DisCo: Display-Camera Communication Using Rolling Shutter Sensors

KENSEI JO
Sony Corporation
MOHIT GUPTA
University of Wisconsin–Madison
and
SHREE K. NAYAR
Columbia University

We present DisCo, a novel display-camera communication system. DisCo enables displays and cameras to communicate with each other while also displaying and capturing images for human consumption. Messages are transmitted by temporally modulating the display brightness at high frequencies so that they are imperceptible to humans. Messages are received by a rolling shutter camera that converts the temporally modulated incident light into a spatial flicker pattern. In the captured image, the flicker pattern is superimposed on the pattern shown on the display. The flicker and the display pattern are separated by capturing two images with different exposures. The proposed system performs robustly in challenging real-world situations such as occlusion, variable display size, defocus blur, perspective distortion, and camera rotation. Unlike several existing visible light communication methods, DisCo works with off-the-shelf image sensors. It is compatible with a variety of sources (including displays, single LEDs), as well as reflective surfaces illuminated with light sources. We have built hardware prototypes that demonstrate DisCo's performance in several scenarios. Because of its robustness, speed, ease of use, and generality, DisCo can be widely deployed in several applications, such as advertising, pairing of displays with cell phones, tagging objects in stores and museums, and indoor navigation.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Image sensor communication, optical wireless communication, user interfaces

Authors' addresses: K. Jo, 4-14-1 Asahi-cho, Atsugi-shi, Kanagawa, 243-0014 Japan; email: kensei.jo@sony.com; M. Gupta, 1210 W. Dayton Street, Room 6395, Madison, WI 53705; email: mohitg@cs.wisc.edu; S. K. Nayar, 450 Mudd Hall 500 West, 120 Street, Computer Science Department, Columbia University, New York, NY 10027; email: nayar@cs.columbia.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2016 ACM 0730-0301/2016/07-ART150 $15.00

DOI: http://dx.doi.org/10.1145/2896818

ACM Reference Format:

Kensei Jo, Mohit Gupta, and Shree K. Nayar. 2016. DisCo: Display-camera communication using rolling shutter sensors. ACM Trans. Graph. 35, 5, Article 150 (July 2016), 13 pages.
DOI: http://dx.doi.org/10.1145/2896818

1. INTRODUCTION

We present DisCo, a novel display-camera communication system that enables displays to send short messages to digital sensors while simultaneously displaying images for human consumption (Figure 1). Existing display-camera communication methods are largely based on spatial-domain steganography, where the information is encoded as an imperceptible spatial signal (e.g., QR code). These methods, while simple to implement, are prone to errors due to common causes of image degradations, such as occlusions, display being outside the sensor's field of view (FOV), defocus blur, and perspective distortion. Due to these limitations, steganography-based techniques have not been widely adopted, especially in uncontrolled settings involving consumer cameras and public displays.

DisCo overcomes these limitations by embedding messages in temporal signals instead of spatial signals. We draw inspiration from the emerging field of visible light communication (VLC), where information is transmitted between a light source (transmitter) and a sensor (receiver) via high-frequency temporally modulated light. Most of these techniques require specialized high-speed cameras or photodiodes as signal receivers [Elgala et al. 2009; Vucic et al. 2010; Sarkera et al. 2009]. Recently, a method was proposed for using low-cost rolling shutter sensors as receivers. However, this method places strong restrictions on the transmitter; only light sources (e.g., LEDs) or surfaces with constant brightness [Danakis et al. 2012] can be used. These systems do not work with displays that need to display arbitrary images. The goal of this work is to design systems that can use a broad range of signal transmitters, especially displays showing arbitrary images, as well as objects that are illuminated with temporally modulated light. The objects can have arbitrary textures. This is shown in Figure 2.

Concept of DisCo: DisCo builds on the method proposed in Danakis et al. [2012] and uses rolling shutter cameras as signal receivers. In rolling shutter sensors, different rows of pixels are exposed in rapid succession, thereby sampling the incident light at different time instants. This converts the temporally modulated light coming from the display into a spatial flicker pattern in the captured image. The flicker encodes the transmitted signal. However, the flicker pattern is superimposed with the (unknown) display pattern.

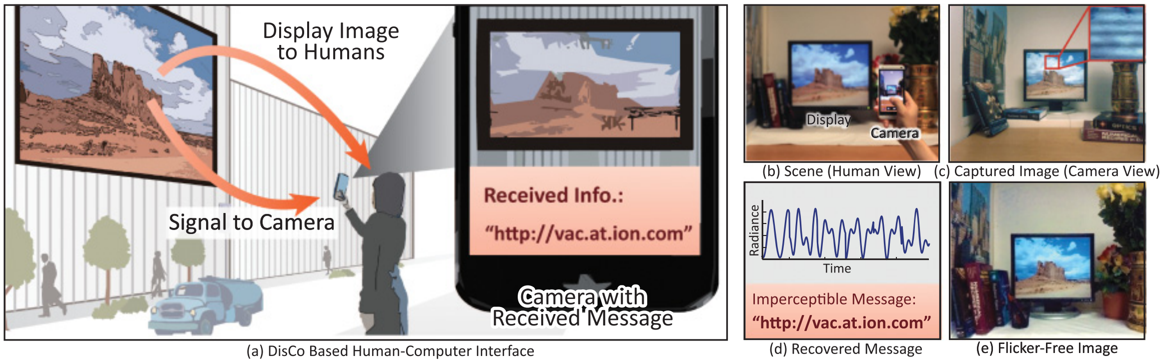


Fig. 1. **Concept of DisCo:** (a) We propose DisCo, a novel display-sensor communication method. It uses fast temporal modulation of displays to transmit messages and rolling shutter sensors to receive them. The messages are imperceptible to humans, allowing displays to serve the dual purposes of displaying images to humans while simultaneously conveying messages to cameras. (b) A scene comprising a display. (c) Image captured by a rolling shutter camera. Due to rolling shutter, temporal modulation of the display is converted into a spatial flicker pattern. The flicker pattern is superimposed on the displayed pattern. By using a sensor that can capture two exposures simultaneously, we can separate the flicker and the display pattern, and thus recover both the message (d), and flicker-free scene image (e) from a single captured image.

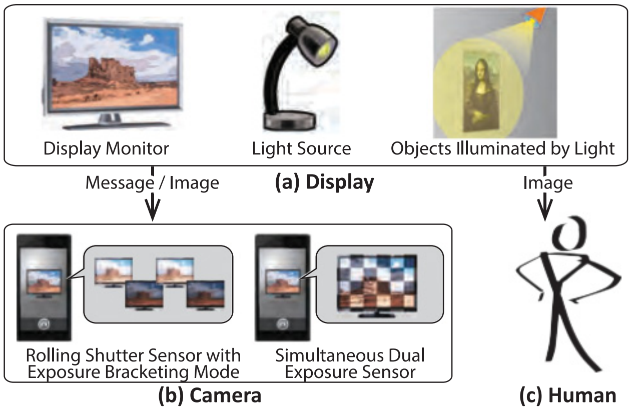


Fig. 2. **Components of DisCo:** DisCo can work with a broad range of devices. (a) For transmitters, DisCo can use display monitors, light sources, or objects illuminated by light as the transmitter ("Display"). (b) The display sends information to the receiver, which can be implemented using rolling shutter cameras. (c) The display also shows images for human consumption while simultaneously communicating with the camera.

This is illustrated in Figure 1. To extract the message, the flicker and the display pattern must be separated. Our key contribution is to show that the two components can be separated by capturing images at two different camera exposures. We also show that the flicker component is invariant to the display pattern and other common imaging degradations (e.g., defocus blur, occlusion, camera rotation, and variable display size). The effect of all of these degradations can be absorbed in the display pattern component. Since the display pattern is separated from the flicker component before signal recovery, the imaging degradations do not adversely affect the communication process.

Hardware implementation and prototypes: DisCo system consists of two main components: the display (transmitter) and the camera (receiver). The (display) transmitter for DisCo can be implemented as an LCD panel with a temporally modulated backlight, or a single LED, or even a nonemitting surface illuminated with a spotlight, as shown in Figure 2. The receiver is a digital camera with a rolling shutter.

We demonstrate two prototype implementations of DisCo. The first prototype uses the exposure bracketing mode available in cameras for acquiring two exposures sequentially. This method, although straightforward to implement on most digital cameras, is prone to errors due to interframe camera motion. Our second prototype is based on simultaneous dual exposure (SDE) sensors. SDE sensors have pixels with two different exposures interlaced with each other [Nayar and Mitsunaga 2000] and can simultaneously capture two exposures in a single image. These sensors are now also commercially available in consumer cameras [OmniVision 2011; Fujifilm 2016] for capturing high dynamic range (HDR) images. This prototype acquires the signal as well as the display pattern from a single image and is thus robust to errors due to motion. This is illustrated in Figure 2. Since it uses easily available hardware for both sending and receiving messages, DisCo can be integrated into existing infrastructure and readily adopted in several applications involving cameras and displays.

Scope and limitations: While designing a communication method, there is a trade-off between the data rate and robustness. On one hand are methods based on high-speed photodiodes [Elgala et al. 2009; Vucic et al. 2010] that can achieve a high data rate, although only in controlled settings. On the other hand, to be applicable as a consumer setting, a communication method must be able to perform reliably in uncontrolled real-world situations while potentially sacrificing the data rate. This is shown in Figure 3. DisCo can work robustly in challenging scenarios, such as when the display is significantly smaller/larger than the sensor FOV, occlusion, perspective distortion, camera rotation, and defocus (Figure 4). In addition to displays, any device emitting temporally modulated light, such as a single or an array of LEDs (e.g., ceiling lights), can be used to convey information. DisCo can also operate in the "spotlight" configuration where a reflective surface illuminated by a light source acts as the transmitter (Figure 2). The surface can have arbitrary shape and texture.[1] The data rate achieved by DisCo is significantly

[1] One exception is if the scene is perfectly black. Since such a scene does not reflect any light, the camera cannot receive the signal.

