Imperceptible Color Vibration for Embedding Pixel-by-Pixel Data into LCD Images

Satoshi Abe
Atsuro Arami
Takefumi Hiraki
Shogo Fukushima
Takeshi Naemura
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Japan
{abe, atsuro, hiraki, shogo, naemura}@nae-lab.org

CHI'17 Extended Abstracts, May 6–11, 2017, Denver, CO, USA.
ACM ISBN 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053074

Abstract
In addition to displaying images, graphic display devices can be configured to transmit information to nearby computing devices, to facilitate human interaction with the image. However, some existing methods require the use of visible markers, which impairs the visual experience, while others require specially modified projectors.. To solve these problems, we propose a novel method that employs imperceptible color vibration to embed pixel-by-pixel data into images on an ordinary LCD display; this is achieved by fixing the luminance and vibrating only the chromaticity of the color of each pixel. We show that five different data values can be embedded and detected with adequate responsiveness for human-computer interaction (HCI). It is expected that our method will be applied to wearable tactile displays, robot controls, and other similar technologies.

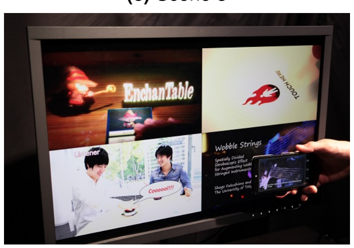
Author Keywords
Interactive Devices; LCD Displays; Augmented Reality; Imperceptible Color Vibration;

ACM Classification Keywords
H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities; I.3.6 [Methodology and Techniques]: Interaction techniques

(a) Scene 1

(b) Scene 2

(c) Scene 3



(d) Scene 4

Figure 1: A smartphone receives data embedded into each thumbnail and plays the video accordingly.

Introduction

Graphic display devices, in addition to their role as passive displays, have been recognized for their value as data transmission tools, because they can facilitate interaction between visual images and computing devices. Using data from the displays, computing devices can be tracked without any external devices and receive a variety of control data. This opens up possibilities for various position-dependent photoreactive applications in the fields of pervasive computing space [10], wearable tactile displays [8, 12, 3], and robot swarm control [2].

When we develop these applications, the data should be imperceptible so as not to impair the visual experience (Req. 1); moreover, the chosen display device should be easily available for the application to be pervasive (Req. 2). In one existing approach, data is transmitted with visible markers on an ordinary LCD display [11]. In another approach, a modified infrared light projector or modified high-speed projector must be employed [6, 5]. However, these approaches do not satisfy Req. 1 and Req. 2, respectively.

To overcome these limitations, we propose a novel method to embed imperceptible data pixel-by-pixel into images on an ordinary LCD display. The imperceptibility is achieved by fixing the luminance and vibrating only the chromaticity of the color of each pixel. An experiment designed to measure the time required to decode data revealed that our system has adequate responsiveness for HCI. The contributions of this paper are as follows:

- We demonstrated transmission of imperceptible data using an ordinary LCD display.
- We designed a technique to embed five different data values using imperceptible color vibration.
- We showed that the data can be detected with adequate responsiveness.
- We demonstrated that our method works as designed by developing an application. (Figure 1)

Related Work

The main concept of our research, i.e., embedding data into images according to position, is the same as that of PVLC [5], although our purpose is to reduce the need for specialty devices. PVLC transmits data by rapidly switching each pixel on and off and encoding data according to the on/off pattern. The maximum frequency of flicker perceptible to the human eye is called critical fusion frequency (CFF), and is known to be about 60 Hz [9]. PVLC does not require the initialization of position because information is embedded over the entire image region, not restricted to marked areas as in DBC [11]. In addition, no positional deviation between the image and the information occurs in principle because each pixel transmits the information itself. However, this technique requires a special, high-speed projector that enables every pixel to blink rapidly.

Although Kaleido [14], whose purpose is to prevent piracy of movies, employs imperceptible color vibration, it only generates noise for cameras and does not transmit understandable data. VRCodes [13] also embeds imperceptible data via color vibration. However, the data is spatially spread and not embedded pixel-by-pixel as in our method. Moreover, it only produces output in gray because it aims only to embed codes unobtrusive to human eyes. In contrast, we conceived a modulation method that can be applied to all colors.

