DOI:10.1145/3375571

Automating Visual Privacy Protection Using a Smart LED

By Shilin Zhu, Chi Zhang, and Xinyu Zhang

Abstract

The ubiquity of mobile camera devices has been triggering an outcry of privacy concerns, whereas existing privacy protection solutions still rely on the cooperation of the photographer or camera hardware, which can hardly be enforced in practice. In this paper, we introduce LiShield, which automatically protects a physical scene against photographing, by illuminating it with smart LEDs flickering in specialized waveforms. We use a model-driven approach to optimize the waveform design, so as to ensure protection against the (uncontrollable) cameras and potential image-processing–based attacks. We have also designed mechanisms to unblock authorized cameras and enable graceful degradation under strong ambient light interference. Our prototype implementation and experiments show that LiShield can effectively destroy unauthorized capturing while maintaining robustness against potential attacks.

1. INTRODUCTION

Cameras are now pervasive on consumer mobile devices, such as smartphones, tablets, drones, smart glasses, first-person recorders, etc. The ubiquity of these cameras, paired with pervasive wireless access, is creating a new wave of visual sensing applications, for example, autonomous photograph, quantified-self (life-logging), photo-sharing social networks, physical analytics in retail stores,[12] and augmented reality applications that navigate users across unknown environment.[19] Zooming into the photo-sharing application alone, statistics report that 350 million photos/videos are uploaded to Facebook every day, majority of which are from mobile users.[15] Many of these applications automatically upload batches of images/videos online, with a simple one-time permission from the user. Although these technologies bring significant convenience to individuals, they also trigger an outcry of privacy concerns.

Privacy is ultimately a subjective matter and often varies with context. Yet, many of the privacy-sensitive scenes occur in indoor environment and are bound to specific locations. For example, recent user studies[2] showed that people's acceptability of being recorded by augmented reality glasses has a strong correlation with location. User studies of life-logging cameras[6] also indicate that 70.2% of the cases when the user disables capturing are associated with specific locations. In numerous real-world scenarios, cameras are forbidden, for example, concerts, theaters, museums, trade shows, hospitals, dressing rooms and exam rooms, manufacturing plants, etc. However, *visual privacy protection in such passive physical spaces still heavily relies on rudimentary approaches* like warning signs and human monitors, and there is no way to automatically enforce the requirements. In personal visual sensing applications like life-logging, even if a user were to disable the camera in private space, malware could perform remote reconnaissance and target visual theft by hijacking the victim's camera.[16]

In this paper, we propose LiShield, a system that deters photographing of sensitive indoor physical space and automatically enforces location-bound visual privacy protection. LiShield protects the physical scenes against undesired recording without requiring user intervention and without disrupting the human visual perception. Our key idea is to illuminate the environment using smart LEDs, which are intensity-modulated following specialized waveforms. We design the waveform in such a way that its modulation pattern is imperceptible by human eyes but can interfere with the image sensors on mobile camera devices.

Adversary model and protection goals. LiShield aims to prevent ad-hoc capturing from benign camera-phone holders. The physical space under protection can be static or dynamic. In either case, we assume that one or multiple LiShield-enabled smart LEDs can cover the whole area, while providing illumination similar to normal office lighting without human-perceptible flickering. Although conventional lighting and sunlight may co-exist with LiShield's smart LEDs, covering the entire target scene with LiShield will ensure the strongest protection.

Now consider an unauthorized user (attacker) who wants to take pictures or videos within the protected space, with cameras and ashes embedded in smartphones, but with no professional equipment such as global shutter cameras, filters, or tripods. The attacker has full control over the camera parameters (e.g., exposure time, capturing time, and white-balancing) and can run any postprocessing on the captured images. Nonetheless, with LiShield's protection, the image frames are corrupted, so that major fraction of each frame is either blank or overexposed, whereas colors are distorted (Section 2), which deters image viewing/sharing.

