IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 October 2023, accepted 4 November 2023, date of publication 9 November 2023,
date of current version 20 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331754

RESEARCH ARTICLE

Video Display Field Communication: Practical Design and Performance Analysis

YU-JEONG KIM, (Graduate Student Member, IEEE), PANKAJ SINGH, (Senior Member, IEEE), AND SUNG-YOON JUNG, (Senior Member, IEEE)
Department of Electronic Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding author: Sung-Yoon Jung (syjung@ynu.ac.kr)

This work was supported by the Leaders in INdustry-University Cooperation 3.0 (LINC3.0) Project funded by the Ministry of Education and the National Research Foundation of Korea (NRF) under Grant LINC3.0-2022-54.

ABSTRACT Display field communication (DFC) is a spectral domain-based, unobtrusive display-to-camera (D2C) communication method in which a commercially available digital display serves as the transmitter and an off-the-shelf camera serves as the receiver. The data are transmitted through video and image frame sequences over D2C links in such a way that it do not obstruct the normal viewing experience. In this study, we propose a Video-DFC approach to detect data embedded in running videos. First, we propose the design of a frame packet structure which is used to facilitate encoding input data as a specific frame sequence. Then, we design a color code-based 4-point block pattern, that is inserted into the four vertices of the transmitted image frames. This block pattern enables frame-to-frame synchronization and accurate extraction of the data-embedded region from the received frames. Our proposed Video-DFC approach demonstrates the potential of transmitting a large amount of data and achieving a data transfer by embedding distinct data in each frame of the running video. To evaluate the robustness of Video-DFC, we conducted extensive experiments varying the frame packet structure, input video, and camera resolution. Depending on the frame packet structure, the results showed a maximum achievable data rate of 27 kbps. Overall, our findings suggest that Video-DFC has the potential to be a next-generation, high-capacity D2C communication method.

INDEX TERMS Display field communication (DFC), display-to-camera (D2C) communication, frame synchronization, invisible message, video signal processing.

I. INTRODUCTION

The demand for multimedia content has seen a steady increase in recent decades, in tandem with the rise of digital video platforms [1], [2]. Additionally, advances in camera technology have led to the widespread commercialization of high-resolution camera smartphones. These developments have propelled the growth of display-to-camera (D2C) communications [3], [4], [5], a critical technology that complements the constraints of radio frequency-based communications in the future ubiquitous era. D2C communication is a subfield of optical camera communication [6], [7], that enables short-range communication over a wireless optical link. In this technology, digital displays and cameras function as transmitters and receivers, respectively. The main objective of D2C systems is to embed and transmit data via images or videos in a hidden way while providing high-quality multimedia content to the user.

The research in the area of D2C communication primarily involved the use of 2D barcodes, particularly the ubiquitous QR codes, to transmit a small amount of data from printed media to camera devices [8]. However, the size, location, and data capacity of these barcodes are limited, so researchers began to explore 2D color barcodes [9], [10], [11] that can provide higher data capacity. However, these 2D barcodes still have limitations, such as limited data transfer and obtrusiveness. To address these challenges, researchers have proposed a novel approach to embed data directly into the spatial or spectral domain of an image that enables high-capacity data transmission while minimizing perceptual

The associate editor coordinating the review of this manuscript and approving it for publication was Ding Xu.

artifacts. One category of techniques used to embed data into images is spatial domain-based data embedding [4], [12], [13], [14]. In this approach, small perturbations are introduced to the intensity of image pixels to transmit data while preserving the quality of visual content. In [13], the authors proposed a novel image coding scheme designed for transparent, efficient, and robust transmission of data between screens and cameras which is adaptable to multiple device types. The proposed system, called TERA, is based on a combination of color modulation and transparency control techniques that allow data to be encoded and transmitted with minimal interference to the user's visual experience. Another recent work on D2C communication proposed a real-time visible light communication system based on LED displays and smartphones [14]. The system encodes the hidden data using the alternate bit-flipping repeat coding and then combines the encoded data frame with the image frame using a data insertion process. At the receiver, fast image processing algorithms such as fast ROI detection and adaptive binarization technique were used to reduce the computational complexity of extracting and decoding information from the captured video frames in real time. Experimental results show that with an LED display panel having a refresh rate of 150 Hz and a 30 fps smartphone camera, a data transmission rate of 30 bps can be achieved for one LED display point. However, spatial-domain perturbations can cause changes to pixel intensity that can make data decoding difficult and increase the likelihood of errors in the received embedded data.

Recent developments in the artificial intelligence industry has led to the introduction of various D2C techniques based on deep convolutional neural networks [15], [16], [17], [18], [19], [20]. Stegastamp [15] proposed a technique called steganography, which allows for real-time decoding of hyperlinks embedded in printed or displayed photographs. Deep D2C-Net [16] developed a fully end-to-end encoding-decoding network structure that enables high-quality data-embedded images and robust data acquisition simultaneously. The work in HiDDeN [17] proposed a novel data-hiding method using deep neural networks (DNNs). In this method, a DNN is trained to encode a secret message into an image while minimizing the perceptual differences between the original and modified images. The encoded message can then be decoded from the modified image using another neural network. HiDDeN demonstrated that generative adversarial networks (GANs) based training between the cover image and encoded image finally improves the visual quality of encoded images. Similarly, [19] presented an approach to hide messages (hyperlinks) into common images. The proposed approach, called RIHOOP, can generate images of good visual experience by adversarial training, containing invisible hyperlinks that are detectable by cameras on mobile devices under various unconstrained environments. In particular, RIHOOP designed a novel distortion network based on 3D rendering to enhance the robustness of information extraction. In another work, [20] presented SteganoGAN, a technique for hiding arbitrary binary data in images using GANs, which allows us to optimize the perceptual quality of the images produced by the model.

In contrast, D2C communication employing spectral domain-based data embedding [21], [22], [23], [24] can maintain visual quality while embedding data in spectral-domain coefficients. Display field communication (DFC) [22] is a spectral domain-based data embedding approach that reduces the impact of the D2C wireless channel using the spread-spectrum effect. The pioneering work on DFC [22] includes the analysis of 1D discrete Fourier transform (DFT)-based data embedding, while [23] extends this concept by embedding the data in two dimensions of an image to achieve an increase in achievable data rate (ADR). In another work on DFC, the discrete cosine transform (DCT) was used and it is shown that DCT-based DFC is indeed practically possible [24]. The work performs experiments for different channel conditions and input images, and presents a real-world data detection for color images that demonstrates the effectiveness of DFC for D2C communication.

Although multiple approaches exist in literature to establish a D2C communication system, a significant drawback of the existing technologies is the limited number of data bits that can be transmitted to the user, since the data transmission is done via still images. In this study, we propose a novel approach that overcomes the data capacity limitation of still images by embedding data in a video to enable full-frame communication in the real world. The proposed method, called Video-DFC, enables stable streaming of large-capacity data from a series of frames. Despite the challenges posed by pixel alignment between consecutive frames and the D2C wireless channel, experiments have shown that Video-DFC performs exceptionally well. In particular, we examined the feasibility of Video-DFC in terms of bit error rate (BER) and ADR for various system design parameters. In the experiments, we obtained a maximum ADR of 27 kbps. In general, BER and the maximum data rate have an inverse relationship, which means that higher data rates lead to higher BER. However, from the experiments, we found that the increase in data rate is much higher than the increase in BER, resulting in a higher ADR. This higher ADR is a remarkable feature of our proposed DFC scheme, as it provides the opportunity to reduce BER by using more powerful channel coding schemes, albeit at the cost of some reduction in data rate.

The remainder of this article is organized as follows. Section II provides a detailed overview of the proposed Video-DFC model, including its schematic architecture, frame packet structure, 4-point block pattern, and data embedding and detection mechanisms. Section III describes the distortion caused by the rolling shutter effect in received video frame sequences. Section IV describes the frame packet synchronization process by comparing the received images according to the sampling ratio and explains the extraction

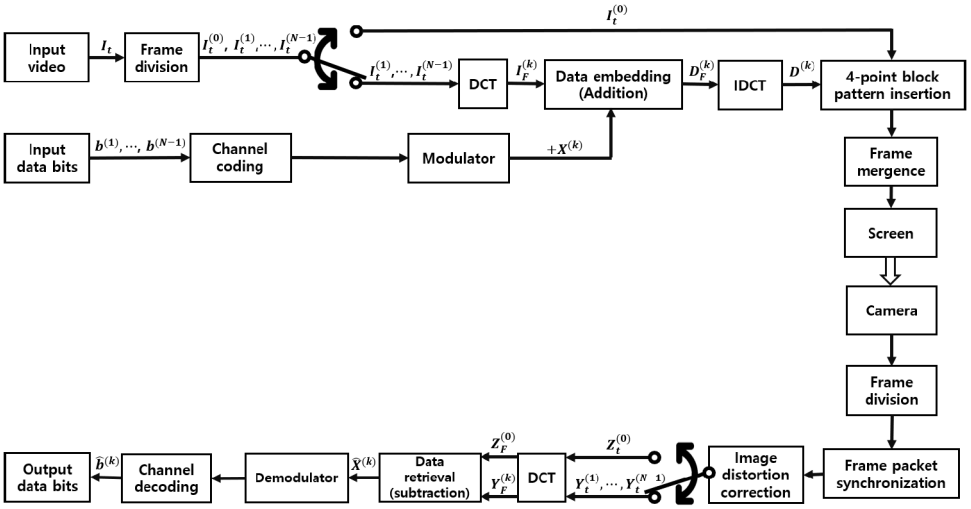


FIGURE 1. Schematic architecture of the proposed Video-DFC model.

process of image regions. Section V presents the experimental results obtained by changing various parameters such as the frame packet structure, type of input video, and camera resolution in a fixed experimental environment. Finally, the paper concludes with Section VI.