Method

We modulate the original color of each pixel to a modulated color 1 and a modulated color 2. A 30 Hz color vibration is generated by locating the frame where modulated color 1 is used and the frame where modulated color 2 is used alternately in a 60-fps video. The information is

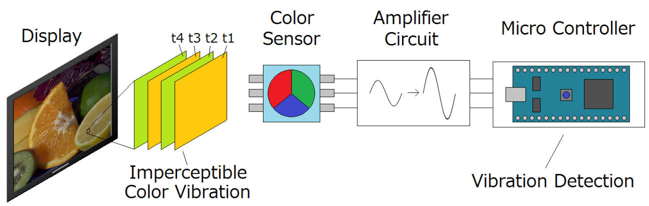


Figure 2: Proposed System

Data ('RGB')	Availability (Cause)
'000'	Yes
'001'	Yes
'010'	No (Luminance)
'011'	No (Color Deviation)
'100'	No (Color Deviation)
'101'	Yes
'110'	Yes
'111'	Yes

Table 1: Available Data

encoded through a combination of two modulated colors, and decoded in the microcontroller using the signals from the color sensor receiving the light (Figure 2). We detect whether each RGB value is vibrating or not, which allows us to transmit eight data values via simple encoding. Actually, we transmit only five data values, owing to two types of cause described below (Table 1).

Data Embedment

When we transmit data using color vibration, there are two requirements: First, it must be difficult for the human eye to perceive. Second, it must be easy for the sensor to detect.

To fulfill the first requirement, we fix the luminance and vibrate only the chromaticity. The maximum frequency of chromatic flicker perceptible to the human eye is called chromatic CFF (CCFF) and known to be about 25 Hz [4]. In the perceptually uniform L*a*b* color space, we select the two colors located at the position symmetric to the original color and having the same L* value (and different a* and b* values). L* denotes lightness and is the bijective function of the luminance; a* and b* denote chromaticity. When we show these two colors alternately using a 60 Hz display, we can generate a chromatic flicker of 30 Hz (higher than CCFF); therefore, it is predicted that only the original color will be perceived, owing to color fusion.

The second requirement is fulfilled on the basis of the features of the sensor. A color sensor is designed to output a value proportional to the square of a displayed RGB value because it contains three-channel photodiodes, each of which has sensitivity peaks $S_R, S_G, \text{and } S_B[\text{A/W}]$ at the red, green, and blue wavelengths, respectively. Thus, we define an evaluation function to predict the detectability of the vibration between the color (R1, G1, B1) and the color (R2, G2, B2), as shown below.

$$E_R(R_1, G_1, B_1; R_2, G_2, B_2) = S_R|{R_1}^2 - {R_2}^2| \quad (1)$$
$$E_G(R_1, G_1, B_1; R_2, G_2, B_2) = S_G|{G_1}^2 - {G_2}^2| \quad (2)$$
$$E_B(R_1, G_1, B_1; R_2, G_2, B_2) = S_B|{B_1}^2 - {B_2}^2| \quad (3)$$

We hypothesize that the vibration of each R, G, B can be detected if this function is large enough.

We embed the data by assigning '1'/'0' to each RGB channel. A color can be modulated by searching for the combination of two colors that transmits the intended data in the a*-b* plane. We let E_{th} be the threshold we must set. Among the infinite number of alternatives that fulfill the first requirement, the one that makes E_X more than E_{th} transmits '1' to the X channel. Similarly, if E_X is less than $E_{th}/10$, '0' is transmitted to the X channel.

The combination to transmit data value '010,' which changes only the G value, is not found in most cases because G contributes significantly to luminance Y, as defined below.

$$Y = 0.222485R + 0.716905G + 0.060610B. \quad (4)$$

Thus, we now have seven data values to be embedded.