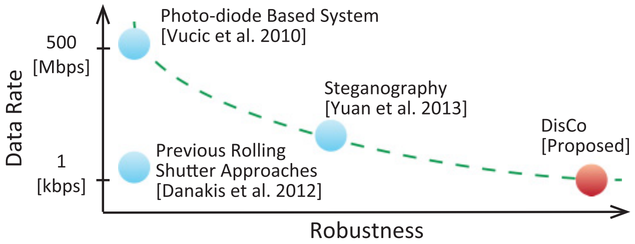


Fig. 3. **Trade-off between data rate and robustness:** Communication techniques face a trade-off between their data rate and robustness. Previous approaches based on photodiodes can achieve high data rate, but in controlled settings such as indoor wireless networks. On the other hand, in consumer settings, a communication method must be able to perform reliably in uncontrolled real-world situations while potentially sacrificing the data rate. The proposed method, DisCo, can work robustly in challenging scenarios while requiring only low-cost consumer devices.

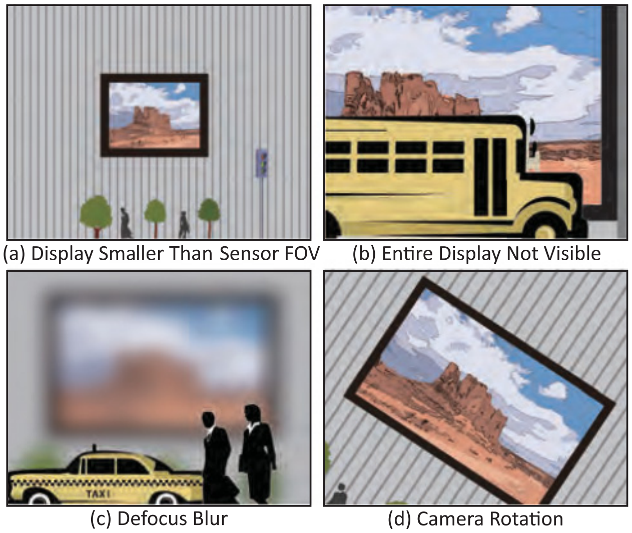

(a) Display Smaller Than Sensor FOV
(b) Entire Display Not Visible
TAXI
(c) Defocus Blur
(d) Camera Rotation

Fig. 4. **Display-Sensor Communication in the Wild:** DisCo performs robustly in challenging real-world situations, such as when the display is significantly smaller than the sensor FOV (a), the display is partially visible to the sensor due to being outside the FOV or due to occlusions (b), there is camera defocus (c), and there is camera rotation (d).

lower than photodiode-based systems but is sufficient to convey short messages such as URLs and pairing keys in a single image, which can enable several user interface applications.

2. RELATED WORK

Spatial-domain steganography: One of the simplest techniques for embedding hidden information in displays is spatial-domain steganography (or watermarking), where a spatial code (e.g., QR code [ISO 2006]) is embedded in the display image [Cheddad et al. 2010; Grundhofer et al. 2007; Yuan et al. 2013; Kamijo et al. 2008; Chan et al. 2010]. The performance of these techniques depends on the distance and inclination of the display with respect to the sensor [Perli et al. 2010]. Moreover, most systems require the entire display to be visible to the sensor. This constraint is a strong limitation, as displays are often only partially visible to the sensor. These techniques are also not robust to common imaging degradations, such as defocus blur (Figure 5(a)).

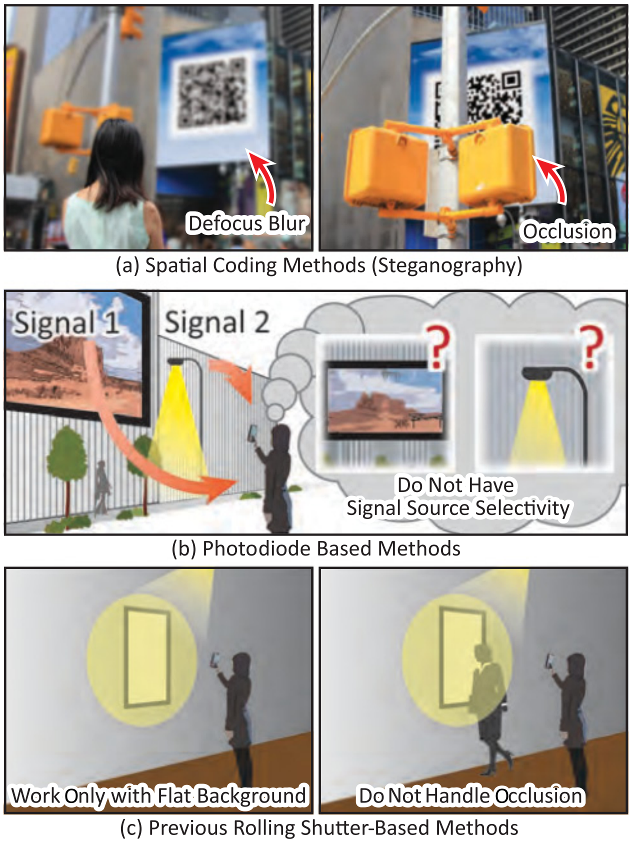
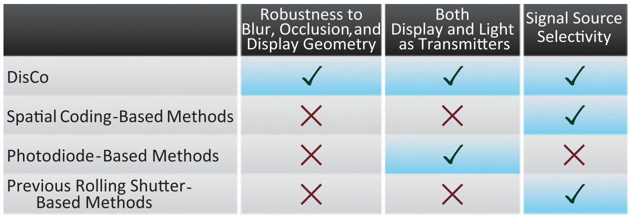

Defocus Blur
Occlusion
(a) Spatial Coding Methods (Steganography)
Signal 1
Signal 2
?
?
Do Not Have
Signal Source Selectivity
(b) Photodiode Based Methods
Work Only with Flat Background
Do Not Handle Occlusion
(c) Previous Rolling Shutter-Based Methods

Fig. 5. **Limitations of previous methods:** (a) Spatial-domain coding methods (e.g., steganography) cannot function reliably in the presence of defocus blur or occlusions. (b) Photodiode-based systems require dedicated high-speed photodiodes for communication. Since they do not capture images, they cannot differentiate between multiple signal sources in a scene. (c) Previous rolling shutter–based methods are limited to work only with flat textureless displays and are not robust to occlusions.

Communication using temporally modulated light: In these systems, the signal transmitter is a light source modulated at high temporal frequencies. These methods require specialized sensors, such as photodiodes [Elgala et al. 2009; Vucic et al. 2010] or high-speed cameras [Yoshimura et al. 2001; Matsushita et al. 2003; Kagawa et al. 2009; Sarkera et al. 2009]. Although photodiodes can receive the transmitted signals, they cannot simultaneously capture images for human consumption. Moreover, since they do not capture images, photodiode-based systems cannot differentiate between multiple signal sources in a scene (Figure 5(b)). High-speed cameras are expensive and cannot capture high spatial resolution images. Hence, these techniques cannot be deployed in consumer devices such as cell-phone cameras. DisCo uses only low-cost off-the-shelf components and can easily be incorporated into existing consumer imaging devices.

Rolling shutter sensors: Rolling shutter image sensors have recently been used for communicating with light sources [Woo et al. 2012; Danakis et al. 2012]. However, these methods have limited applicability, as they cannot work with general-purpose displays



	Robustness to Blur, Occlusion, and Display Geometry	Both Display and Light as Transmitters	Signal Source Selectivity
DisCo	✓	✓	✓
Spatial Coding-Based Methods	✗	✗	✓
Photodiode-Based Methods	✗	✓	✗
Previous Rolling Shutter-Based Methods	✗	✗	✓

Fig. 6. **Comparison with previous methods:** In this table, we compare various communication methods based on their robustness, the ability to use a wide range of light sources as signal transmitters, and the flexibility of selecting the signal source if multiple sources are present in the scene. Most previous communication approaches are optimized for achieving large data rates and are not robust to common imaging degradations. DisCo is designed to be robust, intuitive, and flexible so that it can be widely applicable in user interfaces.

that need to display a large range of images/text. This is because they assume that the transmitter has uniform brightness (spatially). Moreover, they assume that the light source occupies the entire sensor FOV and thus cannot handle occlusion and small light sources (Figure 5(c)). In contrast, DisCo is compatible with a significantly larger class of devices as transmitters, including displays and nonemitting surfaces with arbitrary texture.

Smart displays and light sources: Recently, there has been a lot of research activity toward developing smart displays and light sources that serve an additional purpose. Examples include the Bokode system [Mohan et al. 2009], which uses small physical LED-based tags for communicating with a camera; stereoscopic display for multiple users who can individually observe different stereoscopic images [Kitamura et al. 2001]; and Lumisight Table [Matsushita et al. 2004], which can display different images at different viewing angles without users needing to wear special glasses. DisCo enables displays/light sources to communicate with cameras while allowing them to simultaneously display images to humans.

Radio wave–based communication: Existing radio wave–based communication methods, such as WiFi and Bluetooth, achieve a high data rate over long distances. However, most radio waves do not have directionality and can penetrate walls. To communicate using these modalities, the two devices must be "paired" by manually selecting the device and entering a password. This reduces the overall fluidity of the user experience, which is critical in most consumer applications. Near field communication (NFC) methods perform pairing by bringing the devices close to each other. This physical requirement limits their applicability. With the proposed system, it would be possible to pair devices over large distances with a fast and intuitive "point-and-pair" interface.

Comparison summary: The table in Figure 6 compares DisCo and several existing communication methods, based on their robustness, compatibility with different light sources as signal transmitters, and the flexibility of selecting the signal source if multiple sources are present. Most previous approaches are optimized for achieving large data rates and are not robust to common imaging degradations. Moreover, previous approaches are compatible with only a small set of sources as transmitters (only displays or only uniform background light sources) and receivers (e.g., only high-speed sensors). In comparison, DisCo is designed to be robust and compatible with a wide range of sources and thus is ideally suited for user interface applications.