In addition, LiShield should possess the following capabilities for practical usage scenarios: (i) allowing an authorized camera, which shares secret configuration information with the LED, to recover the image or video frames it captures. (ii) when strong ambient light interferes with the smart LED, LiShield cannot ensure full protection, but it can still emit structured light which embeds invisible "barcode" into the physical environment. The embedded information can convey a "no distribution" message, allowing online servers (e.g., from Facebook and Instagram) to block and prevent the image from being distributed.

The original version of this paper appeared in *ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017.

How does LiShield disrupt camera image capturing? Cameras and human eyes perceive scenes in fundamentally different ways. Human eyes process continuous vision by accumulating light signals, whereas cameras slice and sample the scene at discrete intervals. Consequently, human eyes are not sensitive to high frequency flickers beyond around 80 Hz either in brightness or chromaticity,[7] whereas cameras can easily pick up flicker above a few kHz.[20] Equally importantly, human eyes perceive brightness in a nonlinear fashion, which gives them huge dynamic range, whereas cameras easily suffer from overexposure and underexposure when signals with disparate intensities mix in the same scene.

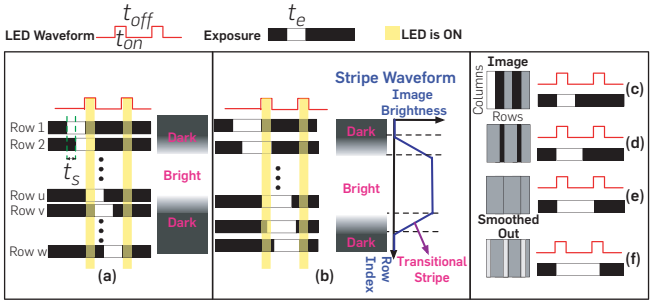
Unlike professional or industrial cameras which may have global shutters that mimic human eyes to some degree, nearly all consumer digital cameras, pinhole cameras, and smartphones use the rolling shutter sampling mechanism,[8] which is the main contributor to their high-frequency sensitivity. When capturing an image frame, a *rolling shutter camera exposes each row sequentially*.

LiShield harnesses the disparity between cameras and eyes to disrupt the camera imaging without affecting human vision. It modulates a smart LED to generate high-frequency flickering patterns. The reflection intensity (or brightness) of target scene also flickers following the same pattern as the LED's illumination, albeit at reduced intensity due to reflection loss. LiShield uses the On-Off Keying (OOK) as the basic modulation waveform (Figure 1), which does not require complicated analog front-ends and is widely supported by smart LEDs. Due to rolling-shutter sampling, the rows of pixels that are fully exposed in the ON period will be bright, and rows in the OFF period become dark, thus causing striped patterns on the captured image (Figure 1(a, b)). Partially exposed rows experience moderate brightness. Meanwhile, human eyes can only perceive the smooth averaged intensity as long as the OOK frequency goes beyond 80 Hz.[7, 21]

In addition, LiShield can turn different numbers of LED bulb/chip on to generate different intensities and control the RGB channels of the LEDs to vary the color. In Section 2, we will show how such flickering corrupts the spatial patterns captured by a camera.

Summary of results. We have implemented LiShield based on a customized smart LED, which allows reconfiguration of intensity modulation waveforms on each color channel. Our experiments on real world scenes demonstrate that LiShield can corrupt the camera capturing to an illegible level, in terms of the image brightness, structure, and color. The impact is resilient against possible attacks, such as multi-frame combining and denoising. On the other hand, it enables authorized cameras to recover the image perfectly, as if no modulation is present. Even under strong sunlight/flashlight interferences, LiShield can still sneak barcode into the physical scenes, which can be decoded with around 95% accuracy.

Figure 1. (a) and (b) Bright, dark and transitional stripes and their width changing with exposure time; (c)–(f) stripe pattern of image changes under different exposure times.