II. THE PROPOSED VIDEO-DFC MODEL

Figure 1 shows a comprehensive schematic architecture of the proposed Video-DFC system. First, the input video is split into sequential frames using frame division. The split frames are then divided into a reference frame and the remaining frames as data-embedded frames. As shown in Fig. 1, one reference frame serves the $N-1$ data-embedded frames. This interlacing of reference frames with data-embedded frames facilitates data decoding at the camera receiver. In Video-DFC, data embedding is performed in the spectral domain of an image frame. Therefore, the $N-1$ split frames were first converted to the spectral domain using the DCT operation. At the same time, the binary input data bits ($b \in \{0,1\}$) are channel-coded and modulated to symbol X, which are then embedded into the image frame using the addition allocator. The spectral-domain data-embedded frames are then converted back to the spatial domain to be displayed on the electronic screen. This was accomplished using the inverse discrete cosine transform (IDCT). However, before the data-embedded frames and the reference frame are merged and multiplexed on the display, a color code-based 4-point block pattern is inserted into the four vertices of each transmitted frame. This process allows accurate data-embedded region extraction and frame-to-frame synchronization at the receiver.

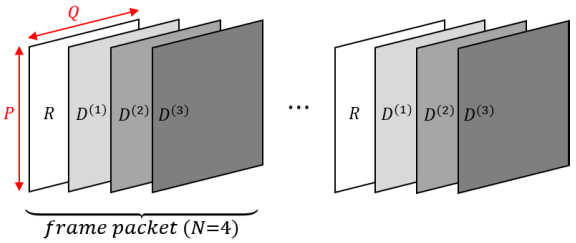
At the receiver end, the transmitted video is captured by a camera. The captured video is once again split into sequential frames. Then we synchronize the frames by distinguishing between reference frames and data-embedded frames using the 4-point block pattern. First, the image region is extracted using the difference of images in consecutive frames. That is, by performing subtractions of consecutive frames, we obtain the difference image. Next, using the four blocks, we can synchronize the frames and extract the four points of the image region. Owing to the D2C channel, the received video frames suffer from harsh channel distortions. Therefore, each distorted frame is reconstructed to obtain a distortion-corrected frame. The reconstructed reference frames and data-embedded frames are then converted to the spectral domain. Subsequently, the data are recovered by subtraction data retrieval, and the final output data bits are estimated by demodulation and channel decoding. The following subsections provide a detailed description of the mathematical model and techniques used in the proposed Video-DFC method.

A. FRAME PACKET STRUCTURE

The videos are transmitted in the form of a frame packet structure. That is, the frames are combined in a specific order to form a frame packet structure, where each packet contains one reference frame and several data-embedded frames. In particular, a frame packet consists of N frames, in which the first frame is the reference frame whereas the remaining $N-1$ frames are data-embedded, as depicted in Fig. 2. The order of the frames is also illustrated. We first transmit the reference frame and then data-embedded frames. Mathematically, the frame packet structure can be given as

$$I_t^{(n)} = \begin{cases} R, & n = 0, N, 2N, \ldots \\ D^{(1)}, & n = 1, N+1, 2N+1, \ldots \\ \vdots & \vdots \\ D^{(N-1)}, & n = N-1, 2N-1, 3N-1, \ldots, \end{cases} \tag{1}$$

where R represents the reference frame, $D^{(k)}$ $(k = 1, \ldots, N-1)$ denotes the k-th data-embedded frame in the spatial domain, and N is the total number of frames allocated per packet.



R : Reference image frame
$D^{(1)}, D^{(2)}, D^{(3)}$: Data-embedded image frame
N : The number of allocated frame sequences

FIGURE 2. The proposed frame packet structure when $N = 4$.

Figure 2 illustrates the frame packet structure with a length of four, that is, $N = 4$. Note that the image frames $D^{(1)}$, $D^{(2)}$, and $D^{(3)}$ correspond to data-embedded frames and different shades indicate that distinct data are embedded in every frame. The transmission of the reference frame will help in the subtraction data retrieval process at the receiver. Assuming that the data are recovered successfully at the receiving end, it seems that the data rate of the Video-DFC system can be increased significantly by setting a large value for N, that is, by transmitting a large number of data frames per packet.

B. DATA EMBEDDING

Prior to embedding data into a frame, the input video I_t is continuously split into frames based on the frame packet structure with each frame cycle ranging from frame 0 to frame $N-1$. Therefore, we can represent one cycle of I_t as a $P \times Q$ spatial-domain video in the form of a 3D matrix as

$$I_t = \left[I_t^{(0)}, I_t^{(1)}, \ldots, I_t^{(N-1)}\right]. \tag{2}$$

Here, one cycle of the video I_t consists of a reference frame $I_t^{(0)}$ and $N-1$ data-embedded frames $\left(\text{i.e., } I_t^{(1)}, \ldots, I_t^{(N-1)}\right)$. As the data embedding is performed in the spectral domain, each image frame in (2) is transformed to the corresponding spectral-domain frame, namely

$$I_F = \left[I_F^{(0)}, I_F^{(1)}, \ldots, I_F^{(N-1)}\right], \tag{3}$$

through the DCT operation. Since the first frame in the frame packet is a reference frame, it is not data-embedded. Without loss of generality, the k-th frame $(k = 1, 2, \ldots, N-1)$ can be transformed into the spectral domain as

$$\begin{aligned} I_F^{(k)} &= C \cdot I_t^{(k)} \\ &= \left[C \cdot i_{t_1}^{(k)}, C \cdot i_{t_2}^{(k)}, \ldots, C \cdot i_{t_Q}^{(k)}\right], \end{aligned} \tag{4}$$

where $i_{t_q}^{(k)} (q = 1, 2, \ldots, Q)$ is the q-th column vector of the k-th spatial-domain frame and C is the 1D-DCT matrix [24].

Similarly, the data to be embedded, b, are split into $N-1$ frames as

$$b = \left[b^{(1)}, b^{(2)}, \ldots, b^{(N-1)}\right], \tag{5}$$

where $b^{(k)}$ is the data embedded in the k-th frame. Then, the data are channel coded and modulated to produce the modulated data $d = \left[d^{(1)}, d^{(2)}, \ldots, d^{(N-1)}\right]$. The data $d^{(k)}$ for the k-th frame can be written as

$$d^{(k)} = \left[d_1^{(k)}, d_2^{(k)}, \ldots, d_Q^{(k)}\right], \tag{6}$$

where $d_q^{(k)}$ is the q-th column vector of the frame $d^{(k)}$, given as

$$d_q^{(k)} = \left[d_q^{(k)}(1), d_q^{(k)}(2), \ldots, d_q^{(k)}(L)\right]^T, \tag{7}$$

where L is the total number of data symbols per column [24].

Considering the power allocation in data symbols [24], the vector $d^{(k)}$ is transformed into $s^{(k)}$ $\left(= \left[s_1^{(k)}, s_2^{(k)}, \ldots, s_Q^{(k)}\right]\right)$, where $s_q^{(k)}$ is calculated as

$$s_q^{(k)} = X_{\text{amp}}^{(k)} \cdot d_q^{(k)} = \left(\alpha^{(k)} \sqrt{P_{\text{avg}}^{(k)}}\right) \cdot d_q^{(k)}. \tag{8}$$

Here, $s_q^{(k)}$ $\left(= \left[s_q^{(k)}(1), s_q^{(k)}(2), \ldots, s_q^{(k)}(L)\right]^T\right)$ is the q-th column vector of the k-th data-embedded frame, $X_{\text{amp}}^{(k)}$ is the scaling value of data, $\alpha^{(k)}$ $(0 < \alpha < 1)$ is a proportionality constant that determines $X_{\text{amp}}^{(k)}$, and $P_{\text{avg}}^{(k)}$ denotes the average power of the data-embedded region in the k-th image frame $I_F^{(k)}$. Finally, the data embedded in the k-th image frame $I_F^{(k)}$ are represented as

$$s = \left[s^{(1)}, s^{(2)}, \ldots, s^{(N-1)}\right]. \tag{9}$$

The data matrix embedded in the image frame $I_F^{(k)}$ has dimensions $P \times Q$ and can be represented as $X^{(k)} = \left[X_1^{(k)}, X_2^{(k)}, \ldots, X_Q^{(k)}\right]$. The q-th column vector of the k-th frame, denoted as $X_q^{(k)}$, can be expressed as

$$X_q^{(k)} = \left[\underbrace{0}_{1\times S} \quad \underbrace{\left(s_q^{(k)}\right)^T}_{L} \quad \underbrace{0}_{1\times(P-S-L)}\right]^T, \tag{10}$$

where S is the start pixel of the data symbol. The data matrix $X^{(k)}$ is then embedded into the spectral-domain image $I_F^{(k)}$ using the addition allocator operation as follows:

$$D_F^{(k)} = I_F^{(k)} + X^{(k)}. \tag{11}$$

To display the resulting image on the screen, the spectral-domain image $D_F^{(k)}$ is converted back to the spatial domain by performing an IDCT operation. This results in the data-embedded image in the spatial domain, expressed as follows:

$$D^{(k)} = C^T \cdot D_F^{(k)}, \tag{12}$$

where C^T is the transpose of the DCT matrix C.