Actually, there are other cases in which combinations cannot be found. These cases occur when the original color is at the edge of the color space, such that it cannot be expressed by the fusion of two colors. Another situation occurs when the RGB value of the color is so small that E_X cannot be obtained. In these cases, we slightly shift the

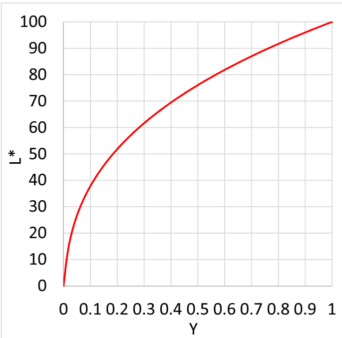


Figure 3: Relation between L and Y

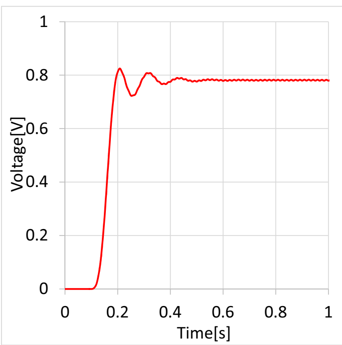


Figure 4: Output Signal of Lock-in Amplifier

original color to gray and again search for the combination. This search-and-shift procedure is iterated until a combination is found or the original color reaches gray.

Most colors can be modulated by utilizing the shift to gray. However, even if a combination is found after the shift, the expressed color will deviate from the original color. The color deviation should have as small a CIEDE2000 color difference (ΔE_{2000}) [7] (the standard currently believed to quantify the color difference we perceive the most correctly) as possible, because the modulated image should not differ significantly from the original.

We do not transmit '011' or '100' as data values because our trial showed that they cause large color deviations from certain original colors. When encoding with '011,' large deviations are observed when the original color has a small B value and does not have a large G value. In this case, there must be a certain difference in G to gain E_G, and that requires much more difference in B, owing to the small contribution of B to Y; this will cause the color to be moved out of its color space. When encoding with '100,' which changes only the R value, large deviations are observed when the color has a small G value. R also contributes to Y and differences in R have a large effect on L* if G is small, because L* has a steep gradient to a small Y, as shown in Figure 3. Eliminating values '010,' '011,' and '100' gives us five values of embeddable data (Table 1). We note that we can decode the data correctly even if the receiver lies over multiple pixels by making all colors of the same data change RGB values in the same polarity.

Vibration Detection

To decode the data, each channel of the color sensor must detect 30 Hz color vibrations, and we utilize a dual phase lock-in amplifier (LIA) as follows. A subject signal $y = A\sin\omega t$ is multiplied by two reference signals, $y_{ref1} = \sin(\omega_0 t + \phi)$ and $y_{ref2} = \cos(\omega_0 t + \phi)$, both of which have the phase difference ϕ. These two signals are filtered by a low-pass filter (LPF) whose cutoff frequency is ω_c rad/s; the sum of their squares is then calculated. Then, the direct current signal below is obtained.

$$Y = \begin{cases} \frac{A}{2} & (\omega_0 - \omega_c < \omega < \omega_0 + \omega_c) \\ 0 & (otherwise) \end{cases} \qquad (5)$$

Thus, we can detect the presence of a frequency near 30 Hz in the input signal with $\omega_0 = 2\pi \times 30$. The signal actually has a transient state, as shown in Figure 4; we call the final value a steady-state value. It should be noted that this signal depends on the LPF implemented in the LIA.

To detect vibrations correctly, we must initialize the receiver for two purposes. The first is to reduce the interference of other channels. For example, interference may cause the G channel to detect a vibration even though only R is vibrating. The second is to determine the threshold of the steady-state value, which judges whether the signal contains the intended frequency. Standard video V_0, which displays color (127, 127, 127), is first fed to the receiver. Next, video V_R displays (126, 127, 127) and (128, 127, 127) alternately, generating vibration only in R. Videos V_G and V_B are applied similarly. Letting the steady-state value of the X channel be X_0 given V_0, and that given V_Y from which X_0 is subtracted be X_Y, we define the matrix M as below.

$$M = \begin{bmatrix} 1 & R_G/R_R & R_B/R_R \\ G_R/G_G & 1 & G_B/G_G \\ B_R/B_B & B_G/B_B & 1 \end{bmatrix} \qquad (6)$$