2. DISRUPTING CAMERA CAPTURING USING SMART LIGHTING

2.1. Maximizing image quality degradation

LiShield aims to *minimize the image capturing quality by optimizing the LED waveform, characterized by modulation frequency, intensity, and duty cycle*. To this end, we derive a model that can predict the image quality as a function of the LiShield's waveform and attacker's camera parameters. For simplicity, we start with monochrome LED with a single color channel that illuminates the space homogeneously. We denote P as the reference image taken under a nonflickering LED and Q as the one taken under LiShield's LED with the same average brightness. We assume each image has m rows and n columns, and the light energy received by each pixel is denoted by $P(i, j)$ and $Q(i, j)$, respectively. Our model focuses on two widely adopted image quality metrics: *PSNR*, which quantifies the disruption on individual pixel intensity levels, and *SSIM*,[18] which measures the structural distortion to the image (i.e., deformation effects such as stretching, banding, and twisting). In general, the minimum PSNR and SSIM corresponding to acceptable viewing quality are in the range of 25–30 and 0.8–0.9, respectively.[1]

Decomposing the image. To compute the image quality, we need to model the intensity and width of each stripe caused by LiShield. As illustrated in Figure 1, we use t_{on}, t_{off}, and I_p to denote the on/off duration and peak intensity of the flickering light source, and t_e and t_s are the exposure time and sampling interval of the rolling shutter camera. For convenience, denote the period of the light source as $t_l = t_{on} + t_{off}$ and duty cycle as $D_c = t_{on}/t_l$. For pixel j in row i which starts exposure at time t_i, its light accumulation would be:

$$Q(i,j) = \alpha_{i,j} \int_{t_i}^{t_i+t_e} \pi_l(\tau) d\tau \quad (1)$$

where $\alpha_{i,j}$ is the aggregated path-loss for pixel (i, j), such as attenuation and reflection on the photographed object, and $\pi_l(\tau)$ represents the illumination waveform of the LED:

$$\pi_l(\tau) = \begin{cases} I_p, & 0 < \tau \bmod t_l \leqslant t_{on} \\ 0, & t_{on} < \tau \bmod t_l \leqslant t_l \end{cases} \quad (2)$$

When the camera's exposure time is equal to or shorter than the LED's OFF period ($t_e \leqslant t_{off}$), the image will contain rows that are completely dark (Figure 1(c)). On the other hand, when $t_e > t_l$, one row-exposure period of the camera will overlap multiple ON periods of the LED, accumulating higher intensity (Figure 1(f)). The special case happens when $t_e = t_l$, where the integration of LED waveform and exposure has fixed value, which eventually smooths out dark stripes (Figure 1(e)). Without loss of generality, assume that the

exposure starts right at the beginning of the ON period. Let $N = \lfloor t_e/t_l \rfloor$, which is the number of whole flicker cycles covered by exposure time, and $t_{\text{rem}} = (t_e \bmod t_l)$, which is the remaining duration after multiple whole cycles, and the light accumulation of the brightest rows Q_B is:

$$Q_B(i,j) = \begin{cases} \alpha_{i,j} I_p (N t_{\text{on}} + t_{\text{rem}}), & 0 < t_{\text{rem}} \leqslant t_{\text{on}} \\ \alpha_{i,j} I_p (N+1) t_{\text{on}}, & t_{\text{on}} < t_{\text{rem}} \leqslant t_l \end{cases} \quad (3)$$

Since the brightest rows appear when the exposure captures most ON periods possible (e.g., row 2 to row u in Figure 1(a)) and rolling shutter effect converts temporal variation into pixels with sampling interval t_s, the width of Q_B is:

$$W_B = \left| t_{\text{rem}} - t_{\text{on}} \right| / t_s \quad (4)$$

Likewise, when the exposure captures least ON periods possible (e.g., from row v to row w in Figure 1(a)), we get the darkest rows with light accumulation Q_D:

$$Q_D(i,j) = \begin{cases} \alpha_{i,j} I_p N t_{\text{on}}, & 0 < t_{\text{rem}} \leqslant t_{\text{off}} \\ \alpha_{i,j} I_p (N t_{\text{on}} + t_{\text{rem}} - t_{\text{off}}), & t_{\text{off}} < t_{\text{rem}} \leqslant t_l \end{cases} \quad (5)$$

and the width of Q_D is:

$$W_D = \left| t_{\text{rem}} - t_{\text{off}} \right| / t_s \quad (6)$$

We refer to a collection of consecutive brightest rows as "bright stripe" and consecutive dark rows as "dark stripe," as shown in Figure 1(b). In addition, there exist intermediate rows containing linear intensity transition between dark and bright, referred to as "transitional stripe."