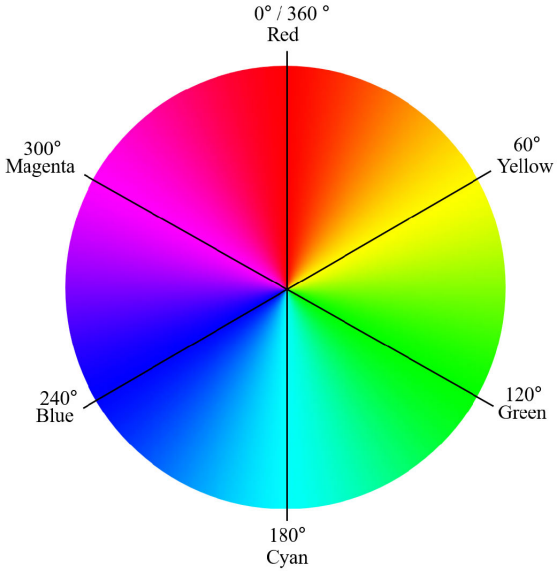


FIGURE 3. Hue circle.

C. 4-POINT BLOCK PATTERN

In this section, a 4-point block pattern is proposed to achieve synchronization between video frames and precise extraction of the image region. The block pattern is designed based on a complementary color code, which employs complementary color relationships to transmit data in a user-friendly and non-obtrusive manner. That is, if we rapidly switch between complementary colors at a high refresh rate, the human eye perceives two colors as a combined white color. In Fig. 3, the hue circle is presented, where each angle corresponds to a specific color. Colors located 180° apart from each other on the circle have a complementary relationship. The red (R), green (G), and blue (B) values for each color are presented in Table 1. It can be observed that when R, G, and B values and their complementary colors are added together, the resulting value is white, with RGB values of (255, 255, 255). Therefore, by displaying complementary colors with a high refresh rate, the human eye perceives two colors as white, enabling user-unobtrusive delivery of data.

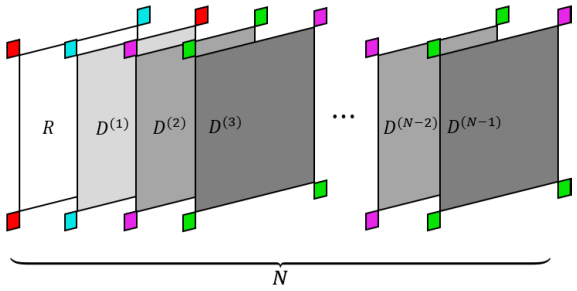
TABLE 1. Color value for each angle.

Angle	Hue	Color value		
		Red	Green	Blue
0	Red	255	0	0
60	Yellow	255	255	0
120	Green	0	255	0
180	Cyan	0	255	255
240	Blue	0	0	255
300	Magenta	255	0	255

Table 2 illustrates the indexing relationship between the colors of the block pattern and the frames, considering a single frame packet of length N. If $N = 4$, the four colors listed in the table are chosen in the same serial order to represent the block pattern. Specifically, the reference frame R is assigned the red color, whereas the first data-embedded frame is assigned the cyan color, and the second and third data-embedded frames are assigned magenta and green colors, respectively. When N exceeds 4, the block color pattern repeats in a specific order: red, cyan, magenta, green, magenta, green, and so on. This ordering is selected to ensure a minimum angle difference of 120° between the current color pattern and the next one, thus maximizing the color contrast. Additionally, N should be an even number to ensure that the block color pattern angle difference remains at a minimum of 120° when the frame packets are repeatedly transmitted. In this way, assuming the monitor refresh rate of 60 Hz, the designed block pattern can achieve data transmission without any observable artifacts on the screen.

TABLE 2. Block color pattern according to the frame.

Block color pattern	Frame
Red	R
Cyan	$D^{(1)}$
Magenta	$D^{(l)}; l \in \{2k\}, 1 \leq k \leq N/2-1, N \geq 4$
Green	$D^{(l)}; l \in \{2k+1\}, 1 \leq k \leq N/2-1, N \geq 4$



R : Reference image frame
$D^{(1)}, D^{(2)}, \cdots, D^{(N-1)}$: Data-embedded image frame
N : The number of allocated frame sequences

FIGURE 4. Illustration of the color pattern in one frame packet.

Figure 4 presents the 4-point block pattern for a single frame packet. The pattern comprises four colors, that is, red, cyan, magenta, and green. As mentioned above, we can observe that the reference frame has the red block pattern and the first data-embedded frame has the cyan block pattern, maintaining a 180° angle difference. Subsequently, the second data-embedded frame has the magenta block pattern, differing by 120°, and so on. This approach also helps in the frame packet synchronization at the receiver, that is, to distinguish between the reference frame and the data-embedded frames. Second, we can observe that the same colors are inserted at each of the three vertices (top left, bottom left, and bottom right) of the frame. The block color pattern for the fourth vertex (top right) is chosen to be the complementary color to the other three corners. This design is reminiscent of the finder pattern of conventional QR codes [25] and functions as a reference point for the camera to decode the correct orientation of the block pattern. Given that there exist three reference corners, namely top left, bottom

left, and bottom right, the block pattern can be read in a consistent direction, irrespective of its rotation angle.

Altogether, the 4-point block pattern serves the following functions:

- *Frame packet synchronization:* Frame synchronization implies distinguishing the reference frame from the data-embedded frames. As the reference frame has the 3-points as red and the 4th point as cyan, we can distinguish it from the rest of the data-embedded frames.
- *Rotation correction:* It helps in detecting the accurate rotation of the captured image. Therefore, even if the image is captured upside-down or in any rotation, the captured image can be accurately rotated back to its original position based on the 4-point block pattern.
- *Precise image region extraction:* In addition, the four points are exactly located at the corners of the image frames. Hence, at the receiver, they help in the precise extraction of image region for decoding the data. That is, it allows us to crop the relevant image region, which is further discussed in Sec. IV.

D. DATA DETECTION

When a camera captures a video on a display screen, a lot of noise is generated during the process. Assuming that the noise introduced is constant across the reference frame and the data-embedded frames, the frames captured from the camera can be represented as

$$\begin{aligned} Z_t^{(0)} &= I_t^{(0)} + N_t \\ Y_t^{(k)} &= D^{(k)} + N_t = C^T \cdot D_F^{(k)} + N_t, \end{aligned} \tag{13}$$

where $Z_t^{(0)}$ is the received reference frame, $Y_t^{(k)}$ is the received data-embedded frame, and N_t is the D2C channel noise at spatial domain. The received frames are then transformed into the spectral domain using the DCT as

$$\begin{aligned} Z_F^{(0)} &= C \cdot I_t^{(0)} + C \cdot N_t = I_F^{(0)} + N_F \\ Y_F^{(k)} &= C \cdot D^{(k)} + C \cdot N_t = D_F^{(k)} + N_F \\ &= I_F^{(k)} + X^{(k)} + N_F, \end{aligned} \tag{14}$$

where $I_F^{(0)}$ and $D_F^{(k)}$ represent the spectral domain representations of the original reference frame and k-th data-embedded frame. The effect of noise in spectral domain is represented by N_F.

The k-th data matrix can then be obtained using a subtraction data retrieval process that subtracts the reference frame from the data-embedded frame as [24]

$$\hat{X}^{(k)} = Y_F^{(k)} - Z_F^{(0)}. \tag{15}$$

Since $\hat{X}^{(k)} \left(= \left[\hat{X}_1^{(k)}, \hat{X}_2^{(k)}, \ldots, \hat{X}_Q^{(k)}\right]\right)$ is the data matrix to be estimated, which is embedded in the image frame $I_F^{(k)}$, the q-th column data of the k-th frame are given as

$$\hat{s}_q^{(k)}(l) = \hat{X}_q^{(k)}(S + l, q) \quad l = 1, 2, \ldots, L. \tag{16}$$

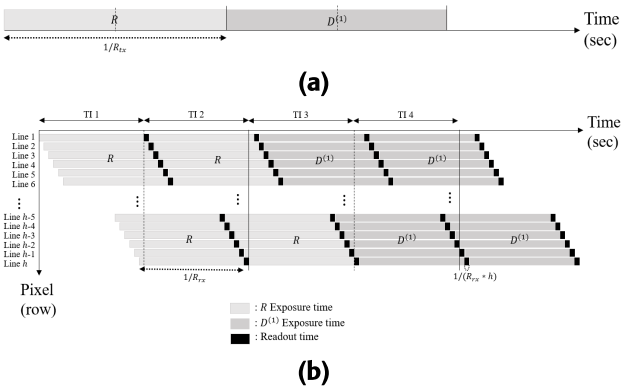


FIGURE 5. Exposure and readout time in rolling shutter effect. (a) Transmitter. (b) Receiver.