Then, compared to the RGB variation R_s, G_s, B_s transmitted by the display, the color variation R_r, G_r, B_r received by the color sensor is considered to be

$$\begin{bmatrix} R_r & G_r & B_r \end{bmatrix}^T = M \begin{bmatrix} R_s & G_s & B_s \end{bmatrix}^T \qquad (7)$$

and therefore we can reduce the interference by the inverse

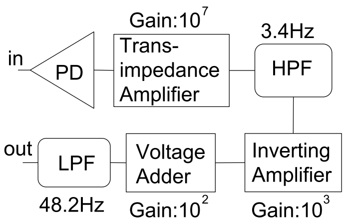


Figure 5: Amplifier Circuit

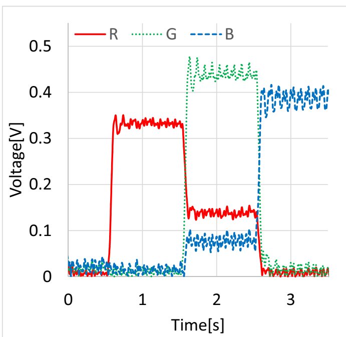


Figure 6: Before Initialization

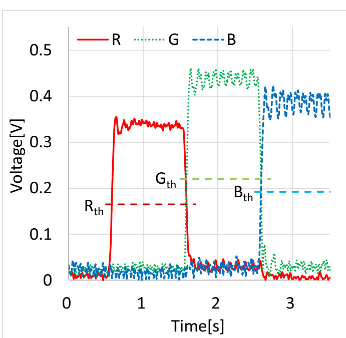


Figure 7: After Initialization

matrix calculation. Furthermore, we define the threshold of the steady-state value of each RGB R_{th}, G_{th}, B_{th},

$$\begin{bmatrix} R_{th} & G_{th} & B_{th} \end{bmatrix}^T = M^{-1} \begin{bmatrix} \frac{R_R}{2} & \frac{G_G}{2} & \frac{B_B}{2} \end{bmatrix}^T \qquad (8)$$

adopting the middle of the steady-state value in the detecting state and further reducing the interference by M. Here, although the formula (5) produces only positive values, we need the information about the sign because it is unknown whether the vibration in the detecting state and that in the non-detecting state have the same or opposite phase. We thus employ the signed LIA given the sign of $\sin\phi$.

Implementation

We use a Hamamatsu S9032-02, whose sensitivity can be coordinated by the amplifier circuit, as the color sensor. It generates values of $S_R = 0.16, S_G = 0.23, S_B = 0.18$ in formula (1), (2), (3). In addition, we use an mbed LPC1768 (NXP Semiconductors) as the microcontroller. A block diagram describing the function of the amplifier circuit is shown in Figure 5. In the microcontroller, the produced signal is filtered by a 25 Hz high-pass filter, which reduces the noise appearing in the final result; the signal is then fed to the LIA, which has a 10 Hz LPF. Both filters are third-order Butterworth filters. At the modulation, we set the threshold of evaluation function E_{th} to 100; however, this needs further analysis because the sensitivity of the amplifier circuit and the amount of color deviation depend on this value.

To confirm the validity of the initialization, we show the two signals obtained when the initialization video was displayed using this implementation before and after the initialization (Figure 6, 7). The resulting matrix (M) is shown below.

$$M = \begin{bmatrix} 1.000 & 0.304 & 0.042 \\ 0.094 & 1.000 & 0.106 \\ 0.025 & 0.211 & 1.000 \end{bmatrix} \qquad (9)$$

Experiment

To evaluate whether the proposed system has adequate responsiveness for HCI, we conducted an experiment in which the delay was measured from the moment a non-modulated video made the transition to a modulated one (rise) or the opposite (decay) to the moment the receiver decoded the data correctly. Accurately speaking, "The moment the receiver decoded the data correctly" means the first time when all the steady-state values of the '1'/'0' channels were more/less than the threshold (i.e., detecting correctly) and that state was maintained steadily (i.e., no further detection errors occurred). For instance, the signal shown in Figure 8 is obtained from the data value '110' and its delay is defined in Figure 12.