Meanwhile, if the LED were not flickering and provided the same average brightness, the pixel intensity would be:

$$P(i,j) = \alpha_{i,j} I_p \cdot D_c \cdot t_e \quad (7)$$

Since $D_c \cdot t_e$ remains constant within each frame, *the image captured under LiShield is equivalent to the original image multiplied by a piecewise function* (cf. Equations (3) and (5)).

Other common camera parameters (i.e., ISO, white balance, and resolution) do not affect the structure of the stripe pattern. By default, we assume that the attacker sets the ISO to its minimum (typically 100) to maximally suppress noise.

Optimizing the LED waveform. Since the stripe pattern follows a piecewise function, a closed form expression of PSNR and SSIM becomes hard to analyze. We thus use numerical simulation to evaluate the impact of LiShield, based on the above model. We generate the piecewise function with $Q_B(i,j)$, W_B, $Q_D(i,j)$, and W_D and multiply it on reference images to obtain the disrupted image Q just like the process inside real cameras. We assume t_s = 1/75,000 s, which matches the capability of a Nexus 5 camera. The quality metrics are calculated between the reference image P and LiShield-corrupted image Q with same average intensity. We make pixel intensity range infinite, which allows quantifying quality loss caused by overexposure.

By default, we use OOK waveform with frequency f = 100 Hz, peak intensity I_p = 10 kLx, and duty cycle D_c = 0.5. We vary one parameter while keeping others to the defaults. Note that the typical light intensity is ~700 Lx in office environments and ~100,000 Lx outdoor in sunny days. Our numerical results lead to the following design choices for LiShield.

(i) A single frequency cannot ensure robust protection. For a given waveform frequency f, there exist several exposure time settings that lead to high-quality images. This is because when $t_e \approx N t_l$, the stripes become smoothed out (Figure 1(e)). Although the waveform parameters are unknown to the attacker, a determined attacker may launch a brute-force search for the t_e that satisfies this condition, thus circumventing the protection. To counteract such attackers, *LiShield includes a countermeasure called frequency randomization*, which we discuss in "Frequency scrambling" section.

(ii) LiShield must leverage overexposure to prevent attackers from using long exposures. The image quality increases with exposure time t_e, until overexposure happens, because longer exposure leads to more waveform cycles being included as a constant base in the brightness of the stripes (larger N in Equations (3) and (5)), making the contrast of stripes Q_B/Q_D lower and weakening the quality degradation. *LiShield should leverage overexposure to limit attacker's exposure time*. On the other hand, when exposure goes beyond a threshold, the image always suffers from over-exposure. If not, the image is always corrupted due to the dominance of stripes under LiShield's frequency randomization mechanism ("Frequency scrambling" section). With power efficiency and eye health in mind, *LiShield sets I_p to 20 kLx by default*. Optimal parameters may vary slightly across different scenes (e.g., different reflectivity) but can be easily obtained by running the aforementioned simulation.

2.2. Circumventing potential attacks

Based on the foregoing analysis, we identify the following potential holes that can be exploited by attackers to overcome the striping effect. *(i) Manual exposure attack*. If an attacker can configure t_e to satisfy $t_e \approx N t_l$, it can guarantee that every row receives almost the same illumination, thus eliminating the stripes during a capture (Figure 1(e)). In practice, t_l is unknown to the attacker, but it can try to capture images with different t_e, until seeing a version without obvious stripes. *(ii) Multiframe attack*. When the scene is static, an attacker may also combine multiple frames (taking a video and playback) to mitigate the stripes with statistical clues, for example, by averaging or combining rows with maximum intensities from multiple frames. Note that the attacker must keep the camera highly stable, otherwise even pixel-level shift will cause severe deformation when combining multiple frames. *(iii) Post-processing attack*. Common postprocessing techniques (e.g., denoising and de-banding) might be used to repair the corrupted images.