Finally, the estimated data symbol of the k-th frame can be obtained in the matrix form as

$$\hat{s}^{(k)} = \left[\hat{s}_1^{(k)}, \hat{s}_2^{(k)}, \ldots, \hat{s}_Q^{(k)}\right]. \tag{17}$$

Here,

$$\hat{s}_q^{(k)} = \left[\hat{s}_q^{(k)}(1), \ldots, \hat{s}_q^{(k)}(L)\right]^T, \tag{18}$$

is the estimated data symbol vector at the q-th column of the k-th image frame, where the whole estimated data symbol is given as

$$\hat{s} = \left[\hat{s}^{(1)}, \hat{s}^{(2)}, \ldots, \hat{s}^{(N-1)}\right]. \tag{19}$$

Then, after channel decoding, the output bits of the k-th frame are demodulated and the estimated data bits are give as

$$\hat{b} = \left[\hat{b}^{(1)}, \hat{b}^{(2)}, \ldots, \hat{b}^{(N-1)}\right]. \tag{20}$$

III. EFFECT OF ROLLING SHUTTER

Many smartphones these days are equipped with high-performance complementary metal-oxide-semiconductor image sensor arrays that mostly utilize rolling shutters. Unlike global shutters that expose all pixels simultaneously, rolling shutters expose rows sequentially from top to bottom by setting different exposure times for each row. This sequential exposure can cause geometric distortion in moving objects or videos, commonly known as the rolling shutter effect [26]. Owing to the different exposure times, parts of the image may appear stretched or compressed, resulting in a distorted image.

Considering the distortion caused by the rolling shutter effect, if the refresh rate of the transmitter (R_{tx}) is faster than the frame rate of the receiver (R_{rx}), the camera may not capture the transmitted image correctly. Therefore, to obtain an accurate image, at least two images need to be captured at the receiver per transmitted image, satisfying the Nyquist-Shannon sampling theorem as follows:

$$R_{rx} \geq 2R_{tx}. \tag{21}$$

Figure 5 illustrates the effect of rolling shutter on the captured image frames when $N = 2$. It shows the exposure and readout

FIGURE 6. Received reference frame. (a) Sampling ratio: $R_{rx}/R_{tx} = 120/60 = 2$. (b) Sampling ratio: $R_{rx}/R_{tx} = 120/40 = 3$.

time resulting from the rolling shutter effect. Figure 5a shows the transmission of a new image frame every $1/R_{tx}$ seconds, whereas Fig. 5b shows the time interval (TI) every $1/R_{rx}$ seconds. We set the total number of row lines of the captured image frame (h) at 1080, the refresh rate of the transmitter at 60 Hz, and the frame rate of the receiver at 120 fps. Then, the exposure time of the received frame is $1/R_{rx} = 1/120$ s, and the readout time is $1/(R_{rx} \times h) = 1/(120 \times 1080)$ s. As the R_{rx} increases, the readout time decreases. However, the delay introduced by the readout time can result in a time lag between the exposures of consecutive frames.

Owing to this effect, the exposure time varies for each row, causing image frame distortion due to time delay. Please remind that (R_{rx}/R_{tx}) is 2 so that two images are captured. As shown in Fig. 5b, the time intervals for the received R frame are TI 1 and TI 2, and TI 3 and TI 4 for the received $D^{(1)}$ frame. For a specific received image frame, when using the image frames from the first time interval, for example, TI 1 and TI 3, the previous frame pattern area before exposure still remains at the bottom of the frame. As a result, it is difficult to determine whether these frames are reference frames or data-embedded frames. Therefore, data recovery may not be performed correctly. In other words, owing to the rolling shutter effect, frames from first time interval got affected by previous patterns and are not suitable for data decoding. In addition, note that there is distortion from pre-existing D2C channel noise, as shown in (13). Hence, we suggest that the image frames from the second time interval, such as TI 2 and TI 4, should be used as they exhibit relatively lower levels of distortion.

IV. FRAME PACKET SYNCHRONIZATION

To decode the data, a process of frame packet synchronization is performed. For that, we first need to locate the reference frame in which the red block pattern was inserted during transmission. This frame serves as a reference point for the decoding process and is crucial for accurately extracting the embedded data from subsequent data-embedded frames. Figure 6 illustrates a received reference frame at TI 2 when the frame packet structure is $RD^{(1)}$. In this experiment, the frame rate was fixed at 120 fps, while the refresh rate was varied to observe the changes in the block pattern. The sampling ratio in Fig. 6a is set to 2 in accordance with (21). As mentioned previously, when two complementary colors are alternately displayed on the screen at a rate higher than 60 Hz, the human vision system cannot observe any visual difference. Therefore, the 4-point block pattern was recognized as white in Fig. 6a. As shown in Fig. 4, the reference frame should have three red corners, excluding cyan at the top right. However, as shown in Fig. 5, owing to the time delay that occurred due to the readout time, bottom part of the image does not get exposed, as observed in Fig. 6a. In Fig. 6b, the sampling ratio is set to 3. That is, three frames are captured per transmitted frame. As expected, with more frames captured, the time delay due to readout time is less significant, resulting in a received image closer to the reference frame. Therefore, we can observe that the three corners of the block pattern are red and one is cyan as shown in Fig. 4. This is a more advantageous setup for data recovery as the extracted image (Fig. 6b) is clearer compared to Fig. 6a. If we further reduce the refresh rate, the flickering of the block pattern becomes more conspicuous.

It is important to understand this trade-off. In order to achieve perfect frame packet synchronization and accurately recognize the rotation of the block pattern, it may become necessary to acquire precise 4-point color information, which can be achieved by taking at least three or more images per frame. However, to ensure visual quality, the display's refresh rate must be no less than 60 Hz under 120 fps receiver camera

capture rate. In consideration of this, further experiments were conducted with a fixed sampling rate of capturing two images per transmitted frame. That is, considering the rolling shutter effect, the top points (top left and top right) with little time delay can be used to find the reference frame.

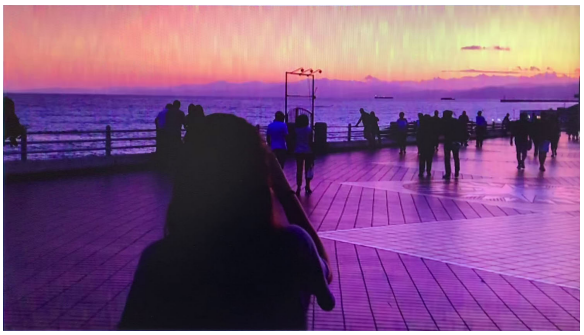


FIGURE 7. Distortion corrected reference frame.

Once the frame packet synchronization and rotation correction have been achieved, the $2N$ received frames are accurately aligned. Then, we have to take the second image frame of each pattern and crop the relevant image region. Figure 7 depicts a reference frame that has been corrected from distortion, which has been obtained from Fig. 6a. This has been done using a series of steps. Firstly, we obtain a difference image by subtracting the consecutive frames, sequentially for each channel. Next, we convert the resulting image to a one-channel grayscale image and binarize it to reveal the block pattern. Subsequently, a specific algorithm is employed to identify and draw four minimum-size boxes that encloses the four points of the block pattern. The bounding boxes provide us with critical information, including the top-left coordinates of the image, its width, and height. The order of the block pattern's four corners is sorted clockwise via the top left coordinates provided by the bounding box. We then determine the center point of the bounding box. The center point is used to determine whether the frame is a reference frame or a data-embedded frame and to identify the frame packet structure. Finally, a distortion-corrected image can be acquired by cropping the image based on the coordinates provided by the bounding box. This cropped image is then used for data decoding, which allows for the estimated output bits to be obtained. In addition, please note that the more often the reference frames are repeated, the less flicker is visible on the screen. That is, if we use $RD^{(1)}$ instead of $RD^{(1)}D^{(2)}D^{(3)}$, we can observe that $RD^{(1)}$ is repeated twice in the same duration as of $RD^{(1)}D^{(2)}D^{(3)}$. Hence, frame packet $RD^{(1)}$ will show less flicker than the frame packet $RD^{(1)}D^{(2)}D^{(3)}$. Therefore, the design of the frame packet structure should be carefully done based on the fact that how much visual artifacts are tolerable on the screen.

V. REAL-WORLD BASED EVALUATION AND RESULTS

Table 3 lists the equipment and experiment setup for the proposed Video-DFC scheme. A Samsung monitor with a resolution of 1920 × 1080p was employed to transmit the video, which was then captured by an Apple iPhone XS Max camera with a resolution of 1080p (Full HD) mounted on a tripod, as depicted in Fig. 8. From Table 3, we can see that the transmission rate, R_{tx}, was set at 60 Hz and the frame rate, R_{rx}, was set at 120 fps, allowing for the camera to capture two frames per transmitted frame. To mitigate data loss resulting from D2C channel, an error correction code, specifically turbo coding, was employed. Moreover, the experiment was conducted indoors under ambient LED lighting conditions, with a fixed display-camera distance of 50 cm. In addition, no angle distortion is assumed. That is, as shown in Fig. 8, the actual laboratory experimental environment consists of the camera angle (A) and display rotation (R) fixed at 0° and the distance (D) is 50 cm.