3. IMAGE FORMATION MODEL

DisCo consists of a spatiotemporally modulated display (transmitter) and a rolling shutter sensor (receiver). The display brightness is temporally modulated with the function $f(t)$, which encodes the signal to be transmitted. We call $f(t)$ the signal function. Conceptually, the display can be thought of as having two layers: a signal layer and a texture layer. The texture layer consists of the image that is displayed to humans. This is illustrated in Figure 7(a). The display could be realized either as an LCD panel with a temporally modulated LED backlight, or as a single LED, or even with a spotlight shining on a reflective surface such as a painting on a wall (Figure 2(a)). In the last case, the illuminated part of the surface is considered the display, and the texture of the surface forms the texture layer.

In the following, we assume that the display completely occupies the sensor FOV so that every sensor pixel receives light only from the display area. This assumption is made only for ease of exposition and is not a requirement of the proposed method.[2] Let $l(x, y, t)$ be the radiance incident at sensor pixel (x, y) at time t.[3] This is illustrated in Figure 7(a). Because the entire display is modulated by a single temporal function $f(t)$, the radiance $l(x, y, t)$ can be factorized into spatial and temporal components:

$$l(x, y, t) = l_{tex}(x, y)\, f(t), \qquad (1)$$

where $l_{tex}(x, y)$ is the amplitude of the temporal radiance profile at pixel (x, y) and is determined by the display's texture layer. Note that the temporal radiance profiles at different sensor pixels differ only in their amplitudes $l_{tex}(x, y)$. This is illustrated in Figure 7(b).

Let $e(x, y, t)$ be the exposure function at pixel (x, y). If pixel (x, y) is *on* (i.e., it captures light) at time t, $e(x, y, t) = 1$; otherwise, if the pixel is *off* (i.e., blocks incident light), $e(x, y, t) = 0$. The measured brightness value $i\,(x, y)$ is

$$i\,(x, y) = k \int_{-\infty}^{\infty} l\,(x, y, t)\, e(x, y, t) dt + n(x, y), \qquad (2)$$

where k is the sensor gain that converts radiance to pixel brightness and $n(x, y)$ is the image noise. We assume that the display has sufficient brightness, and thus the captured images have a sufficient signal-to-noise ratio (SNR). Moreover, as we will discuss shortly, before we perform analysis, we first sum the intensities along each image row, which further increases the SNR. Thus, for the rest of the article, we assume the image noise to be negligible.[4] Since the sensor has a rolling shutter, different rows capture light during different, shifted time intervals. The amount of shift is determined by the row index y and the speed of the rolling shutter. The exposure function $e(x, y, t)$ can be modeled as a time-shifted function $e'(t)$:

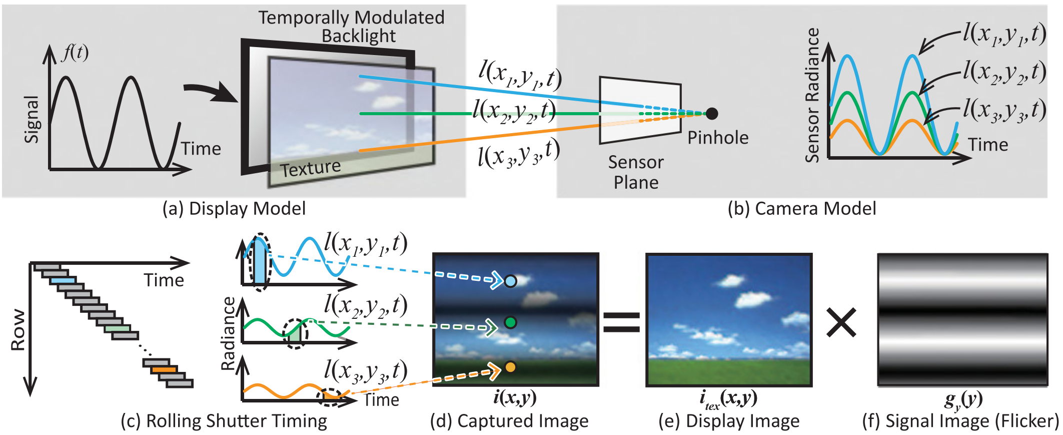
$$e(x, y, t) = e'(t - t_y), \qquad (3)$$

where t_y is the temporal shift for a pixel in row y. The exposure timing of a rolling shutter sensor is illustrated in Figure 7(c). $e'(t)$ can be a rect function, a temporal Gaussian, or even a

[2] In general, a sensor pixel may receive light from outside the display due to the display not completely occupying the sensor's FOV or due to occlusions. It can be shown that the image formation model for the general case has the same form as that of the special case where pixels receive light only from the display. For details, please see the supplementary technical report.

[3] For simplicity, a single color channel is considered. For colored sensors, similar analysis can be done individually for each color channel.

[4] This assumption is not valid when the display image $i_{tex}(x, y)$ is perfectly black. In that case, DisCo cannot function reliably.



(a) Display Model (b) Camera Model

(c) Rolling Shutter Timing (d) Captured Image (e) Display Image (f) Signal Image (Flicker)

Fig. 7. **System overview:** DisCo consists of a spatiotemporally modulated display (transmitter) and a rolling shutter sensor (receiver). (a) The display is modeled as having two layers: the texture layer, which is an image to be viewed by humans, and the backlight layer, which conveys the signal. The intensity of the backlight is temporally modulated with the signal $f(t)$. The temporal frequency of $f(t)$ is significantly higher than what humans can perceive, and thus they only see the image displayed by the texture layer. (b) $l(x, y, t)$ is the temporal radiance profile incident at camera pixel (x, y) at time t. Because the entire display is modulated by the same temporal function, the radiance profiles for different pixels differ only by a multiplicative scale factor, which is determined by the display pattern. (c) Due to the rolling shutter, pixels in different rows sample the temporal radiance profiles at different instants. (d) This creates a spatial flicker in the captured image. We show that the captured image can be factorized as a product of the flicker-free display image (e) and the flicker image (f). The flicker image contains the information embedded in the temporal signal $f(t)$.

high-frequency binary code [Raskar et al. 2006]. We define the shutter function $s(t)$ as $s(t) = e'(-t)$. Then, substituting Equations (1) and (3) in Equation (2), we get

$$\begin{aligned} i\,(x, y) &= k\ \ l_{tex}\,(x, y) \int_{-\infty}^{\infty} s(t_y - t) f\,(t)\,dt \\ &= k\ \ l_{tex}\,(x, y)\ \ g'(t_y), \end{aligned} \qquad (4)$$

where $g'(t_y) = (s * f)\,(t_y)$ is the convolution of the signal and the shutter functions. $g'(t_y)$ is a function of the temporal shift t_y, which in turn depends on the sensor row index y. Typically, $t_y = \frac{y}{r}$, where r rows/second is the speed of the rolling shutter. We rewrite the preceding equation as

$$i\,(x, y) = \underbrace{i_{tex}(x, y)}_{\text{display image}} \times \underbrace{g(y)}_{\text{signal image}}\ , \qquad (5)$$

where $i_{tex}(x, y) = k \times l_{tex}(x, y)$ is called the *display image*, as it is determined by the image being displayed, and $g(y) = g'(t_y) = (s * f)\,(t_y)$ is the signal image that encodes the signal function $f(t)$. The preceding equation states that the texture and the signal layers of the display are observed as two separable (and unknown) components: the display image and the signal image. This is illustrated in Figure 7(e) and (f). The temporal signal $f(t)$ manifests only in the signal image $g(y)$, and the display's texture layer is captured only in the display image $i_{tex}(x, y)$. Equation (5) is the image formation model for DisCo and forms the basis of our method.

Structure of the signal image: The signal image $g(y)$ varies only along the y dimension because different sensor rows sample the signal function $f(t)$ at different instants (Figure 7(d)) and thus have different intensities. However, all pixels in a given row sample $f(t)$ at the same instant and thus have the same intensity. As a result, $g(y)$ has the form of a horizontal flicker pattern, as illustrated in Figure 7(f).

Since the signal image $g(y)$ is 1D, for computational efficiency we perform analysis on horizontal sum images that are 1D signals—that is, $i(y) = \sum_x i(x, y)$ and $i_{tex}(y) = \sum_x i_{tex}(x, y)$. Saturated image pixels are excluded from the summation. Then, Equation (5) can be written as $i(y) = i_{tex}(y) \times g(y)$. For the rest of the article, we use this 1D form of the image formation equation.

Invariance of the signal image to display-camera geometry, partial occlusions, and imaging parameters: The image formation model in Equation (5) is derived without making any assumptions about the display's shape, orientation or location with respect to the sensor, or about imaging parameters such as zoom and defocus. Since the signal component $g(y)$ depends only on the signal function $f(t)$ and the shutter function $s(t)$, any changes in display-sensor geometry or imaging parameters (zoom and focus) manifest only in the display image $i_{tex}(x, y)$. Specifically, the display's orientation and location determine the shape of display's projection in the captured image, sensor zoom influences the size of the display's projection, and camera focus determines the amount of blur in the display image.

If the display is partially occluded so that it is visible to a (nonempty) subset of pixels in each sensor row, because the captured image is summed horizontally, the signal image $g(y)$ is still sampled at every row location. If $\alpha_y > 0$ is the fraction of pixels in sensor row y that see the display, the amplitude of the signal image is scaled by α_y. Under mild assumptions, α_y can be assumed to be locally constant and absorbed in the display image (see the Appendix for details).