In what follows, we introduce countermeasures to the first two attacks. In Section 6.4, we will verify that LiShield's distortion does not fit empirical noise or banding models, so the common post-processing schemes become ineffective.

Frequency scrambling. To thwart the manual exposure attack, we design a *frequency scrambling* mechanism, which packs multiple waveforms with different frequencies within each image frame duration. Since the camera exposure time

t_e is always fixed within each frame, no single t_e can circumvent all the frequency components.

However, we cannot choose and switch the flickering frequencies in an arbitrary manner, for three reasons. (i) Multiple frequency values that share a common divisor can satisfy $t_e = Nt_l$ under the same t_e (recall N can be an arbitrary integer). We need to ensure the common divisor is small enough (i.e., least common multiplier of t_l large enough), so that overexposure occurs even for the smallest N. (ii) Frequencies should be kept low to maximize image corruption, as evident in Optimizing the LED waveform section, since camera's analog gain decreases at high frequencies.[20] (iii) Switching between different frequencies may create an additional level of modulation, which will spread the spectrum and generate unexpected low frequency components that become perceivable by eyes.

To explore the design space under these constraints, suppose we switch among M frequencies $f_1, f_2, \ldots, f_M$ (in ascending order) at a switching rate f_B. The whole pattern thus repeats itself at rate $f_p = f_B/M$. To pack at least M different frequencies in an image frame, we need $f_B > (M-1)f_r$ or, preferably, $f_B > Mf_r$, where f_r is the frame rate, typically around 30 Hz (fps). To maximize image corruption, we choose the smallest value for f_1 (i.e., $f_1 = f_B$) and empirically set $f_n = f_B + (n-1)\Delta f$, $n = 2, 3, \ldots, M$, where Δf is frequency increment, set to $\Delta f \neq f_B$ to lower the common divisor frequency.

The frequency scrambling can be considered as an M-FSK modulation, thus creating side lobes around each scrambling frequency, spacing f_p apart (Interested readers can refer to the full version of this work for the theoretical underpinning.[22]). These side lobes might appear at low-frequency region and become perceptible by eyes. To ensure no side lobe exists below the perceivable threshold $f_{\text{th}} \approx 80$ Hz, we need a small M and large f_B and hence higher flickering frequency components f_n. Yet, increasing the flickering frequencies may weaken LiShield's protection. To find the optimal Δf and showcase the effectiveness of the frequency scrambling, we repeat the numerical simulation (Section 2.1) to evaluate the attacker's maximum image quality. Based on the simulation, we set $\Delta f = 50$ Hz to maximize image disruption. The optimal Δf for other peak intensity settings can be obtained following a similar procedure.

Illumination intensity randomization. If attackers repetitively capture a static scene for a sufficiently long duration, they may eventually find at least one clean version for each row across all frames, thus *recovering* the image. LiShield can increase the number of frames needed for image recovery, so that the attack becomes infeasible unless the camera can stay perfectly still over a long period of time, during which the attackers may have already been discovered by the owners of the physical space. LiShield achieves the goal by employing illumination intensity randomization, where it randomly switches the magnitude of each ON period across multiple predefined levels, which extends the attacker's search space.

3. SCENE RECOVERY WITH AUTHORIZED CAMERAS

To allow authorized users to capture the scene while maintaining protection against unauthorized attackers, we need to impose additional constraints on the LED waveform. LiShield's solution leverages a secure side channel (e.g., WiFi[4]) between authorized users and the smart LED, which conveys secret information such as frame timing and waveform parameters.

A naive solution is to stop flickering when authorized users are recording. However, since attackers may be co-located with the authorized users, this enables them to capture one or more frames that have part of the clean scene, which compromises privacy and security. Instead, we design special waveforms for the LED to counteract such cases.