TABLE 3. Default experiment setup.

Parameter	Specification
Transmitter display spec.	Samsung monitor, 51 cm × 29 cm, Res. 1920 × 1080, Refresh rate 60 Hz
Receiver camera spec.	Apple iPhone XS Max, 12 MP (1920 × 1080), Frame rate 120 fps
Channel coding scheme	Turbo coding
Modulation format	BPSK
Test location	Indoor
Lighting condition	Ambient light
Distance (D)	50 cm (tripod mounted)
Camera angle (A)	0°
Display rotation (R)	0°

FIGURE 8. Laboratory experiment setup.

In comparison to the work presented in [24], which utilized still images for data transmission, this study adopts videos for DFC. As a result, we have introduced a novel frame packet structure specifically tailored for video-DFC. Additionally, considering the rolling shutter effect, we have devised a 4-point block pattern to achieve frame synchronization and accurately extract the image region. The key advantage of video-DFC lies in its capacity to transmit a larger volume of data by embedding distinct input data into different frames. By employing videos as the medium for data transmission and incorporating our novel frame packet structure and 4-point block pattern, we aim to significantly enhance the data rate of DFC, offering a valuable contribution to this research area.

FIGURE 9. Visual quality comparison according to PSNR. (a) Original. (b) 30 dB. (c) 40 dB. (d) 50 dB.

A. PERFORMANCE ACCORDING TO THE TYPES OF FRAME PACKET STRUCTURE

The first and most important thing about the Video-DFC is that the quality of the video displayed on the screen should be good. That is, the data should remain unobtrusive and should not interfere the normal viewing experience. In other words, the display should faithfully perform its primary task of delivering the video content. Therefore, first, we analyze the quality of the video displayed on the electronic screen. In the current experiment, we used the 'Beach.mp4' [27] video for data embedding. Figure 9 depicts the first frame from the video. The video quality is evaluated using the metric peak signal-to-noise ratio (PSNR). PSNR is a measure of the difference between the original video and a data-embedded version of the video, expressed in decibels (dB). The higher the PSNR, the less difference between the two videos, and hence, the better the quality of the data-embedded video. Therefore, we must check at what PSNR values, the embedded data does not distort the image frame. Figure 9a shows the original image frame without any data embedding. On the other hand, Fig. 9b, 9c, and 9d illustrate the data-embedded images for different PSNR values. It can be inferred from the figures that the severity of image quality distortion increases as the PSNR decreases. Moreover, at 40 dB and above PSNR, the image frame does not exhibit any noticeable artifacts. Therefore, we can say that the embedded data in our proposed Video-DFC approach becomes unobtrusive over 40 dB PSNR.

Table 4 presents the parameters associated with the frame packet structures used in the conducted experiments. We can observe that two kinds of frame packet structures were used in the experiment. First, we conducted the experiment using two frames, that is one reference and one data-embedded frame. After that, we performed the experiment by increasing the frame packet size to $N = 4$, that is, by using one reference and three data-embedded frames. In both experiments, the 'Beach.mp4' video with a resolution of 324×576 was utilized. Regarding the number of embedded data bits, each R, G, and B channel of the image frames had 200 data bits embedded, totaling to 600 and 1800 data bits per frame

TABLE 4. Experiment parameters for frame packet structure.

Parameter	$RD^{(1)}$	$RD^{(1)}D^{(2)}D^{(3)}$
No. of frames per packet structure	$N = 2$	$N = 4$
Input video	Beach.mp4	Beach.mp4
Input video size	324×576	324×576
Input video frame rate	60 Hz	60 Hz
No. of binary data bits per channel/total transmitted bits	200/600	200/1800
No. of turbo encoded data bits per channel/total encoded bits	1018/3054	1018/9162
Lower sub-band position	row 5 to 15	row 5 to 15
No. of bits used with/without soft-decision decoding	200/600	600/1800

packet structure, for $N = 2$ and $N = 4$, respectively. The number of encoded data bits increased after the turbo coding process. The encoded data bits were eventually embedded in the low sub-bands (rows 5 to 15) of the image frames, considering the performance and frequency characteristics of the spectral-domain image frame [24]. Soft-decision decoding was employed to reduce the overall system error rate.

Figure 10 displays the BER according to the frame packet structure when soft-decision decoding is not applied. In this experiment, different data bits were embedded in each of the R, G, and B channels of the data-embedded frames. Figure 10a shows the result when a single data-embedded frame was transmitted. In other words, a total of 600 data bits were transmitted per frame packet, that is, $N = 2$. On the other hand, Fig. 10b illustrates the performance when the frame packet structure size was $N = 4$. That is, a total of 1800 data bits were embedded per frame packet. It can be observed that all channels have a BER of zero until 30 dB PSNR. As the PSNR increases further, the error rates of the R and B channels increase progressively, whereas G maintains a BER of zero up to 40 dB, in both experiments.

Notably, the green channel is the least and the blue channel is the worst affected. Despite embedding data within the same spectral range— specifically, spanning rows 5 to 15, in all three R, G, and B channels, it seems that the B channel frequency coefficients in this region are very different compared to the R and G channel, making them more vulnerable to noise and distortion. This can be attributed to the fact that the color of the blue component of LED displays tend to decrease as the display ages, which arises from multiple factors [28]. First, the blue LED utilized in LED displays is constructed using gallium nitride-based semiconductor components, which can develop defects over time as they are driven for extended periods. These defects, commonly known as dislocations, compromise the structure of the semiconductor component and create defects in the material that maintain the color of the emitted light. These issues ultimately contribute to a change in the color of the blue LED light. Secondly, the blue LED has a shorter wavelength compared to the red and green LEDs. Consequently, it is more sensitive to certain wavelengths, which contributes to a reduction in its wavelength and causes a color shift. Other factors such as the color distribution of the original video

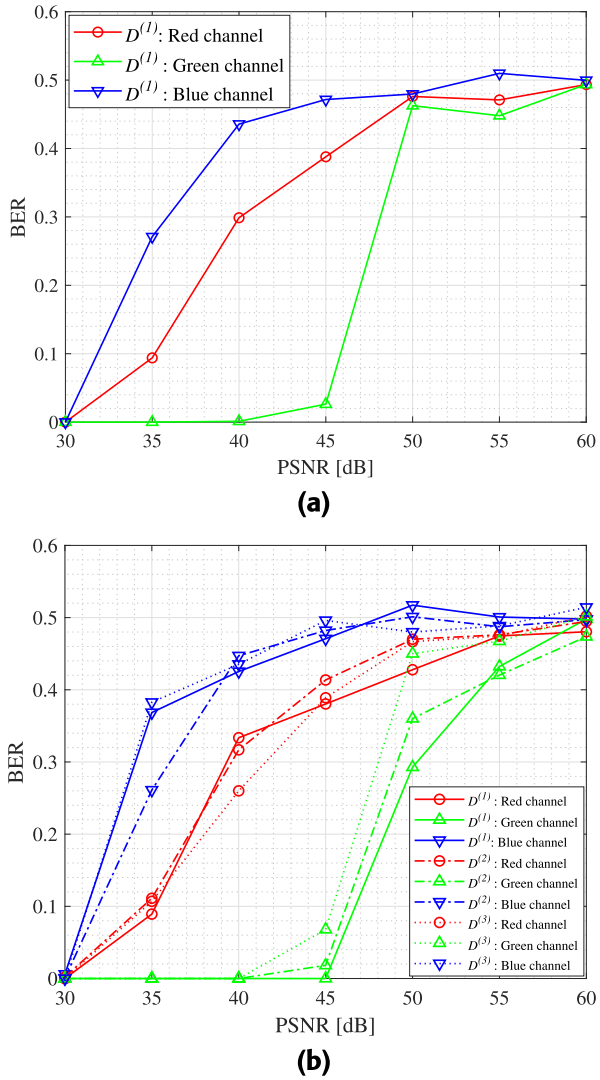


FIGURE 10. BER performance with respect to PSNR when soft-decision decoding is not applied. (a) $RD^{(1)}$. (b) $RD^{(1)}D^{(2)}D^{(3)}$.