As a result, the signal image is always a horizontal flicker pattern. Its functional form and structure are invariant to the display-camera geometry, partial occlusions, and camera parameters. A few examples are illustrated in Figure 8. In the shown examples, $f(t)$ is a 500Hz sinusoidal signal, and the shutter $s(t)$ is a rect function of 0.5ms width such that $s(t) = 1$ when $0 \leq t \leq 0.5$ms, and otherwise $s(t) = 0$. This results in a sinusoidal flicker pattern. Notice that the period of the flicker, h_{sine}, is independent of camera-display geometry or camera zoom. Even if only a small fraction of the display is visible to the camera due to large zoom (Figure 8(c)), the

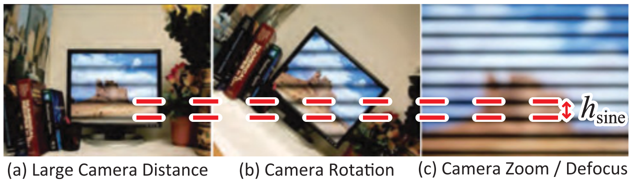


Fig. 8. **Invariance of the signal (flicker) image to imaging geometry:** The flicker pattern in the signal image is invariant to imaging geometry parameters, such as display-camera distance (a), camera rotation (b), and camera zoom and defocus blur (c). In these examples, the display back light was temporally modulated with a 500Hz sinusoid. This results in a spatial sinusoidal flicker. Notice that the period of the flicker, h_{sine}, is the same in all cases.

flicker image retains the same structure and captures the information contained in the signal function.

4. SIGNAL RECOVERY BY CAPTURING TWO DIFFERENT EXPOSURES

To decode the information in the signal image $g(y)$, we need to separate it from the display image $i_{tex}(y)$. Since both signal and display components are unknown, in general, they cannot be separated from a single captured image. The key idea is that if we capture two images $i_1(y)$ and $i_2(y)$ with two different shutter functions $s_1(t)$ and $s_2(t)$, we can get two different equations, which enable performing the separation. The two images can be captured sequentially using the exposure bracketing mode available in most digital cameras. This approach, while suitable for static scenes and cameras, is prone to errors if there is scene/camera motion. As we will describe later in Section 5, to deal with motion, we propose using a camera that captures two images with different exposure functions simultaneously in a single shot.

The two images are given as

$$i_1(y) = i_{tex}(y) \times (s_1 * f)(t_y), \tag{6}$$

$$i_2(y) = i_{tex}(y) \times (s_2 * f)(t_y). \tag{7}$$

This is a system of two equations in two unknowns: signal $f(t)$ and the flicker-free display image $i_{tex}(y)$. Since the shutter functions $s_1(t)$ and $s_2(t)$ are known, these two equations can be solved simultaneously to recover both $f(t)$ and the flicker-free image $i_{tex}(x, y)$. In the following, we provide details of the signal recovery algorithm.

4.1 Signal Model and Recovery Algorithm

We consider the signal $f(t)$ to be a sum of sinusoids of different frequencies (the set of frequencies is typically a small, discrete set). This signal encoding scheme is called orthogonal frequency division multiplexing (OFDM) [Nee and Prasad 2000] and is one of the most popular schemes in communication literature.[5] For each frequency, information is embedded in the phase of the sinusoids. This method of embedding information is called *phase-shift keying* (PSK). For instance, in binary PSK, binary symbols of 0 and 1 are embedded by using sinusoids of phase 0 and π, respectively. Bits (sinusoids with different phases) are transmitted sequentially in time. An example for a single frequency is illustrated in Figure 9.

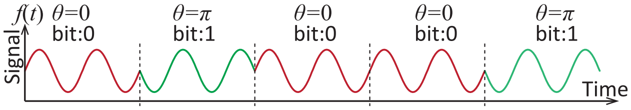


Fig. 9. **Signal coding method:** We use the PSK signal coding technique, where information is embedded in the phase of sinusoidal signals. For example, in binary PSK, the phase θ of sinusoids takes binary values (0 and π), thus encoding binary bits. Bits are transmitted sequentially in time.

Since we use sinusoidal signals, for computational efficiency, we perform computations in the Fourier domain. Equations (6) and (7) can be written in the Fourier domain as

$$I_1(\omega) = I_{tex}(\omega) * (S_1(\omega)\, F(\omega)), \tag{8}$$

$$I_2(\omega) = I_{tex}(\omega) * (S_2(\omega)\, F(\omega)), \tag{9}$$

where ω is the spatial frequency. The functions denoted by uppercase letters are the Fourier transforms of the functions denoted by the corresponding lowercase letters. These two equations can be combined as follows:

$$I_1(\omega) * (S_2(\omega)\, F(\omega)) - I_2(\omega) * (S_1(\omega)\, F(\omega)) = 0\,. \tag{10}$$

The temporal signal $f(t)$ consists of a small, discrete set of temporal frequencies $\Omega = [\omega_1, \ldots, \omega_M]$. We need to solve Equation (10) only for the frequency set Ω. Let $\vec{I_1}$ be the vector of values $[I_1(\omega_1), \ldots, I_1(\omega_M)]$. The vectors $\vec{I_2}$, $\vec{S_1}$, $\vec{S_2}$, and $\vec{F}$ are defined similarly. By observing that convolution can be expressed as multiplication by a Toeplitz matrix and element-wise multiplication as multiplication by a diagonal matrix, Equation (10) can be compactly represented in matrix form as

$$(\mathbf{I}_1\mathbf{S}_2 - \mathbf{I}_2\mathbf{S}_1)\,\vec{F} \;=\; 0, \tag{11}$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are Toeplitz matrices defined by vectors $\vec{I_1}$ and $\vec{I_2}$, respectively. $\mathbf{S}_1$ and $\mathbf{S}_2$ are diagonal matrices defined by vectors $\vec{S_1}$ and $\vec{S_2}$, respectively.

The matrices $\mathbf{I}_1$ and $\mathbf{I}_2$ are defined by captured image intensities, and $\mathbf{S}_1$ and $\mathbf{S}_2$ are defined in terms of the known shutter functions. The goal is to recover the unknown vector $\vec{F}$. The preceding equation can be solved as a linear system of the form $\mathbf{AX} = 0$. To avoid the degenerate solution ($\vec{F} = 0$) and ambiguity (if $\vec{F}$ is a solution, then $s\vec{F}$ is also a solution for any complex number s), we impose the constraint that $F(0) = 1.0$—that is, the DC level of the signal $f(t)$ is 1.0.

Recall that the signal comprises multiple bits that are transmitted sequentially and are thus captured at different spatial locations in the signal image. We recover each bit individually by applying the signal recovery algorithm to a small interval of the captured image at a time. The interval size h_{bit} is the number of image rows required to encode a single bit. h_{bit} is determined by the signal frequency; the higher the frequency of $g(y)$ (due to $f(t)$ having high temporal frequency), the smaller the interval size. Thus, we divide the captured images $i_1(y)$ and $i_2(y)$ into small 1D intervals and recover $\vec{F}$ by computing Equation (11) on each interval individually. Since computations are done locally, $I_1(\omega)$ and $I_2(\omega)$ are the short-time Fourier transforms (STFTs) of $i_1(y)$ and $i_2(y)$. Once $\vec{F}$ is computed, we recover the signal $f(t)$ and the embedded information by

[5] In general, the proposed system can work with any of the several signal encoding schemes proposed in the communications literature.

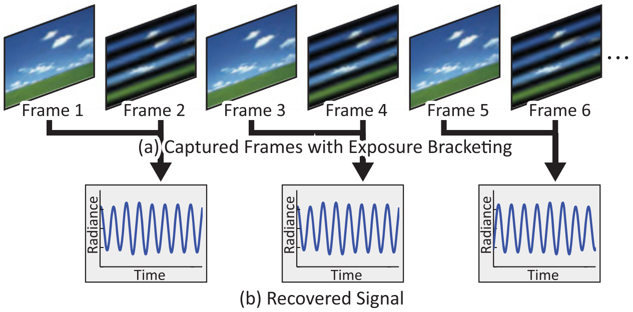


Fig. 10. **Input and output of DisCo with exposure bracketing mode:** (a) DisCo can be implemented by using the exposure bracketing mode available in most existing digital cameras. Two images with different exposures (one long and one short) are captured sequentially. The long exposure is chosen so that the captured image is nearly flicker free (Frames 1, 3, 5, . . .). (b) The two images are then divided to recover the signal image, from which the temporal signal is estimated.

applying inverse Fourier transform. The display image $i_{tex}(x, y)$ is then computed by using Equation (5): $i_{tex}(x, y) = \frac{i(x,y)}{g(y)} = \frac{i(x,y)}{(s*f)(t_y)}$.[6]

Simulations: We evaluated the performance of the signal recovery algorithm using several simulations. The simulations illustrate that brighter and larger display images result in high SNR communication. In addition, although higher signal frequencies may achieve a higher data rate, they result in lower SNR. This trade-off must be considered while designing practical systems. For details, please see the supplementary technical report.