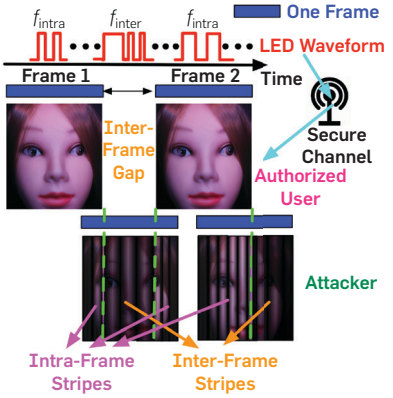
3.1. Authorized video recording

To authorize a camera to capture a dynamic scene, each individual frame within the video must be recoverable. To achieve this, the authorized camera needs to convey its exposure time setting t_e^u to the smart LED via the secure side channel and synchronize its clock (for controlling capturing time) with the smart LED's clock (for controlling the waveform), so the smart LED can send recoverable waveforms precisely during the capture of the authorized camera. State-of-the-art time synchronization mechanisms (e.g.,[4]) can already achieve μs of accuracy, sufficient to synchronize the LiShield smart LED with camera at a resolution that is finer than the rolling shutter period (typically tens of μs).

Recall that the camera can evade the striping effects if $t_e = Nt_l$. So to authorize the user with exposure t_e^u, LiShield simply needs to set its flickering frequency $f_a = 1/t_l = N/t_e^u$ ($N = 1, 2, \ldots$) and maintain its peak intensity within each frame. In addition, the t_e^u and corresponding flickering frequency f_a can be varied on a frame by frame basis, making it impossible for an attacker to resolve the correct exposure time by trial-and-error (Section 2.2).

Meanwhile, when the authorized camera is not recording at its maximum possible rate, there will be an interval (i.e., inter-frame gap) where the camera pauses capturing. LiShield packs random flickering frequencies f_{inter} other than $f_{\text{intra}} = f_a$ into the interframe gap, so as to achieve the same scrambling effect as described in "Frequency scrambling" section, without compromising the authorized capturing, as shown in Figure 2.

Figure 2. Enabling authorized users to capture dynamic scenes while corrupting unauthorized users.



3.2. Static scene recovery

When the target scene is static, the authorized user may capture a few complementary frames at a specific time to recover the scene as depicted in Figure 3, where frequency and intensity randomization (Section 2.2) are employed in each frame to ensure robustness. Although it does require recording a very short video, the process is extremely short (200 ms at most) and barely noticeable to the authorized user. Meanwhile, an out-of-sync attacker will still receive corrupted images that cannot reconstruct the original scene by direct frame combination.

Suppose a static scene is to be recovered using L_f frames, referred to as *critical frames*. To prevent attackers from launching the multiframe attack, the timing of the critical frames is negotiated only between the smart LED and the authorized user through the secure side channel. These L_f frames together must contain the information of the entire scene, that is, they must be complementary, as shown in Figure 3. Meanwhile, all other frames will follow the normal flickering pattern. Since the attackers can neither identify nor predict the timing of the critical frames, the best they can do is to launch the brute-force multiframe attack, which has proven to be ineffective ("Illumination intensity randomization" section).

4. AUTOMATIC PHYSICAL WATERMARKING FOR PRIVACY ENFORCEMENT

High-intensity ambient light sources (e.g., sunlight, legacy lighting, and ash lights) can create strong interference to LiShield's illumination waveform, degrading the contrast by adding a constant intensity to both the bright and dark stripes, which may weaken LiShield's protection. In such scenarios, LiShield degrades itself to a *barcode mode*, where it embeds barcode in the physical scene to convey privacy policies. The barcode forms low-contrast stripes, which may not fully corrupt the images of the scene, but can still be detected by online photo-distributing hubs (e.g., social website servers), which automatically enforce the policies, without co-operation of the uploader or evidence visible by naked eye. LiShield forms the watermark with just a single light fixture, instead of active displays (e.g., projectors) that are required by conventional systems. The key challenge here is how should LiShield encode the information, so that it can be robustly conveyed to the policy enforcers, despite the (uncontrollable) attacker camera settings?