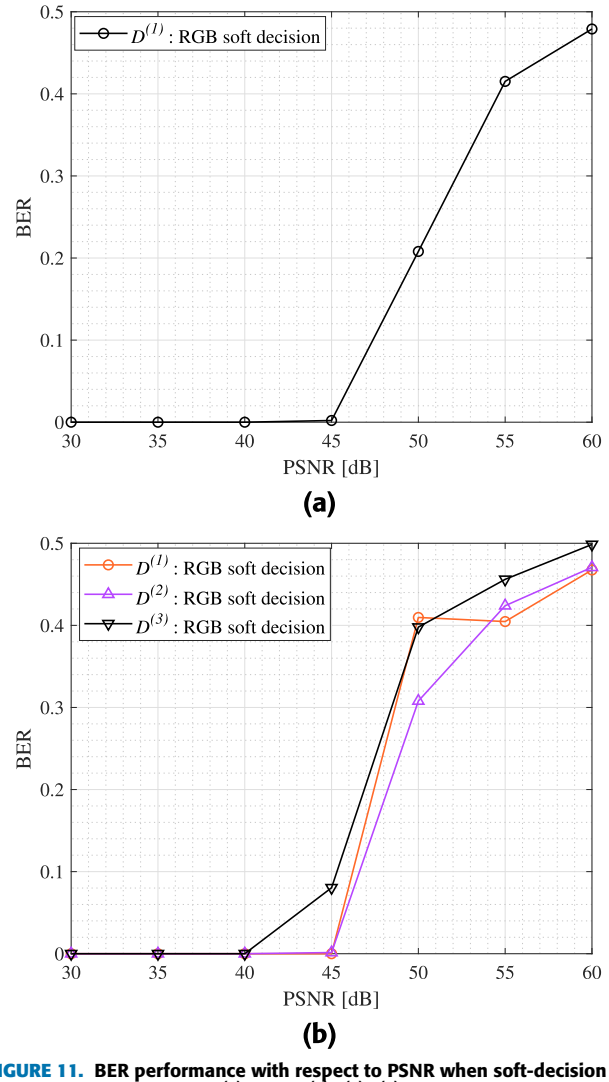


FIGURE 11. BER performance with respect to PSNR when soft-decision decoding is applied. (a) $RD^{(1)}$. (b) $RD^{(1)}D^{(2)}D^{(3)}$.

and the image transformation techniques used, that is, DCT and IDCT, may also contribute to the different error rates observed in the three channels.

Figure 11 depicts the BER with respect to the frame packet structure when soft–decision decoding is applied. In this experiment, the same data were embedded in each of the R, G, and B channels, resulting in 200 distinct data bits embedded in a single data-embedded frame when $N = 2$, and 600 distinct data bits in three data-embedded frames when $N = 4$. In other words, the number of input data bits was reduced to one-third as compared to the previous experiment, where soft decision decoding was not employed. Owing to the same data being embedded in all three channels of an image frame, we can observe that the BER performance has improved. Specifically, in Fig. 11a, the error rate remained zero up to 45 dB. Similarly, in Fig. 11b, the error rate remained zero up to 45 dB, except for the third data-embedded frame. Note that soft-decision decoding provides a more robust estimate of the transmitted bits compared to non-soft-decision decoding, which leads to a lower overall error rate. This is because embedding the same data in all three R, G, and B channels of an image adds redundancy to the data. This redundancy helps to reduce the error rate by providing additional information to the decoder. In addition, the difference in BER between soft-decision and non-soft-decision decoding is more pronounced at lower values of PSNR. This is because the received signal is more noisy at lower values of PSNR, and soft decision decoding is better at handling noise. Overall, we can observe that soft decision decoding is a better approach for reducing BER. However, it should be noted that adding redundancy will reduce the overall data rate of the system, as shown next.

The ADR gives an idea about how much data can be transmitted per unit time. It refers to the maximum number of transmitted data bits per second without error, defined as

$$\text{ADR} = (1 - \text{BER})D_{\max}, \tag{22}$$

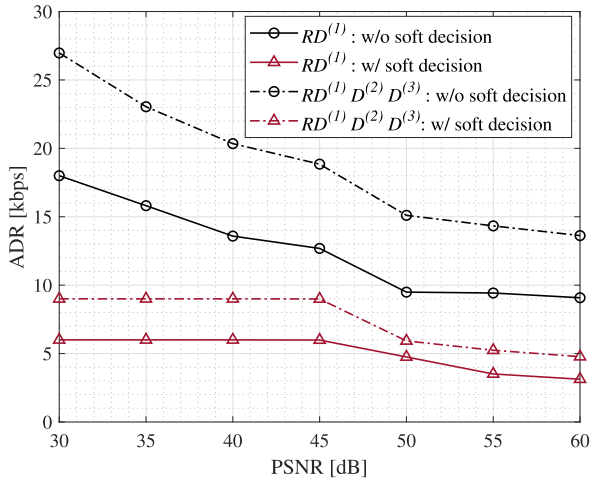


FIGURE 12. Achievable data rate comparison for with and without soft-decision decoding.

TABLE 5. Experiment parameters for different input videos.

Parameter	Specification
Input video	Bridge.mp4 [29] / People.mp4 [30]
Input video size	324×576, 60 Hz
Frame packet structure	$RD^{(1)}D^{(2)}D^{(3)}$
No. of binary data bits per channel	200
No. of turbo encoded data bits per channel	1018
Lower sub-band position	rows 5 to 15

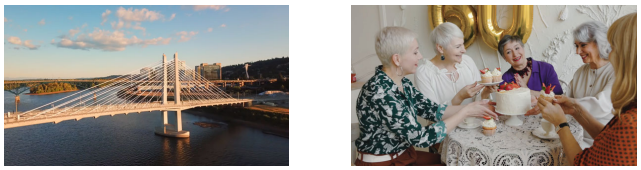


FIGURE 13. First frame of different input videos. (a) Bridge.mp4 [29]. (b) People.mp4 [30].

where $D_{\max}$ is the maximum number of transmitted data bits per second, given as

$$D_{\max} = \frac{\kappa L_{ch}(N-1)}{N/f_{\text{refresh}}} \text{bps}. \tag{23}$$

Here, L_{ch} is the number of data bits per channel, f_{refresh} is the refresh rate of the display, κ ($\in \{1, 3\}$) is the receiver parameter, which describes whether the soft-decision decoding is applied or not. If κ is set to 1, it indicates that soft-decision decoding is applied. On the other hand, if κ is set to 3, it indicates that soft-decision decoding is not applied. Lastly, $N-1$ is the number of data-embedded frames allocated per packet. In each data-embedded frame, a total of $L_{ch} \times \kappa$ data bits were embedded. Note that data embedding pattern depends on the presence or absence of soft-decision decoding. If soft-decision decoding is applied, the same data are embedded in the R, G, and B channels. Conversely, if soft-decision decoding is not used, distinct data are embedded in each channel.

Figure 12 presents the ADR as a function of PSNR for four cases illustrated in Fig. 10 and 11. We can see that ADR is higher when soft-decision decoding is not employed. Particularly, it is observed that ADR reaches around 27 kbps when the PSNR of $RD^{(1)}D^{(2)}D^{(3)}$ is 30 dB. Therefore, by comparing the frame packet structures, it is confirmed that increasing the number of data-embedded frames and not using soft-decision decoding is beneficial in achieving a higher ADR. However, there is a trade-off as mentioned before.

Now, we know that the highest ADR is achieved at 30 dB when the frame is $RD^{(1)}D^{(2)}D^{(3)}$ and soft-decision decoding was not applied. For the same experiment, the BER is also zero at 30 dB (cf., Fig. 10b). It means that 30 dB PSNR is ideal to transmit lots of data. However, even though we can transmit lots of data at 30 dB, looking at Fig. 9, we can observe that the displayed video on screen has some visual artifacts that may distract the normal viewer. Conversely, if we consider the case of 40 dB, we can get an ADR of 20.34 kbps. On the other hand, if we consider the case of 40 dB PSNR with soft-decision decoding, we can see that the BER performance is almost zero and the data-embedded video is also artifact-free. However, the ADR is reduced to 9 kbps due to soft-decision decoding. Hence, based on the required data rate, video quality, and BER, we have to choose the experiment parameters.

We can analyze another trade-off in this context. As seen in (22), there exists an inverse relationship between the BER and the maximum data rate $D_{\max}$. In simpler terms, higher data rates lead to a higher BER. Therefore, to increase the ADR, higher values of $D_{\max}$ are required. From Fig. 12, we observe that in the absence of soft-decision decoding, we still get a higher ADR than the soft-decision decoding case. This is despite the fact that non-soft-decision decoding gives rise to comparatively higher BER. This means that the effect of degradation due to BER is comparatively less than the effect of $D_{\max}$. In addition, as the PSNR increases, the slope of decrease in ADR is higher in case of non-soft-decision decoding. On the other hand, if we use soft-decision decoding, the slope of decrease in ADR is comparatively less. This slope comes from the degradation of the BER. Overall, we can see that even though non-soft-decision decoding gives us higher BER (cf. Fig. 10), the effect of increase in data rate overcomes the degradation caused by the BER, resulting in higher ADR. Moreover, this higher ADR can be used to further reduce the BER by using more powerful channel coding schemes. Therefore, increase in ADR is an outstanding feature of our proposed video-DFC.