The non-modulated video displays only one color (R, G, B) and the modulated video alternately displays two colors, to which the (R, G, B) is encoded with the data value D. Both videos have 1024×768 pixels and a frame rate of 60 fps. R, G, and, B will each have one of three values: 0, 127, or 255. Further, D will have a value of '001,' '101,' '110,' or '111.' Thus, the number of combinations is $3 \times 3 \times 3 \times 4 = 108$. Here, to detect whether the video is modulated, we provided 64×48 pixels that become black when the video is non-modulated and white when the video is modulated; these appear at the bottom-left corner of the video. We fixed the photodiode circuit at that position, generating the signal as 'PD' in Figure 8. We measured the delay once for each of the 108 combinations, fixing the color sensor 3 cm vertically away from the center of the display. The measurement was conducted with a color calibrated liquid crystal display (CG-2420, EIZO); its luminance was 60 cd/m^2 in the assembled darkroom (ADR-F2, ASONE), into which no ambient light could enter. (Figure 9)

A box-plot of the results (excluding the combination which

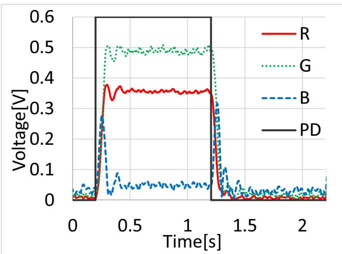


Figure 8: Example of Waves

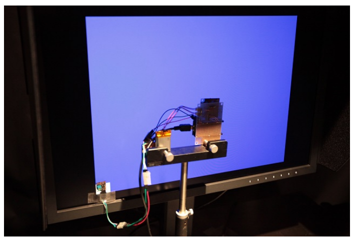


Figure 9: Experiment Setup. The lighting was actually turned off.

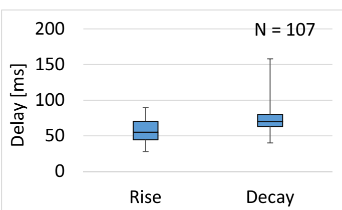


Figure 10: Box-Plot of Delay

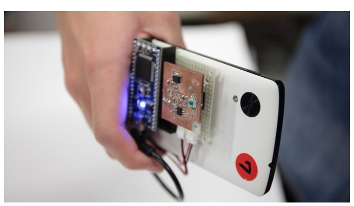


Figure 11: Receiver on Phone

was not decoded correctly) is shown in Figure 10. The rise delays and decay delays were less than 100 ms for 100% and 96% of the combinations, respectively, revealing that this system achieved adequate responsiveness for HCI [1]. The delays were larger when the modulated colors had a different '0' channel value (e.g., the G value in '101') from that of the original, which occurred owing to color deviations. This difference results in a step input, and temporarily causes a false detection in a channel that should not detect the vibration. Delays may also be increased when R or B vibrations are so intense that they exceed the input range of the microcontroller. This exceeding hinders the interference reduction process, and the insufficient rise causes G to be unstable around the threshold. It should be noted that this delay is not unique to the apparatus used here, because it depends mostly on the filter designs of the amplifier circuit and the signal processing in the microcontroller.

As for the modulations, 86% had color deviations within 1.0 in ΔE_{2000}; 10% were between 1.0 and 3.5, and the remaining 4% were between 5.0 and 7.0. It is assumed that these deviations would have a negligible impact on the visual experience. Furthermore, only one of the 108 combinations was not decoded correctly. As stated above, this was caused by malfunctions in the interference reduction process, which will be improved by coordinating the sensitivity of the receiver.

Application

To demonstrate the validity of our method, we developed a data acquisition application employing an LCD display and a mobile phone, which works as Figure 1 illustrates. The four different tags ('001,' '101,' '110,' and '111') embedded into the region of each image were decoded by the receiver circuit mounted on the mobile phone, as shown in Figure11, and the phone plays a preinstalled video in accordance

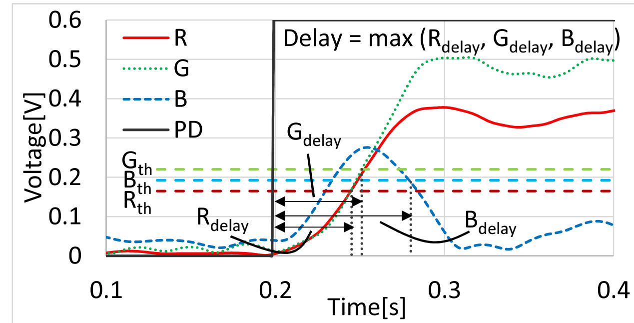


Figure 12: Definition of Delay

with the tag. This application suggests the potential for this method to be employed in practical scenarios.