4.2 Capturing Two Exposures With Exposure Bracketing

Most existing digital cameras have an exposure bracketing mode for capturing HDR images, where multiple images with different exposures are captured sequentially. We use the exposure bracketing functionality for capturing the two different exposures required for DisCo. However, because the two images are taken sequentially, the second image samples the emitted temporal signal at a different time instant than the first image and thus captures a different temporal signal $f'(t)$. The two images are given as

$$i_1(y) = i_{tex}(y) \times (s_1 * f)(t_y), \tag{12}$$

$$i_2(y) = i_{tex}(y) \times (s_2 * f')(t_y). \tag{13}$$

Since our decoding algorithm assumes that both the images observe the same signal $f(t)$, it cannot recover the signal. This problem is solved by capturing two images, i_{short} and i_{long}, with alternating short and long exposures, s_{short} and s_{long}, respectively, as shown in Figure 10. If s_{long} is chosen so that it is significantly longer than the period of the temporal signal, the signal image $g_{long}(y) = (s_{long} * f)$ is approximately constant, irrespective of the time instance when the signal is sampled. Thus,

$$(s_{long} * f)(t_y) \approx (s_{long} * f')(t_y) \approx K, \tag{14}$$

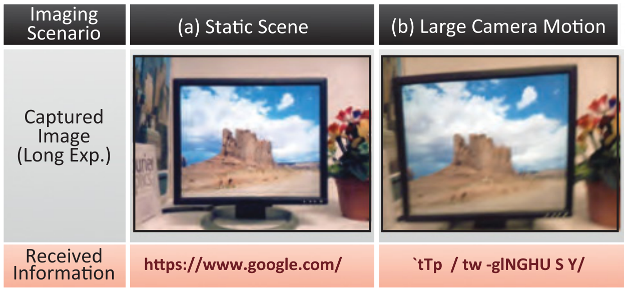


Fig. 11. **Experimental demonstration of DisCo with exposure bracketing:** The two exposures are 0.25ms and 16ms. The message *https://www.google.com/* is embedded in the phases of sinusoids of frequencies of 1 and 2 kHz. (a) If the scene and camera are static (or if the motion is small), the signal is recovered accurately. (b) However, if there is strong camera motion, the images cannot be registered reliably, resulting in incorrect signal recovery.

where K is a constant. By using the preceding approximation, the two images i_{short} and i_{long} can be expressed as

$$i_{short}(y) = i_{tex}(y) \times (s_{short} * f)(t_y), \tag{15}$$

$$\begin{aligned} i_{long}(y) &= i_{tex}(y) \times (s_{long} * f')(t_y), \\ &\approx i_{tex}(y) \times (s_{long} * f)(t_y). \end{aligned} \tag{16}$$

Equations (15) and (16) are the same as Equations (6) and (7). Thus, the signal $f(t)$ can be estimated using the same algorithm given in Section 4.1. Note that the data transmit rate is halved since two images are captured sequentially instead of simultaneously.[7]

Scene and camera motion: The implementation using exposure bracketing assumes that both the scene and the camera are static while the two images are captured. If there is scene/camera motion during capture, the images need to be aligned by computing relative motion between them. Unfortunately, if the interframe motion is large, image alignment techniques often produce inaccurate results. This can result in erroneous signal recovery. Figure 11 shows an example using a conventional camera with exposure bracketing mode. The two exposures are 0.25ms and 16ms. The message *https://www.google.com/* is embedded in the phases of sinusoids of frequencies of 1 and 2 kHz. If the scene and camera are static, the exposure bracketing–based implementation recovers the signal accurately. However, if there is strong camera motion (e.g., due to hand shake), the images cannot be registered reliably, resulting in incorrect signal recovery.

5. CAPTURING TWO EXPOSURES WITH AN SDE SENSOR

To avoid errors in the recovered signal due to motion, the two images with different exposures must be captured simultaneously. One way to achieve this is by using two synchronized cameras that are co-located using additional optics. Although theoretically feasible, this is not a practically viable option, especially in consumer settings.

We propose capturing two different exposures in a single image by using an SDE sensor. An SDE sensor has an array of pixels with two different exposures interlaced with each other. Such sensors

[6] If one of the shutter functions is significantly longer than the period of the signal $f(t)$, the corresponding $g(y)$ will be approximately constant. In that case, the corresponding captured image $i(x, y)$ is nearly flicker free and can directly be used as the display image.

[7] Because $i_{long}(x, y)$ can be approximated as the texture image, it is also possible to estimate flicker component by calculating image ratio $i_{ratio}(x, y) = \frac{i_{short}(x,y)}{i_{long}(x,y)} \approx \frac{g_{short}(y)}{K}$.

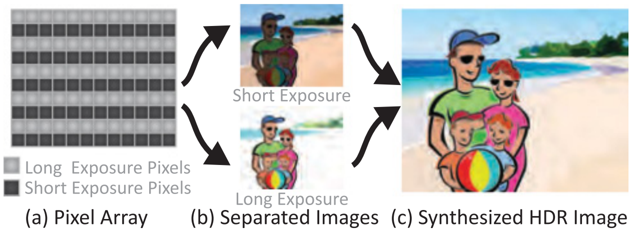


Fig. 12. **SDE sensor:** (a) SDE sensors have an array of pixels with two different exposures. (b) This allows them to capture images with different exposures in a single shot and then synthesize an HDR image (c). Such sensors are now available in consumer cameras due to their ability to capture HDR images.

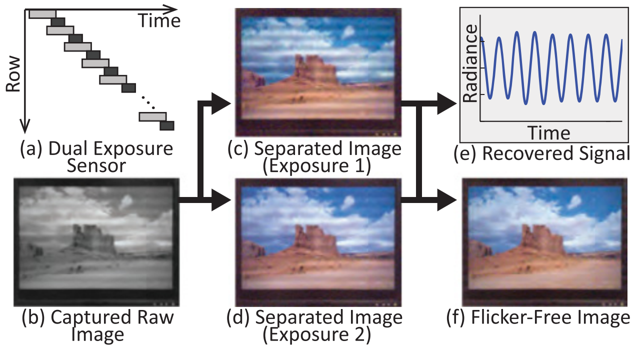


Fig. 13. **Input and output of DisCo with SDE:** (a) Exposure timing of an SDE rolling shutter sensor with two different exposures. (b) An example captured image (input) with a rolling shutter SDE sensor. The display was modulated with a 500Hz temporal sinusoid. (c, d) Images of different exposures are extracted from (b), from which the signal (e) and the flicker-free image (f) are recovered (output).

are now commercially available [Fujifilm 2016; OmniVision 2011] and are expected to be deployed in future consumer cameras given their ability to capture HDR images in a single shot. An example SDE sensor with two different exposures (long and short) is shown in Figure 12. We use an SDE sensor for capturing two subimages with different exposures in a single image. Figure 13 shows an SDE image captured with our prototype. Notice the flicker in the two extracted subimages. Using the two subimages as input, the recovery algorithm estimates the transmitted signal and the flicker-free display image.

5.1 Implementation Details

Our hardware prototype for DisCo consists of a temporally modulated LCD display and an SDE sensor with a rolling shutter (Figure 14). The temporally modulated display was implemented by replacing the backlight of a Dell LCD monitor with an LED array. The LED array is driven with an RECOM RCD-24-1.2 LED driver. We have also developed a prototype where the signal source is a single LED, as well as a prototype in the spotlight configuration where the sensor receives light after reflection from a scene. The SDE rolling shutter camera is based on a Sony IMX135 sensor that is used in smartphones for capturing HDR images. The camera frame rate is 30fps. The camera acquires a single image that contains two subimages of different exposures. The two subimages are separated in a postprocessing step from the raw captured image. The signal frequencies used for the OFDM method range from 1 to 4 kHz. We used quadrature PSK, where each sinusoid can have four possible phases and hence carries 2 bits of information. The camera exposures are selected to capture both signal and scene texture. Specifically, the shorter exposure is chosen to be half of the largest sinusoid period. For example, the exposures of the scenarios that use frequencies of [1, 2] KHz are 0.25ms and 16ms. More hardware details and the result for these prototypes are shown in the supplementary technical report and video.

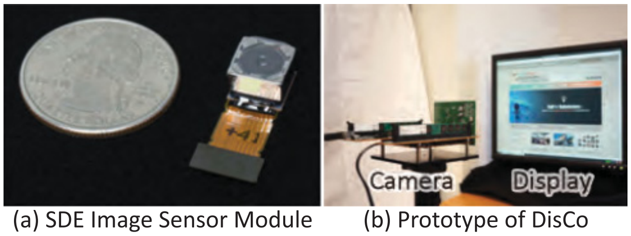


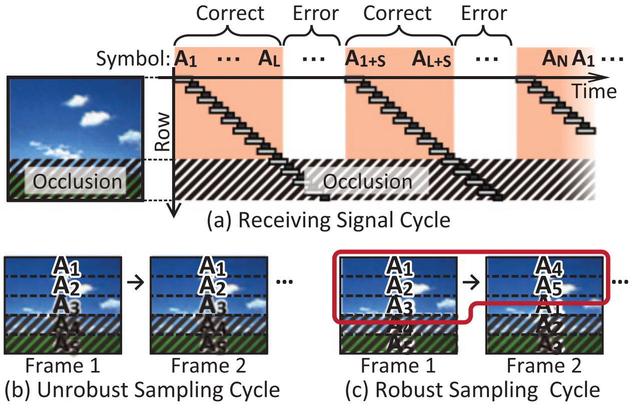
Fig. 14. **Hardware prototype:** (a) The SDE image sensor module. The sensor can capture two different exposures in a single image. (b) A prototype of DisCo with the display and the camera. We have also built a prototype with a single LED as source and a prototype in the spotlight configuration for non–line-of-sight communication (Figure 4(f)). Please see the supplementary video for results.

Temporal synchronization: If the sensor and the display are not temporally synchronized, the start of the transmitted signal cannot be localized in the signal image and the signal cannot be decoded. To handle lack of synchronization, we use two well-known techniques in communications literature. First, a *pilot symbol* is embedded in the signal to determine the beginning of the signal. In our implementation, the pilot symbol is a sinusoid of a frequency that is not used to encode the main signal, so it is readily detected. Second, *guard interval*-based synchronization [van De Beek et al. 1997] is used to determine the start of every symbol (bit). In this scheme, the end of each symbol is copied to its beginning. Then, by self-correlating the signal with itself, the beginning location of every symbol is computed.

Error detection: There are several sources of errors in the signal recovery process (Equation (11)), namely sensor saturation, low display brightness, small display area, and sensor noise. Moreover, although the recovery algorithm is robust to partial occlusions, severe occlusions where none of the pixels in a sensor row sees the display can lead to errors. Finally, if the display occupies only a small area in the captured image, the signal image has low amplitude and the recovered signal has low SNR. In all of these scenarios, the recovered signal may have errors, which must be detected. Let the recovered solution for a region of the captured image be $\vec{F}$. We detect errors by computing the left-hand side of Equation (11), $((\mathbf{I}_1\mathbf{S}_2 - \mathbf{I}_2\mathbf{S}_1)\vec{F})$. If the value is greater than a prescribed threshold, we declare the recovered signal to be erroneous. To compare the raw error rates, we did not use error detection and correction schemes (e.g., CRC) in our implementation. However, in a practical implementation, these schemes can be used to further increase robustness.