B. PERFORMANCE ACCORDING TO DIFFERENT INPUT VIDEOS

The rationale for utilizing different videos in our study is to check the effect of different RGB distributions and scene changes. Table 5 presents the experiment setup. Two videos, namely 'Bridge.mp4' [29] and 'People.mp4' [30] have been selected in addition to 'Beach.mp4', all having a resolution of 324×576p. The proposed method was evaluated using the frame packet structure $RD^{(1)}D^{(2)}D^{(3)}$ when soft-decision decoding was applied. Figure 13 shows the first frame of

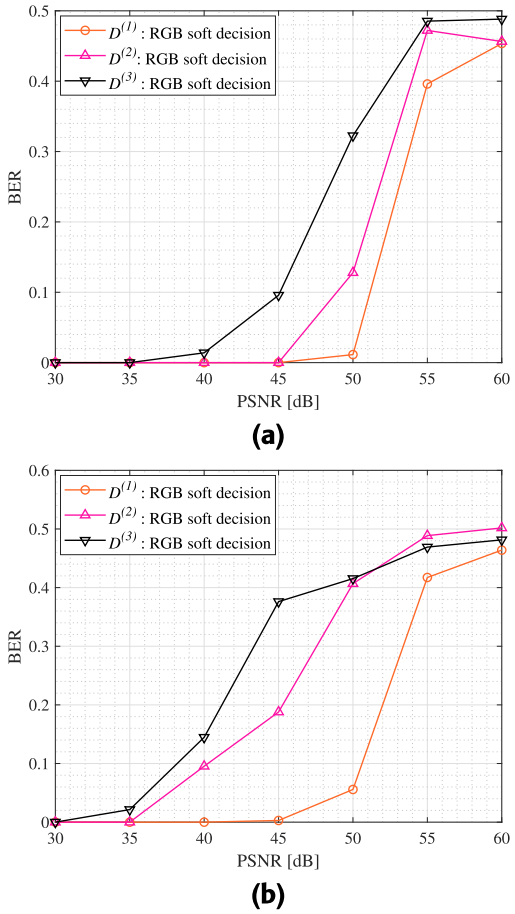


FIGURE 14. BER performance with respect to PSNR for two different input videos. (a) Bridge.mp4. (b) People.mp4.

the two videos. If we compare the quality of three videos, we found that the Beach.mp4 has simple colors and dominant shades of red. On the contrary, Bridge.mp4 contains a range of colors with blue dominant in some parts. Moreover, Beach.mp4 showcases a slow scene change between frames than Bridge.mp4, resulting in low complexity picture. Lastly, People.mp4 stands out as the most complex video. It features a wide variety of colors with RGB evenly distributed across each frame. Additionally, People.mp4 includes frequent scene changes between frames.

Figure 11b and 14 depicts the BER performance based on the PSNR for all the three videos used as input on the screen. To check the effect of RGB color distribution on the BER, let's examine the first data-embedded frames' performance of all the three videos. We can observe that the Beach.mp4 performs the worse among all the three cases. For instance, at 50 dB PSNR, $D^{(1)}$ frame of Beach.mp4 shows a BER of 0.4 compared to Bridge.mp4 and People.mp4, which show much lower BER, close to 0.1. This is attributed to the fact that even though Bridge.mp4 and People.mp4 have complex picture, they have better RGB color distribution, which makes it easier for them to transmit data reliably. On the other hand, in Beach.mp4, which has dominant frequency components of red, makes it difficult to transmit data reliably over the D2C link, resulting in higher BER. In other words, the RGB components in Bridge.mp4 and People.mp4 seem to serve as effective channel coefficients (as in an OFDM scheme). That is, color distribution in the image frames influences data transmission efficiency. For example, in the case of the Beach.mp4, where red dominates, the transmission of data using blue and green components becomes challenging. This deficiency in the blue and green components adversely affects the soft decision decoding process, manifesting in higher BER values. In contrast, videos like Bridge.mp4 and People.mp4, with more balanced RGB distributions, shows improved BER performance due to the presence of sufficient frequency components in all the three channels.

To understand the effect scene change, we have to observe the performance of $D^{(2)}$ and $D^{(3)}$, the second and third data-embedded frames. In case of Beach.mp4, we can see that the difference of BER performance between frames is not significant. This is due to very few scene changes between the frames. On the other hand, looking at Bridge.mp4 and People.mp4, the BER performance difference between frames is significant. Particularly, $D^{(2)}$ performs worse than $D^{(1)}$ and $D^{(3)}$ performs worse than $D^{(2)}$. This is because there is a lot of scene changes between frames in these videos. That is, the $D^{(3)}$ frame is more far away from the reference frame. Overall, our analysis underscores the intricate relationship between BER, RGB distribution, and scene changes and sheds light on how these factors interact and influence the overall performance of demodulation, particularly in the context of soft-decision decoding.

C. PERFORMANCE ACCORDING TO THE RESOLUTION OF THE CAMERA

At last, we performed the experiment using two different cameras. Table 6 presents the experiment conditions for different camera resolutions. The full high-definition (FHD) camera used in the experiment had a resolution of 1920 × 1080p, whereas the ultra-high-definition 4K (UHD) camera had a resolution of 3840 × 2160p. For the UHD camera experiment, the refresh rate and frame rate were set to 30 Hz and 60 fps, respectively. Therefore, two image frames were captured at the camera per transmitted frame.

TABLE 6. Experiment parameters for different camera resolutions.

Parameter	FHD	UHD (4K)
Horizontal pixels	1920	3840
Vertical pixels	1080	2160
Refresh rate	60 Hz	30 Hz
Frame rate	120 fps	60 fps
Input video	Beach.mp4	Beach.mp4
Input video size	324 × 576	324 × 576
Data embedding pattern	$RD^{(1)}D^{(2)}D^{(3)}$	$RD^{(1)}D^{(2)}D^{(3)}$
No. of binary data bits per channel	200	200
No. of turbo encoded data bits	1018	1018
Lower sub-band position	rows 5 to 15	rows 5 to 15

In Fig. 15, the BER performance is presented according to PSNR using cameras with two different resolutions.

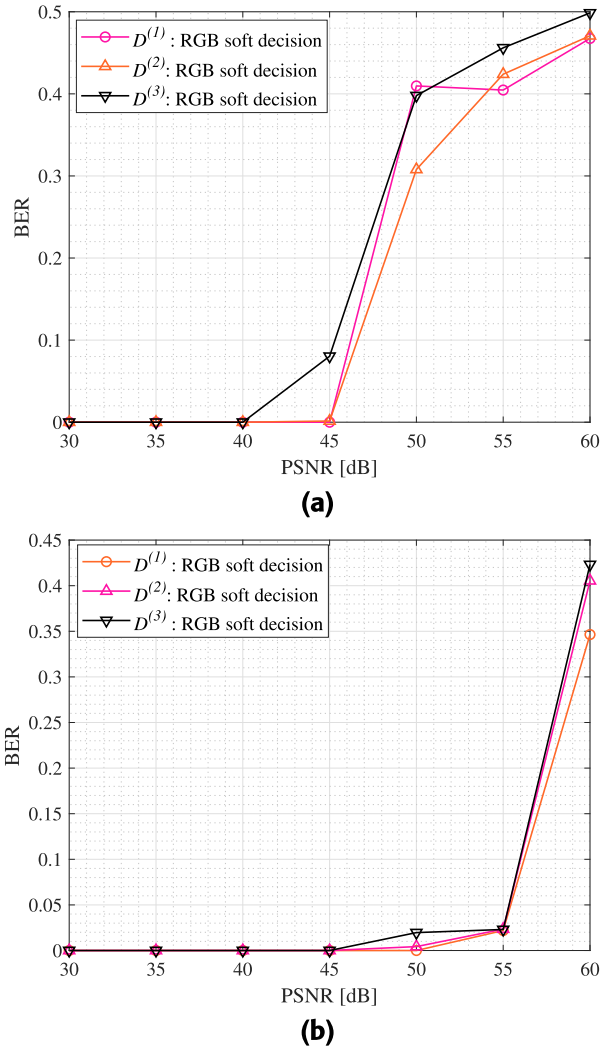


FIGURE 15. BER performance with respect to PSNR. (a) FHD. (b) UHD (4K).

Specifically, Fig. 15a is based on the results of using a FHD camera, whereas Fig. 15b is based on the results of using a UHD camera. The results show that using a high-resolution camera provides better performance. For example, at the same PSNR value of 50 dB, Fig. 15a exhibits an error rate of 0.3 or higher in all data-embedded frames, whereas Fig. 15b shows that the error rate is close to zero. This is because cameras with higher resolution can capture more detailed information about the scene, including finer details and textures. This increased level of detail help improve the accuracy of the data decoding.

Here, we want to compare our work with the recent state-of-the-art work proposed in [14]. Table 7 presents the quantitative comparison of both works in terms of certain common parameters. However, note that both papers present different approaches to hidden communication between displays and smartphones. They use different techniques and experiment scenarios to obtain their results. This means that the results of the two papers are not directly comparable. Nevertheless, both works provide valuable insights into the design of D2C communication and their results can be used in the development of future D2C communication systems.

TABLE 7. Comparison of the proposed work with [14].

