medium-dark and dark configurations described in Table III. Three iterations were run for each experiment; the values in Figure 6 report the values obtained in a representative run, which are consistent with the rest of the iterations in the experiments.

Figure 6 shows the communication parameters over time for the bright conditions. The results show that the block size reduces over time (eventually reaching the value of 26 pixels) and that the frame rate is adjusted during these variations. The variation in the display time occurs during the corresponding changes in the frame length as a result of increasing processing time. Even though there is some fluctuation in these parameters, a stable value is eventually reached when the block size does not change any more.



(a) (b) (c)

Fig. 6: MAMBA adaptation behavior for (a) bright, (b) medium-dark, and (c) dark conditions.

The display time exhibits a similar trend for medium-dark and dark conditions. The fluctuation does not accompany the sudden reduction in the block size shown in Figure 6b as it is due to varying channel conditions. The transient ends with an increase in the display time, which is consistent with the lower block size used at the steady state. Figure 6c shows an opposite trend than the one for bright conditions: the block size increases until it reaches a value of 36 pixels. The display time varies quickly before settling to 66 ms. This behavior is also consistent with MAMBA's adaptation policy.

The results in Figure 6 overall demonstrate that MAMBA is able to adapt to a variety of challenging environmental conditions. Note that the configurations considered in the experiments and reported in Table III are close to extreme, as they include very bright and very dark scenarios. In this regard, these conditions are much more challenging than those used in the evaluation of CamTalk [19].

Comparison with other bi-directional schemes. The solutions similar to MAMBA in the context of bi-directional camera-display communications are represented by CamTalk [19] and the work in [7], both based on QR codes. We compared the performance of MAMBA with the latter work [7]. The experiment involved running both applications under uncontrolled light conditions at different distances as performed in [7]. The file exchanged between the smartphones has a size of 2KB. We set the block size of MAMBA barcode to 55 pixels to ensure that both solutions embed (initially) the same amount of data. Figure 7 depicts the results obtained for the execution times as distance increases between the smartphones. As expected, MAMBA requires less time to complete the file exchange (i.e., up to 8× faster). Moreover, MAMBA's execution time remains substantially constant across the three different distances. In fact, the probability of losing a packet is reduced by tuning the block size so as the image appears sharper at the receiver side. In contrast, the execution time for the other solution increases linearly with the distance.

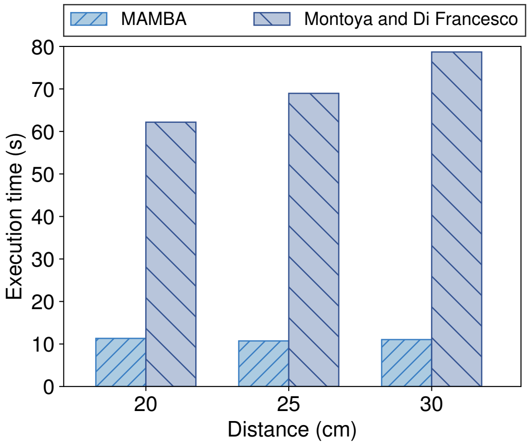


Fig. 7: Execution time of MAMBA and [7] over distance.

V. Discussion

Reliability and throughput. As pointed out by several works [8, 10], capacity and throughput are not the primary goals in designing codes for screen-camera links. In fact, high throughput can achieved at the expense of a high message loss rate [6], especially when communication parameters suitable for the environmental conditions are not known a priori. This has originated significant research in layered and hierarchical codes embedding multiple streams for different receivers in diverse operating conditions (e.g., those close to the sender, those with a bad channel, and so on). Despite the improvements in reliability due to such an approach and the use of forward error correction, it is enough to miss one frame to prevent decoding in many cases, for instance, when source data are compressed or encrypted. Indeed, this work demonstrates that is possible to achieve 100% reliability: even though the obtained throughput is relatively low, the robustness and flexibility of MAMBA cannot be matched by unidirectional communication schemes.

Comparison with radio-based technologies. Camera-display communication can be employed as an alternative to radio-based wireless communication technologies. In this respect, Montoya and Di Francesco [7] have shown that bi-directional camera-display communication is practical for exchanging a small amount of data, despite the much lower throughput than Bluetooth Low Energy and WiFi. This happens as other communication technologies have a higher overhead to initiate the transmission, for instance, due to device pairing or first-time association to an access point. Indeed, this work does not advocate using screen-camera links as a replacement for radio spectrum. Instead, the proposed approach demonstrates that they can be used with the same flexibility and reliability of wireless communications. In fact, camera-display communication with MAMBA allows robust bi-directional message exchange that can be employed, for instance, to carry out a Diffie-Hellman key exchange [26].