5.2 Achieving Robustness to Occlusions

DisCo handles occlusions by creating redundancy in the transmitted signals and optimizing the signal length. The display transmits the signal, $f(t)$, repeatedly, and the sensor captures a sequence of frames (assuming small interframe motion). However, since the



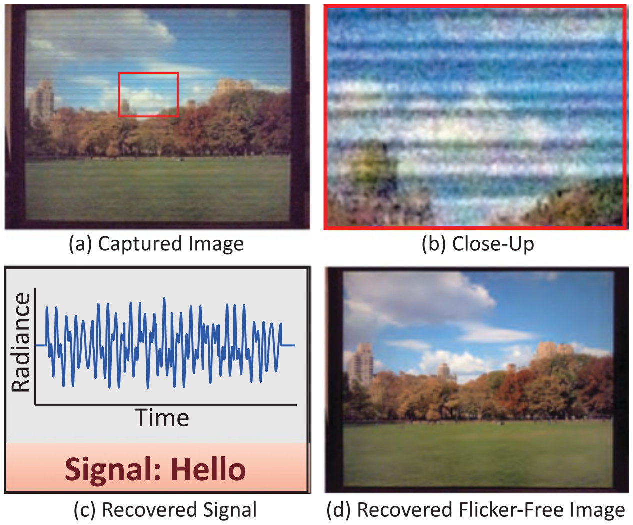
(a) Receiving Signal Cycle

(b) Unrobust Sampling Cycle

(c) Robust Sampling Cycle

Fig. 15. **Dealing with occlusions by optimizing the signal length:** (a) If the display is occluded, then the sensor receives signal only from the visible portion of the display. As a result, the occluded part of the signal cannot be recovered from a single image. (b) If the signal is such that every symbol is transmitted from the same fixed location on the display repeatedly, then even small occlusions will prevent communication. (c) On the other hand, by optimizing the signal length so that a display location transmits different symbols in successive frames, multiple frames can be combined to receive a message even in the presence of strong occlusions.

errors are location specific (due to occlusions or low texture brightness), subsequent signals will have the same pattern of readable and unreadable data. Figure 15(a) illustrates this scenario. If the signal is designed such that every symbol is transmitted from the same fixed location on the display repeatedly, then small occlusions will prevent communication. An example is shown in Figure 15(b). In this case, the symbols $\mathbf{A_4}$, $\mathbf{A_5}$ will never be received by the sensor. We mitigate this problem by designing a robust signal encoding system that delivers a message by varying the spatial location of symbols over multiple frames, as shown in Figure 15(c). Specifically, let $\mathbf{A_1}$ to $\mathbf{A_N}$ be the transmitted symbols, let $\mathbf{N}$ be the message length (number of symbols), and let $\mathbf{S}$ be the number of symbols that can be transmitted between camera frames (1/30 seconds). To ensure that a spatial location on the screen transmits different symbols across different frames, we choose $\mathbf{N}$ so that it is co-prime with $\mathbf{S}$. In this case, even if single symbol is visible from the occluded display, the entire message will be received by the sensor over multiple frames. If the original message length is different from the chosen $\mathbf{N}$, we add dummy symbols to the message to ensure that the total number of transmitted symbols becomes the chosen $\mathbf{N}$. This procedure reduces the data rate of a single frame. However, it increase robustness to occlusion that results in higher effective data rate. Ultimately, the data rate depends on $\mathbf{N}$, $\mathbf{S}$, and the amount of occlusions. Given an $\mathbf{S}$, we find $\mathbf{N}$ that maximizes the average data rate for all possible amounts of occlusions by using a search-based procedure (details of the procedure are given in the supplementary technical report). For instance, in our implementation where $\mathbf{S} = 20$, $\mathbf{N} = 29, 49, 69, 89$, and so on, achieves the higher data rate given our system parameters. By using this approach, DisCo handles challenging situations (e.g., strong occlusions or small display areas) using multiple images. Even for such situations, DisCo degrades the data rate gracefully instead of entirely preventing communication. Examples demonstrating the functionality of DisCo in various levels of occlusion are shown in Section 5.3.



(a) Captured Image

(b) Close-Up

(c) Recovered Signal

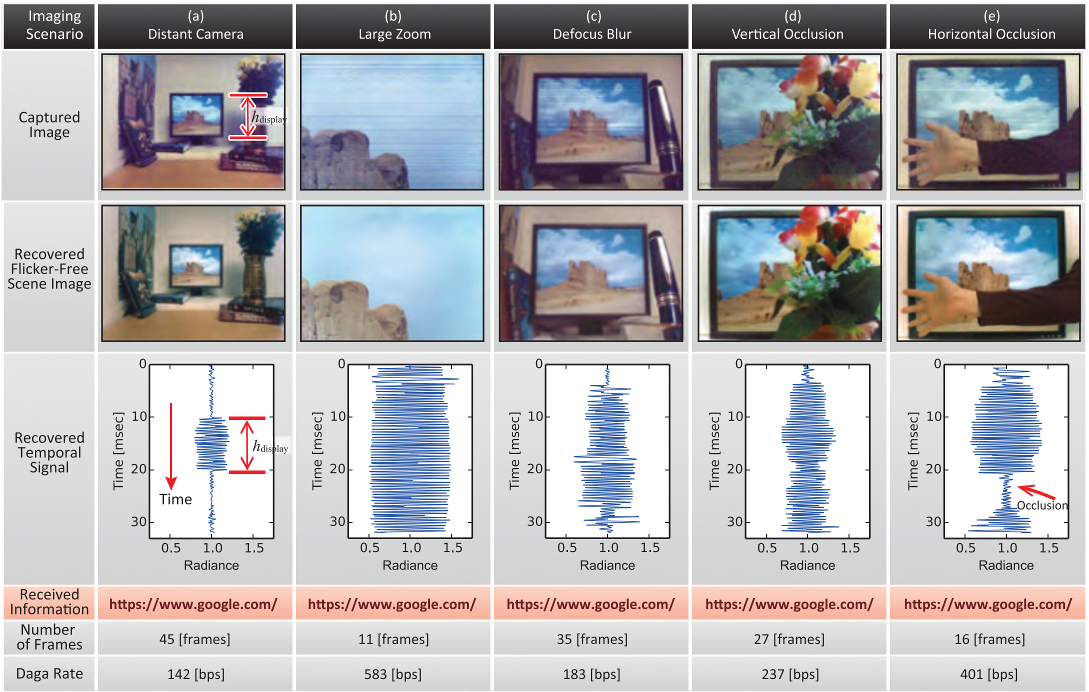
(d) Recovered Flicker-Free Image

Fig. 16. **Experimental demonstration of single-image communication:** (a) An image captured using our prototype. (b) Close-up of (a). Notice the flicker pattern. (c) Recovered temporal signal and the embedded text "Hello." (d) Recovered flicker-free display image.

Imaging Scenario	(a) Static Scene	(b) Large Camera Motion
Captured Image		
Received Information	https://www.google.com/	https://www.google.com/
Number of Frames	27 [frames]	35 [frames]
Data Rate	237 [bps]	183 [bps]

Fig. 17. **Performance of SDE sensor-based DisCo in the presence of camera motion:** The camera motion is the same as in Figure 11. Although exposure bracketing–based implementation can recover the signal reliably only when the camera is static, the SDE-based implementation achieves accurate results even in the presence of large camera motion. Note that more frames are required for signal recovery when there is motion compared to when the scene is static.

Performance of DisCo under motion: As discussed previously, to achieve robustness to occlusions, DisCo may require capturing multiple frames. Thus, we need to consider the effect of possible background texture variation (e.g., due to the screen displaying a video instead of a static background) and motion due to camera shake or scene movement between successive frames. DisCo is robust to variations in background texture. This is because during decoding, the texture component is separated from each captured frame independently. Thus, even large variations in the background texture do not adversely affect the decoding process. On the other hand, DisCo assumes that the relative location of the sensor and display remains constant during capture of multiple frames. However, in practice, if the motion is not large (e.g., relative small motion resulting from camera shake), DisCo can perform reliably due to the redundancy built in the signal encoding (see the supplementary video for results).



Imaging Scenario	(a) Distant Camera	(b) Large Zoom	(c) Defocus Blur	(d) Vertical Occlusion	(e) Horizontal Occlusion
Received Information	https://www.google.com/	https://www.google.com/	https://www.google.com/	https://www.google.com/	https://www.google.com/
Number of Frames	45 [frames]	11 [frames]	35 [frames]	27 [frames]	16 [frames]
Daga Rate	142 [bps]	583 [bps]	183 [bps]	237 [bps]	401 [bps]

Fig. 18. **Experimental demonstration of DisCo in challenging imaging scenarios:** We have extensively tested the proposed system in a variety of real-world imaging situations. (a) Display smaller than the camera's FOV due to large display-camera distance. (b) Display larger than the camera's FOV due to large zoom/small display-camera distance. (c) Display blurred due to camera defocus. (d–e) Occluding objects between the camera and display. In these examples, the signal (URL: https://www.google.com/) was transmitted and received by using exposures (0.25ms and 16ms). The system adapts to challenging conditions by capturing multiple frames. To capture the high-frequency temporal signal, one of the two exposures is short. As a result, the captured image appears noisy. However, since the other exposure is long, the recovered flicker-free image has significantly lower noise. For more results, please see the supplementary video.

Note that compared to static scene, more frames are required for signal recovery when there is motion because the signal size is optimized, assuming the scene is static. Moreover, we assume that α remains fixed between consecutive frames. If the display moves, α changes between frames, which requires capturing more frames.