Parameter	Proposed work	[14]
Data Transmission Rate	27 kbps	7.68 kbps
Distance	50 cm	30 cm
Display resolution	1920 × 1080	16 × 16
Display refresh rate	60 Hz	150 Hz
Camera frame rate	120 fps	30 fps
Camera resolution	1920 × 1080	3840 × 2160
Key feature	High data rate	Real-time

VI. CONCLUSION

In this study, we proposed a novel video-based display-to-camera (D2C) communication system and performed practical real-world experiments. To the best of authors knowledge, it is the first trial in DFC research field. Our proposed method involves embedding and extracting data from the spectral domain of each video frame, which is called Video-DFC. We designed the frame packet structure composed of the reference frame and the data-embedded frame. In addition, we also inserted a 4-point block pattern at the four vertices of the transmitted image frames to enable accurate data-embedded region extraction and frame-to-frame synchronization. We also investigated the sampling rate of the transceiver to account for the rolling shutter effect of the camera. In the experiments, we evaluated the BER and ADR of the Video-DFC according to the frame packet structure. As a result, we found that applying soft-decision decoding reduces the overall error rate and increasing the number of data-embedded frames can obtain more ADR. Furthermore, from BER experiments with respect to various input videos, we deduced that the error rate is lower for input videos with evenly distributed RGB colors and few changes in motion. Finally, we demonstrated that using a higher resolution camera results in better performance. Overall, our proposed Video-DFC system shows promising results for an efficient data transmission via video contents.

REFERENCES

[1] Statistica. *Number of Digital Video Viewers Worldwide From 2019 to 2023*. Accessed: Nov. 8, 2023. [Online]. Available: https://www.statista.com/statistics/1061017/digital-video-viewers-number-worldwide/

[2] Research and Markets. *Online Video Platform Market: Global Industry Trends, Share, Size, Growth, Opportunity and Forecast 2023–2028*. Accessed: Nov. 8, 2023. [Online]. Available: https://www.researchandmarkets.com/reports/5769108/online-video-platform-market-global-industry?gclid=Cj0KCQjwsIejBhDOARIsANYqkD241cCkeQyx72Jk5_RVSJvRrXHfHQlAjWE5HOYrzaQ89EsGkZPJRAkaAv4FEALw_wcB

[3] J. Xu, J. Klein, J. Jochims, N. Weissner, and R. Kays, "A reliable and unobtrusive approach to display area detection for imperceptible display camera communication," *J. Vis. Commun. Image Represent.*, vol. 85, May 2022, Art. no. 103510.

[4] C. Chen, W. Huang, L. Zhang, and W. H. Mow, "Robust and unobtrusive display-to-camera communications via blue channel embedding," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 156–169, Jan. 2019.

[5] 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. 13th Annu. Int. Conf. Mobile Syst., Appl., Services*, May 2015, pp. 181–195.

[6] T. Nguyen, M. D. Thieu, and Y. M. Jang, "2D-OFDM for optical camera communication: Principle and implementation," *IEEE Access*, vol. 7, pp. 29405–29424, 2019.
[7] N. T. Le, M. A. Hossain, and Y. M. Jang, "A survey of design and implementation for optical camera communication," *Signal Process., Image Commun.*, vol. 53, pp. 95–109, Apr. 2017.
[8] H. Han, K. Xie, T. Wang, X. Zhu, Y. Zhao, and F. Xu, "RescQR: Enabling reliable data recovery in screen-camera communication system," *IEEE Trans. Mobile Comput.*, early access, May 17, 2023.
[9] V. N. Yokar, H. Le-Minh, Z. Ghassemlooy, and W. L. Woo, "Performance evaluation technique for screen-to-camera-based optical camera communications," *IET Optoelectron.*, vol. 17, no. 4, pp. 184–193, Aug. 2023.
[10] M. T. Kim and B. W. Kim, "DeepCCB-OCC: Deep learning-driven complementary color barcode-based optical camera communications," *Appl. Sci.*, vol. 12, no. 21, p. 11239, Nov. 2022.
[11] S.-Y. Jung, J.-H. Lee, W. Nam, and B. W. Kim, "Complementary color barcode-based optical camera communications," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–8, Feb. 2020.
[12] M. Dalal and M. Juneja, "Video steganography techniques in spatial domain-a survey," in *Proc. Int. Conf. Comput. Commun. Syst.* Cham, Switzerland: Springer, 2018, pp. 705–711.
[13] H. Fang, D. Chen, F. Wang, Z. Ma, H. Liu, W. Zhou, W. Zhang, and N. Yu, "TERA: Screen-to-camera image code with transparency, efficiency, robustness and adaptability," *IEEE Trans. Multimedia*, vol. 24, pp. 955–967, 2022.
[14] X. Bao, J. Pan, Z. Cai, J. Li, X. Huang, R. Chen, and J. Fang, "Real-time display camera communication system based on LED displays and smartphones," *Opt. Exp.*, vol. 29, no. 15, p. 23558, 2021.
[15] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2114–2123.
[16] L. D. Tamang and B. W. Kim, "Deep D2C-net: Deep learning-based display-to-camera communications," *Opt. Exp.*, vol. 29, no. 8, pp. 11494–11511, 2021.
[17] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–672.
[18] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," *Proc. SPIE*, vol. 11433, pp. 991–1005, Jan. 2020.
[19] J. Jia, Z. Gao, K. Chen, M. Hu, X. Min, G. Zhai, and X. Yang, "RIHOOP: Robust invisible hyperlinks in offline and online photographs," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 7094–7106, Jul. 2022.
[20] K. Alex Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "SteganoGAN: High capacity image steganography with GANs," 2019, *arXiv:1901.03892*.
[21] L. D. Tamang and B. W. Kim, "Spectral domain-based data-embedding mechanisms for display-to-camera communication," *Electronics*, vol. 10, no. 4, p. 468, Feb. 2021.
[22] B. W. Kim, H.-C. Kim, and S.-Y. Jung, "Display field communication: Fundamental design and performance analysis," *J. Lightw. Technol.*, vol. 33, no. 24, pp. 5269–5277, Dec. 15, 2015.
[23] S.-Y. Jung, H.-C. Kim, and B. W. Kim, "Implementation of two-dimensional display field communications for enhancing the achievable data rate in smart-contents transmission," *Displays*, vol. 55, pp. 31–37, Dec. 2018.
[24] Y.-J. Kim, P. Singh, and S.-Y. Jung, "Experimental evaluation of display field communication based on machine learning and modem design," *Appl. Sci.*, vol. 12, no. 23, p. 12226, Nov. 2022.
[25] H. Kato, K. T. Tan, and D. Chai, "Development of a novel finder pattern for effective color 2D-barcode detection," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl.*, Dec. 2008, pp. 1006–1013.
[26] C.-K. Liang, L.-W. Chang, and H. H. Chen, "Analysis and compensation of rolling shutter effect," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1323–1330, Aug. 2008.
[27] Caelan. (2023). *Beach People Sunset*. Accessed: Nov. 8, 2023. [Online]. Available: https://pixabay.com/videos/beach-people-sunset-ocean-sunrise-31633/
[28] G. Verzellesi, D. Saguatti, M. Meneghini, F. Bertazzi, M. Goano, G. Meneghesso, and E. Zanoni, "Efficiency droop in InGaN/GaN blue light-emitting diodes: Physical mechanisms and remedies," *J. Appl. Phys.*, vol. 114, no. 7, Aug. 2013, Art. no. 071101.
[29] Upstreamcity. (2023). *Bridge River Architecture*. Accessed: Nov. 8, 2023. [Online]. Available: https://pixabay.com/videos/bridge-river-architecture-city-22608/
[30] V. Karpovich. (2023). *Women Having Dessert at Birthday Party*. Accessed: Nov. 8, 2023. [Online]. Available: https://www.pexels.com/video/women-having-dessert-at-birthday-party-7583204/

YU-JEONG KIM (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Yeungnam University, Gyeongsan, Gyeongsangbuk, South Korea, in 2020 and 2022, respectively, where she is currently pursuing the Ph.D. degree with the Ubiquitous Communications Laboratory, Department of Electronic Engineering.

Her current research interests include artificial intelligence and optical wireless communications.

Ms. Kim is a member of the Korean Institute of Communications and Information Sciences (KICS) and the Institute of Electronics and Information Engineers (IEIE).

PANKAJ SINGH (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from Gujarat University, Ahmedabad, Gujarat, India, in 2009, and the joint M.S. and Ph.D. degree in electronic engineering from Yeungnam University, Gyeongsan, Gyeongsangbuk, South Korea, in 2019.

From March 2019 to August 2019, he was a Postdoctoral Researcher with the Ubiquitous Communications Laboratory, Department of Electronic Engineering, Yeungnam University, where he is currently an Assistant Professor. His research interests include electromagnetic nanonetworks, molecular communications, and optical camera communications.

Prof. Singh is a member of IEIE. He was a recipient of the Outstanding Graduate Student Award in the Daegu and Gyeongbuk Province from the Institute of Electronics and Information Engineers (IEIE), South Korea, in 2018.

SUNG-YOON JUNG (Senior Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2002 and 2006, respectively.

From 2006 to 2009, he was a Senior Engineer with the Telecommunication Research and Development Center, Samsung Electronics Company Ltd. Since 2009, he has been with Yeungnam University, where he is currently a Professor with the Department of Electronic Engineering. He was a Visiting Professor with the Department of Electrical and Computer Engineering and the Department of Epidemiology, University of Florida, in 2015 and 2019, respectively. His research interests include signal processing for wireless communications, optical wireless communications, electromagnetic nanocommunications, molecular communications, and 5G and 6G communications.