VI. CONCLUSION

This article has addressed scalable and reliable camera-display communication between smartphones through the MAMBA (Medium-Aware Mobile Barcode Adaptation) application framework. MAMBA employs color barcodes and provides efficient decoding through advanced computer vision techniques as well as efficient algorithms, which can take significant advantage from modern mobile device hardware. Efficiency in MAMBA has two major consequences: decoding scales with the number of blocks in the barcode, rather than with the number of pixels in the source image (as in the state of the art); very short decoding time enables fast feedback when bi-directional communication is employed. MAMBA is also the first solution (to the best of the authors' knowledge) that leverages scalable decoding to perform dynamic adaptation of communication parameters for bi-directional camera-display communication. This approach allows to address both synchronization issues and the impact of environmental factors (such as ambient light) on communication reliability. Experimental results have demonstrated that MAMBA is efficient as well as robust, as it can tailor communication parameters to even extreme environmental conditions.

A few future research directions stemming from this work are of particular interest. Among them, a user study could be carried out to evaluate the usability of the system when users are actually holding phones in front of each other. From a different perspective, it would also be very valuable to identify new application scenarios that can benefit from bi-directional communication with MAMBA.

REFERENCES

[1] P. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 4, pp. 2047–2077, 2015.

[2] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Capacity of pervasive camera based communication under perspective distortions," in *Proc. of PerCom '14*, March 2014, pp. 112–120.

[3] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 48, no. 1, pp. 28–35, January 2015.

[4] T. Langlotz and O. Bimber, "Unsynchronized 4d barcodes," in *International Symposium on Visual Computing*. Springer, 2007, pp. 363–374.

[5] R. Zurawski, *Industrial communication technology handbook*. CRC Press, 2014.

[6] A. Wang, S. Ma, C. Hu, J. Huai, C. Peng, and G. Shen, "Enhancing reliability to boost the throughput over screen-camera links," in *The 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14, 2014, pp. 41–52.

[7] M. L. Montoya Freire and M. Di Francesco, "Reliable and bidirectional camera-based communications with smartphones," in *The 7th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2016)*, June 2016, pp. 1–7.

[8] F. Hermans, L. McNamara, G. Sörös, C. Rohner, T. Voigt, and E. Ngai, "Focus: Robust visual codes for everyone," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16, 2016, pp. 319–332.

[9] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *The 19th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '13, 2013, pp. 15–26.

[10] W. Hu, J. Mao, Z. Huang, Y. Xue, J. She, K. Bian, and G. Shen, "Strata: Layered coding for scalable visual communication," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14, 2014, pp. 79–90.

[11] S. D. Perli, N. Ahmed, and D. Katabi, "PixNet: interference-free wireless links using LCD-camera pairs," in *Proc. of MobiCom '10*, 2010, pp. 137–148.

[12] N. Saha, M. Ifthekhar, N. T. Le, and Y. M. Jang, "Survey on optical camera communications: challenges and opportunities," *IET Optoelectronics*, vol. 9, no. 5, pp. 172–183, 2015.

[13] Z. Yang, Y. Bao, C. Luo, X. Zhao, S. Zhu, C. Peng, Y. Liu, and X. Wang, "Artcode: Preserve art and code in any image," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: ACM, 2016, pp. 904–915.

[14] W. Huang and W. H. Mow, "Picode: 2d barcode with embedded picture and vicode: 3d barcode with embedded video," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 139–142.

[15] T. Hao, R. Zhou, and G. Xing, "Cobra: color barcode streaming for smartphone systems," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 85–98.

[16] T. Zhan, W. Li, X. Chen, and S. Lu, "Capturing the shifting shapes: Enabling efficient screen-camera communication with a pattern-based dynamic barcode," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 52:1–52:25, Mar. 2018.

[17] 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*, ser. VLCS '14, 2014, pp. 27–32.

[18] S. Hranilovic and F. R. Kschischang, "A pixelated MIMO wireless optical communication system," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 4, pp. 859–874, July 2006.

[19] M. Xie, L. Hao, K. Yoshigoe, and J. Bian, "Camtalk: A bidirectional light communications framework for secure communications on smartphones." in *SecureComm*. Springer, 2013, pp. 35–52.

[20] T. Li, C. An, A. T. Campbell, and X. Zhou, "HiLight: Hiding bits in pixel translucency changes," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 18, no. 3, pp. 62–70, January 2015.

[21] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '15, 2015, pp. 197–211.

[22] 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 *Proc. of MobiSys '10*, 2015, pp. 181–195.

[23] S. Shi, L. Chen, W. Hu, and M. Gruteser, "Reading between lines: High-rate, non-intrusive visual codes within regular videos via ImplicitCode," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 157–168.

[24] W. Burger and M. J. Burge, *Digital image processing: an algorithmic introduction using Java*. Springer, 2016.

[25] L. Jagannathan and C. Jawahar, "Perspective correction methods for camera based document analysis," in *Proc. First Int. Workshop on Camera-based Document Analysis and Recognition*, 2005, pp. 148–154.

[26] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.