Embedding. LiShield's barcode packs multiple frequencies in every image following "Frequency scrambling" section but aims to map the *ratios between frequencies* into digital information. Suppose LiShield embeds two waveforms with frequencies F_0 and F_1, it chooses the two frequency components such that F_1/F_0 equals to a value R_p well known to the policy enforcers. In other words, the presence of R_p conveys "no distribution/sharing allowed." Although width of stripes is affected by sampling interval t_s and exposure time t_e (Figure 1(a) and (b)), ratio of stripe widths resulted from two frequencies (which equals to R_p) remains constant. Therefore, this encoding mechanism is robust against camera settings.

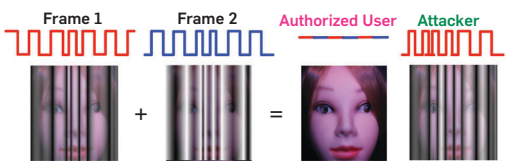
Since physical scenes usually comprise a mix of spatial frequencies, and spectral power rolls off in higher spatial frequencies, thanks to camera lenses' limited bandwidth while temporal frequencies are unaffected, LiShield's barcode uses frequencies that are much higher than the natural frequencies (>400 Hz) in the scene to reduce interference. It is worth noting that since the rolling-shutter sampling rate of all cameras falls in a range (30 kHz to slightly over 100 kHz[20]), LiShield limits its highest flickering frequency to 15 kHz, which respects the Nyquist sampling theorem, so that the barcode can eventually be recovered without any aliasing effect.

To further improve robustness, LiShield leverages redundancy. It embeds multiple pairs of frequency components to make multiple values of R_p either at different rows of the image or in different color channels, further mitigating interference caused by intrinsic spatial patterns within the scene.

Detection. Since the barcode contains M frequencies, that is, $f_n = f_B + (n-1)\Delta f$, $n = 2, 3, \ldots, M$ ("Frequency scrambling" section), there are $M_R = C_M^2$ possible frequency ratio values across the image for monochrome barcode ($M_R = C_{M\times 3}^2$ for RGB barcode). Δf must be set large enough to avoid confusion (Δf = 200 Hz in experiments). The barcode decoder, running on the policy enforcer, recognizes the image as protected if there are at least M_b values that roughly match the known ratio R_p, that is, when the value falls within T_b of R_p. Suppose M_{att} is the number of R_p removed by manual exposure attack (Section 2.2), these parameters are determined by bounding the false positive rate following an empirical procedure (to be discussed in Section 6.3).

To detect the frequency ratios, LiShield averages the intensity of each row to get a one-dimension time series s_r. LiShield then runs FFT over each series to extract the M_p strongest frequencies. Finally, LiShield combines all unique frequencies extracted and computes all frequency ratios. The redundancy in barcode ensures that it can be robustly detected.

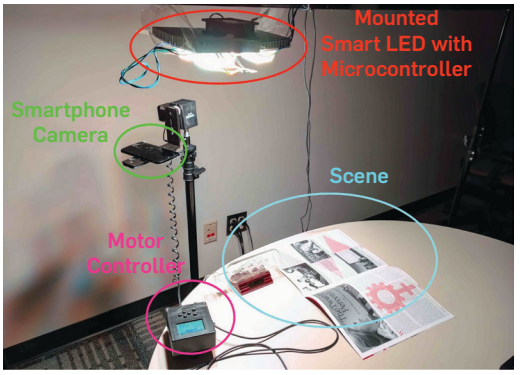
Figure 3. The impact of multiframe recovery on authorized user and attacker, respectively.



5. IMPLEMENTATION

Testbed setup. Figure 4 shows our smart LED prototype, and the target scenes containing five capture-sensitive objects (document and painting are 2-D objects and others are all 3-D objects). We mount the LED inside a diffusive plastic cover similar to conventional ceiling light covers. We use a programmable motor to hold the camera and control its distance/orientation, in order to create static or dynamic scene setup in a repeatable manner.

Figure 4. Experimental setup of LiShield.