Conclusion and Future Work

We proposed a novel method to embed imperceptible data into images on an LCD display by vibrating only the chromaticity. Moreover, we showed that our system has adequate responsiveness for HCI and developed an application demonstrating that our method works as designed.

A limitation of our method is false detection caused by the step input. For instance, we cannot decode data correctly in situations where the receiver moves in and out rapidly between two regions.

In future work, we aim to investigate the robustness of decoding in situations where the receiver moves on the display. It also should be discussed whether the flicker is truly imperceptible to the user, although the imperceptibility is supported by the reference and we, at least, could not perceive any flicker. There are a number of aspects remaining to be discussed.

Acknowledgements

This work is supported by CREST, JST.

References

[1] James R. Dabrowski and Ethan V. Munson. 2001. Is 100 Milliseconds Too Fast?. In *Proceedings of CHI '01 Extended Abstracts on Human Factors in Computing Systems - CHI EA '01*. 317–318.

[2] Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. 2016a. Phygital Field: an Integrated Field with a Swarm of Physical Robots and Digital Images. In *SIGGRAPH ASIA 2016 Emerging Technologies on - SA '16*. Article No.2.

[3] Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. 2016b. Sensible Shadow: Tactile Feedback from Your Own Shadow. In *Proceedings of the 7th Augmented Human International Conference 2016 on - AH '16*. Article No.23.

[4] Yi Jiang, Ke Zhou, and Sheng He. 2007. Human visual cortex responds to invisible chromatic flicker. *Nature Neuroscience* 10, 5 (may 2007), 657–662.

[5] Sho Kimura, Ryo Oguchi, Hideo Tanida, Yasuaki Kakehi, Keita Takahashi, and Takeshi Naemura. 2008. PVLC Projector: Image Projection with Imperceptible Pixel-Level Metadata. In *ACM SIGGRAPH 2008 Posters on - SIGGRAPH '08*. Article No.135.

[6] Johnny C. Lee, Scott E. Hudson, and Paul H. Dietz. 2007. Hybrid Infrared and Visible Light Projection for Location Tracking. In *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*. 57–60.

[7] M. R. Luo, G. Cui, and B. Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research & Application* 26, 5 (oct 2001), 340–350.

[8] Jun Rekimoto. 2009. SenseableRays: Opto-Haptic Substitution for Touch-Enhanced Interactive Spaces. In *Proceedings of the 27th International Conference Extended Abstracts on Human factors in Computing Systems - CHI EA '09*. 2519–2528.

[9] Henry N. Ricciuti and Henryk Misiak. 1954. The Application of the Constant Method in Determining Critical Flicker Frequency (CFF). *The Journal of General Psychology* 51, 2 (oct 1954), 213–219.

[10] Dominik Schmidt, David Molyneaux, and Xiang Cao. 2012. PICOntrol: Using a Handheld Projector for Direct Control of Physical Devices through Visible Light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. 379–388.

[11] Maki Sugimoto, Kazuki Kodama, Akihiro Nakamura, Minoru Kojima, and Masahiko Inami. 2007. A Display-Based Tracking System: Display-Based Computing for Measurement Systems. In *Proceedings of 17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*. 31–38.

[12] Haruya Uematsu, Daichi Ogawa, Ryuta Okazaki, Taku Hachisu, and Hiroyuki Kajimoto. 2016. HALUX: Projection-based Interactive Skin for Digital Sports. In *ACM SIGGRAPH 2016 Emerging Technologies on - SIGGRAPH '16*. Article No.10.

[13] Grace Woo, Andy Lippman, and Ramesh Raskar. 2012. VRCodes: Unobtrusive and Active Visual Codes for Interaction by Exploiting Rolling Shutter. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 59–64.

[14] Lan Zhang, Cheng Bo, Jiahui Hou, Xiang-Yang Li, Yu Wang, Kebin Liu, and Yunhao Liu. 2015. Kaleido: You Can Watch It But Cannot Record It. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*. 372–385.