5.3 Results

Communication using a single captured image: Figure 16 shows results of display-sensor communication using a single captured image with the SDE sensor. The display sends the message *Hello*, and the sensor captures a single SDE image with two interleaved exposures (0.25ms and 16ms). By applying the signal recovery algorithm, the signal and flicker-free scene image are separated.

Signal recovery in the presence of motion: Figure 17 shows the performance of SDE sensor-based DisCo when the camera is moving. The camera motion is the same as in Figure 11. Although exposure bracketing–based implementation can recover the signal reliably only when the camera is static, the SDE-based implementation achieves accurate results even in the presence of large camera motion.

Performance "in the wild": We have evaluated our system in different challenging real-world situations. Figure 18 shows several examples. In each case, the display sent the URL *https://www.google.com*, and the sensor received it without error. A single modulation frequency of 2kHz was used. The frequency of the pilot symbol (used for synchronization) was 1kHz. The number of required frames depends on the display area and the shape of occlusions. In every example, the URL was received within 1.5 seconds (the sensor frame rate is 30fps). For more results, see the supplementary video.

Communicating with a light source and a reflective surface: We have also developed prototypes where a single LED and a nonemitting reflective surface serve as the signal transmitters. Figure 19 shows experimental results for these two cases. When the light source is bright so that the camera observes a high SNR image, it is possible to use multiple frequencies that enable a higher data rate. Figure 19(a) and (b) used four ([1, 2, 3, 4] kHz) and two ([1, 2] kHz) frequencies for embedding information, respectively. We can choose appropriate exposures for the scene. For example, we use shorter exposures (0.065ms and 0.25ms) for Figure 19(a) to avoid saturation, whereas longer exposures (0.25ms and 16ms) are used for Figure 19(b). In the spotlight configuration (Figure 19(b)), an LED lamp illuminating a photograph on the wall was used to tag the photograph with metainformation (the time and location of the photograph). The information was received by the camera viewing the photograph.

Comparison with single exposure–based methods: To demonstrate the advantage of using two different exposures over previous single exposure methods, we compare the communication performance of the single and the dual exposure methods, as shown in Figure 20. The bottom row shows the message received using the single exposure method. We used the same signal encoding and decoding approaches and the same total capture time for both methods,

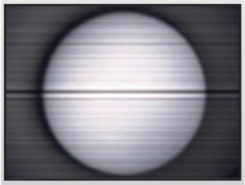
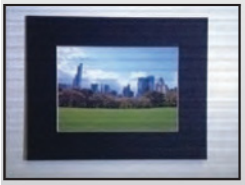
Imaging Scenario	(a) Communicating with Light Source	(b) Non-Line-of-Sight Communication
Captured Image (Close-Up)		
Received Information	https://www.google.com/	Fall 2014, Central Park, New York
Frequencies	1, 2, 3, 4 [kHz]	1, 2 [kHz]
Number of Frames	4 [frames]	7 [frames]
Data Rate	1.6 [kbps]	1.1 [kbps]

Fig. 19. **Communicating with a light source and a reflective surface:** DisCo can use a single LED and a nonemitting reflective surface as the signal transmitters. (a) When directly communicating with a bright light source, because of high SNR, high data rate is achieved. (b) In the spotlight configuration, an LED lamp shining on a photograph is used to tag it with metainformation (the time and location of the photograph). The information was received by the camera viewing the photograph. Since the illuminated surface can have arbitrary shape and texture, this functionality can be used in museums and stores for tagging objects.

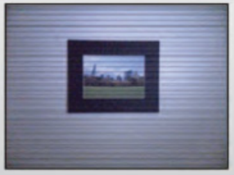
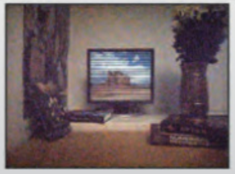
Imaging Scenario	(a) Scene with Mostly Flat Texture	(b) Scene with Complex Background and Texture
Captured Image (Single Exposure)		
SDE	Fall 2014, Central Park, New York	https://www.google.com/
Single Exposure	FQll 2014, Central Park, Ndw Yock	h R ? (cannot receive message)

Fig. 20. **Comparison with the single exposure:** The bottom row shows the intermediate received message of the single exposure when SDE finished the message. Although the single exposure causes much error, it can measure the message from the scene with a large flat area (a). However, in a scene with high texture (b), the single exposure cannot finish communication.

as described earlier in the article. For the single exposure method, the background texture is assumed to be uniform, and the only unknown is the signal component. As a result, the single exposure method performs relatively robustly when the scene has large flat areas, as shown in Figure 20(a). There are only a few errors. However, if the scene has high-frequency components (Figure 20(b)), the single exposure method results in large errors because it interprets the background texture as signal. On the other hand, the dual exposure method can separate the texture and signal.

6. DISCUSSION AND LIMITATIONS

Potential applications: We have proposed DisCo, a novel display-sensor communication technique. It works with commercially available image sensors (both conventional and SDE) and a variety of sources (display, lights, illuminated surfaces), and it performs reliably in difficult real-world scenarios. Since DisCo requires only off-the-shelf components, it can potentially be adopted in various applications, such as advertising billboards communicating metainformation (e.g., URLs), indoor navigation and location-specific services, interactive presentations, and in museums where strategically installed spotlights serve the dual purpose of enhancing the artifacts' appearance while simultaneously communicating information about them. Another application is enabling fast pairing of cell phones with external displays. This can allow users to have a large display as an extension of their small cell-phone screen, and also to share a large display with other users. For a description of potential applications of DisCo, see the supplementary technical report. In the following, we discuss some limitations of DisCo.

Modulation frequencies and signal length: The sensor must know the modulation method, the encoding scheme, and the modulation frequencies of the light sources a priori. This can be achieved by establishing communication standards in display-sensor communication so that a fixed set of modulation frequencies are used. This is similar to radio wave–based communication modalities (e.g., WiFi), where the modulation frequencies are prespecified. It may be possible to relax this restriction and use arbitrary modulation frequencies. For this, the sensors must be able to compute the modulation frequency before performing decoding. Although this functionality will provide more flexibility and will increase the available bandwidth, it may come at the cost of a decreased data rate.

Currently, we also assume that the signal length is known to the receiving sensor. The receiver may be able to estimate the signal length automatically by measuring the time difference of arrival between two consecutive pilot symbols. This is an interesting avenue for future work.

Number of displays: So far, we have assumed that the sensor communicates with a single source (display) at a time. This limits the data transfer rate. It is possible to communicate via multiple sources simultaneously and thus achieve higher data rates by segmenting the captured image into different display regions.

Shutter function and SNR: We have limited ourselves to rect (box) shutter functions due to ease of implementation. However, the algorithm is not limited to rect functions. It is possible to achieve higher SNR by using temporally coded shutter functions [Raskar et al. 2006]. In addition, we used a pair of exposures computed using a search-based procedure, which may not be optimal. Designing theoretically optimal shutter functions requires further analysis of the image formation process and is another promising future work direction.

APPENDIX

A. PARTIAL OCCLUSION

In this section, we derive the image formation model for the DisCo system in the general case, where sensor pixels may see objects other than the display. This may happen due to occlusions or the display not completely occupying the sensor's FOV. Some example scenarios are shown in Figure 4.

In the following, we show that under mild assumptions, the image formation model for the DisCo system in the general case has the same form as the special case where sensor pixels receive light only from the display (the special case is derived in the article). Because of this, the signal recovery algorithm for the special case can be applied to the general case as well. Hence, for the proposed DisCo system, the display (signal transmitter) need not be explicitly segmented in the captured image, and no markers need to be placed on the display.

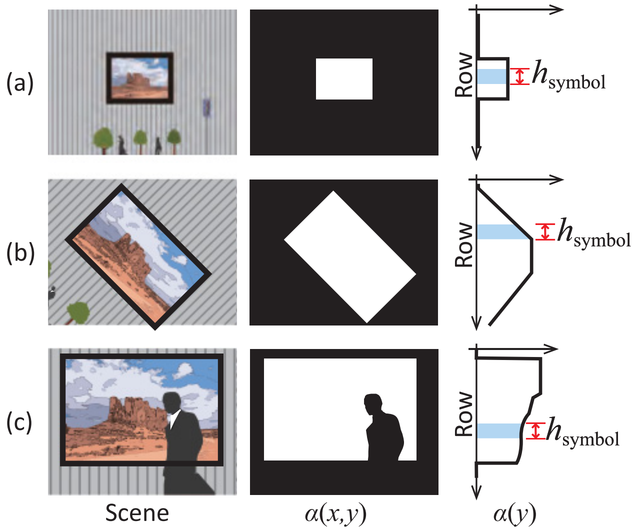


Fig. 21. Illustration of smoothness of the visibility term $\alpha(y)$. (Left column) Real-world situations, such as a display being smaller than the sensor FOV (a), arbitrary display orientation (b), and occlusions (c). (Middle column) The 2D visibility maps $\alpha(x, y)$ may have sharp edges and high spatial frequencies. (Right column) The 1D visibility maps $\alpha(y)$ vary smoothly as a function of y (the row index). If the signal image $g(y)$ has high frequencies, $\alpha(y)$ can be assumed to be locally constant and absorbed in the texture term.

Consider a sensor pixel (x, y) that does not see the display. Such a pixel is called a *nondisplay pixel*. We define the visibility term $\alpha(x, y)$ such that $\alpha(x, y) = 1$ if the pixel (x, y) only sees the display and $\alpha(x, y) = 0$ if it sees only an object other than the display. If a pixel captures a mixture of display and another object (e.g., at display or occlusion boundaries or due to defocus), $0 < \alpha(x, y) < 1$. The visibility maps for a few example scenarios are illustrated in Figure 21. Note that for each sensor pixel (x, y), the visibility term is also an unknown (in addition to the texture and the signal terms).

Consider a nondisplay pixel (x, y) $(\alpha(x, y) = 0)$. For such a pixel, its incident radiance is constant over time. Using the notation in the article, for such pixels we assume $f(t) = 1$, and hence $l(x, y, t) = l_{tex}(x, y)$. Following the derivation in the article, the measured intensity $i(x, y)$ is given as

$$i\,(x, y) = i_{tex}\,(x, y)\;\; E, \tag{17}$$

where $E = \int_{-\infty}^{\infty} s(t)dt$ is the temporal integral of the camera shutter function. For a rect shutter function, E is simply the exposure time. Note that since the camera shutter function is known, E is known a priori.

The combined image formation model for all sensor pixels (display and nondisplay) can thus be written as the following linear combination:

$$\begin{aligned} i\,(x, y) \;&=\; \alpha(x, y)\;[i_{tex}\,(x, y)\;\; g(y)] \\ &\quad + (1 - \alpha(x, y))\;\, [i_{tex}(x, y)\; E] \\ &=\; i_{tex}\,(x, y)\;\, [\alpha(x, y)\;\; (g(y) - E) + E]\,. \end{aligned} \tag{18}$$

Since we consider horizontal sum images, we rewrite the preceding equation in terms of horizontal sums:

$$i\,(y) \;=\; i_{tex}\,(y)\;\, [\alpha(y)\;\; (g(y) - E) + E]\,, \tag{19}$$

where $i\,(y) = \sum_x i(x, y)$, $i_{tex}\,(y) = \sum_x i_{tex}(x, y)$, and $\alpha(y) = \left[\sum_x i_{tex}(x, y) \times \alpha(x, y)\right] / i_{tex}(y)$.

The preceding equation has three unknowns: $i_{tex}(y)$, $\alpha(y)$, and $g(y)$. To reduce the number of unknowns, we make the following observation: even though the visibility maps $\alpha(x, y)$ may have sharp edges, the 1D signal $\alpha(y)$ is relatively smooth. This is illustrated in Figure 21.

Recall from Section 4.1 that the signal comprises multiple symbols (bits), and each bit is recovered individually. The signal recovery algorithm is applied to the image in a patchwise manner, where the patch size h_{symbol} is the number of image rows required to encode a single bit (see Figure 21). Patch size h_{symbol} is determined by the signal frequency; the higher the frequency of $g(y)$, the smaller the patch size. Thus, if the signal image $g(y)$ is a high-frequency pattern (due to $f(t)$ having high temporal frequency), the horizontal sum $\alpha(y)$ can be assumed to be constant within a single patch (a single bit).[8] Note that α may be different for different patches. However, within every patch, α is approximately constant. Thus, Equation (19) can be written as

$$i\,(y) \;=\; i_{tex}\,(y)\;\, [\alpha\;\; (g(y) - E) + E]\,. \tag{20}$$

We use the PSK approach for embedding signal, where the information is embedded in the phase of the sinusoids. Multiplication or addition of a signal by a constant scalar does not change its phase. Thus, the phase of the function $\overline{g}(y) = \alpha \times (g(y) - E) + E$ is the same as the function $g(y)$.[9] If we compute the phase of the function $\overline{g}(y)$, we can recover the embedded information. Hence, we can consider $\overline{g}(y)$ to be the signal image and rewrite the preceding equation as

$$i\,(y) \;=\; i_{tex}\,(y) \times \overline{g}(y)\,. \tag{21}$$

This is the image formation model for the DisCo system in the general case where a camera pixel may see an object other than the display. This equation has two unknowns: the texture image $i_{tex}(y)$ and the signal image $\overline{g}(y)$. The equation has the same form as the special case where a camera pixel sees only the display (Equation (5)). Thus, the signal recovery techniques for the special case can be applied without modification to the general case as well. We demonstrate display-camera communication for several examples of the general case (e.g., occlusions, display smaller than the camera's FOV).

ACKNOWLEDGMENT

This research was conducted in the Computer Vision Laboratory at Columbia University. Kensei Jo is with Sony Corporation and was a visiting researcher at Columbia University. Shree Nayar received a letter of appreciation with a prize from Sony Corporation for his contributions to research and education.

REFERENCES

Li-Wei Chan, Hsiang-Tao Wu, Hui-Shan Kao, Ju-Chun Ko, Home-Ru Lin, Mike Y. Chen, Jane Hsu, and Yi-Ping Hung. 2010. Enabling beyond-surface interactions for interactive surface with an invisible projection.

[8] This assumption is not valid if the display has sharp horizontal boundaries (in the captured image) or if the display is occluded by objects having sharp horizontal edges. These scenarios may result in incorrect signal recovery. We handle these cases by incorporating redundancy in our signal encoding scheme and using error detection and correction techniques during signal recovery.

[9] If an entire row of camera pixels do not see the display, $\alpha(y) = 0$ for that row. In that case, the phase of $\overline{g}(y)$ cannot be recovered. This can be handled by our error detection and correction algorithm.

In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST'10)*. ACM, New York, NY, 263–272. DOI:http://dx.doi.org/10.1145/1866029.1866072

Abbas Cheddad, Joan Condell, Kevin Curran, and Paul McKevitt. 2010. Digital image steganography: Survey and analysis of current methods. *Signal Processing* 90, 3, 727–752. DOI:http://dx.doi.org/10.1016/j.sigpro.2009.08.010

Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. 2012. Using a CMOS camera sensor for visible light communication. In *Proceedings of the IEEE Workshop on Optical Wireless Communications*.

H. Elgala, R. Mesleh, and H. Haas. 2009. Predistortion in optical wireless transmission using OFDM. In *Proceedings of the International Conference on Hybrid Intelligent Systems*.

Fujifilm. 2016. EXR CMOS Technology. Retrieved June 4, 2016, from http://finepix.com/exr_cmos/en/.

A. Grundhofer, M. Seeger, F. Hantsch, and O. Bimber. 2007. Dynamic adaptation of projected imperceptible codes. In *Proceedings of the International Symposium on Mixed and Augmented Reality*.

ISO. 2006. *QR Code 2005 Bar Code Symbology Specification*. ISO/IEC 18004:2006,.

K. Kagawa, J. Ohta, and J. Tanida. 2009. Dynamic reconfiguration of differential pixel output for CMOS imager dedicated to WDM-SDM indoor optical wireless LAN. *IEEE Photonics Technology Letters* 21, 18, 1308–1310.

K. Kamijo, N. Kamijo, and Zhang Gang. 2008. Invisible barcode with optimized error correction. In *Proceedings of the IEEE International Conference on Image Processing*. DOI:http://dx.doi.org/10.1109/ICIP.2008.4712185

Yoshifumi Kitamura, Takashige Konishi, Sumihiko Yamamoto, and Fumio Kishino. 2001. Interactive stereoscopic display for three or more users. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*. ACM, New York, NY, 231–240. DOI:http://dx.doi.org/10.1145/383259.383285

Mitsunori Matsushita, Makoto Iida, Takeshi Ohguro, Yoshinari Shirai, Yasuaki Kakehi, and Takeshi Naemura. 2004. Lumisight table: A face-to-face collaboration support system that optimizes direction of projected information to each stakeholder. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW'04)*. ACM, New York, NY, 274–283. DOI:http://dx.doi.org/10.1145/1031607.1031653

N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, J. Rekimoto, and Y. Yamamoto. 2003. ID CAM: A smart camera for scene capturing and ID recognition. In *Proceedings of the International Symposium on Mixed and Augmented Reality*.

Ankit Mohan, Grace Woo, Shinsaku Hiura, Quinn Smithwick, and Ramesh Raskar. 2009. Bokode: Imperceptible visual tags for camera based interaction from a distance. *ACM Transactions on Graphics* 28, 3, Article No. 98. DOI:http://dx.doi.org/10.1145/1531326.1531404

S. K. Nayar and T. Mitsunaga. 2000. High dynamic range imaging: Spatially varying pixel exposures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Richard van Nee and Ramjee Prasad. 2000. *OFDM for Wireless Multimedia Communications.* Artech House Publishers, Norwood, MA.

OmniVision. 2011. OV4689. Retrieved June 4, 2016, from http://www.ovt.com/products/sensor.php?id=136.

Samuel David Perli, Nabeel Ahmed, and Dina Katabi. 2010. PixNet: Interference-free wireless links using LCD-camera pairs. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom'10)*. 137–148.

Ramesh Raskar, Amit Agrawal, and Jack Tumblin. 2006. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Transactions on Graphics* 25, 3, 795–804.

M. S. Z. Sarkera, I. Takai, M. Andoh, K. Yasutomi, S. Itoh, and S. Kawahito. 2009. A CMOS imager and 2-D light pulse receiver array for spatial optical communication. In *Proceedings of the IEEE Asian Solid-State Circuits Conference*.

Jan-Jaap van De Beek, Magnus Sandell, and Per Ola Börjesson. 1997. ML estimation of time and frequency offset in OFDM systems. *IEEE Transactions on Signal Processing* 45, 7, 1800–1805.

J. Vucic, C. Kottke, S. Nerreter, K.-D. Langer, and J. W. Walewski. 2010. 513 Mbit/s visible light communications link based on DMT-modulation of a white LED. *Journal of Lightwave Technology* 28, 24, 3512–3518.

Grace Woo, Andy Lippman, and Ramesh Raskar. 2012. VRCodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *Proceedings of the International Symposium on Mixed and Augmented Reality*. http://dx.doi.org/10.1109/ISMAR.2012.6402539

S. Yoshimura, T. Sugiyama, K. Yonemoto, and K. Ueda. 2001. A 48 kframe/s CMOS image sensor for real-time 3-D sensing and motion detection. In *Proceedings of the IEEE International Solid-State Circuits Conference*.

Wenjia Yuan, Kristin Dana, Ashwin Ashok, Marco Gruteser, and Narayan Mandayam. 2013. Spatially varying radiometric calibration for camera-display messaging. In *Proceedings of the IEEE Conference on Signal and Information Processing*.

Received February 2015; revised October 2015; accepted February 2016