Smart LED modules. Commercial-of-the-shelf (COTS) household LED bulbs rely on integrated drivers to regulate LED's current. A dimming input is usually available on these drivers for controlling the current dynamically. We build our smart bulb based on the same topology as these COTS LED bulbs. We use 19V DC laptop power supplies and NCL30160 LED drivers, which allow dimming at nearly 100 kHz with arbitrary OOK waveform. The smart bulb has built-in independent RGB/white channels for controlling color/intensity. Each channel can be controlled by a separate waveform, with four LED chips in series, at driving current of 800 mA. In total, the three channels consume approximately 25 W peak power, close to common office LED troffer fixtures. However, since LiShield's OOK waveform has a duty cycle much lower than one (Section 2), the actual perceptible brightness is significantly lower.

The dimming input signals of each channel are controlled by an STM32 microcontroller unit (MCU), which generates the OOK waveform as specified by LiShield. For flexible reconfiguration, we generate digitized waveforms in MATLAB on a laptop or Android app on a smartphone instead, which are then passed to the MCU via USB.

Android app for normal, authorized and attacker's cameras. Unless otherwise noted, we use Nexus 5 with stock ROM as our benchmark device. We assume that normal users use the stock camera app with default settings (such as auto exposure), whereas a malicious attacker can manually tune the camera parameters (e.g., using the Open Camera app). By default, the camera ISO is set to the lowest value (100), since it is most beneficial for attackers, as it allows longer exposure to smooth out the stripes without causing overexposure. To implement the authorization mechanism (Section 3), we develop a specialized app for the authorized smartphone, which uses Android's Camera2 API[5] to precisely control the exposure time, as well as communicating with the smart LED's MCU via USB. Since current Android camera APIs do not support precise frame timing, the app requests the smart LED to synchronize with the camera by altering its waveform.

Attacker's image processing. We have implemented the attacking algorithms in Section 2.2, which are specifically designed to combine/process the captured image, aiming to eliminate LiShield's stripe distortion. In addition, we implement classical image processing techniques, such as denoising and debanding, which may be attempted by attackers. For denoising, we use the Haar-wavelet thresholding, non-local-means (NLmeans), and BM3D, which are among the most popular algorithms.[14] As for debanding, we use the Banding Denoise and Unstrip in the G'MIC plugin.

Metrics. Small displacement and vibration of the camera are inevitable in physical environment, which is known to affect the SSIM. Thus, we quantify the image quality degradation with the enhanced CW-SSIM,[13] which is insensitive under such translations. PSNR shows similar trends with SSIM. Besides, we employ the CIEDE2000[9] to compute the degradation of the images' color quality when the RGB LED is used.

6. EXPERIMENTAL EVALUATION

6.1. Effectiveness of physical scene disruption

Impact of scenes and device heterogeneity. We first verify LiShield's basic protection scheme (Section 2) with five static scenes, monochrome LEDs, and OOK waveform without frequency randomization, although the attacker's camera uses auto-exposure. Without LiShield, the measured image quality stays high, with PSNR > 30 dB and CW-SSIM > 0.9 (slightly lower than simulation results due to digital noises in real cameras). LiShield degrades the image quality to 3–10 dB for PSNR and 0.25–0.6 for CW-SSIM (Figure 5). We cross-validate the impact of LiShield on 10 common mobile cameras. Although the image quality varies slightly due to different sampling rates across devices, the quality remains at an intolerably low level across devices. Thus *LiShield's protection mechanism works across typical smartphone camera models*. As a visual *quality benchmark*, Figure 6 plots the same scene with different qualities under flickering (Figure 7).

We also notice that *the quality worsens slightly as flickering frequency decreases* from 500 Hz to 100 Hz (CW-SSIM decreases by $\approx$0.1), as the image sensor has higher analog gain at lower flickering frequencies.[20]

Impact of RGB color distortion. When the RGB flickering is turned on, *the quality degradation is stronger if the RGB LED has the same average intensity with monochrome LED*. Besides, the color distortion makes an additional independent impact. The corresponding CIEDE2000 metric escalates up to 45, way beyond the *human-tolerable threshold 6*.[9] This implies *the scene is no longer considered acceptable by average viewers*.

Figure 5. Quality with autoexposure camera. Error bars show *std.* across OOK waveforms with different frequencies (100–500 Hz) and scenes.

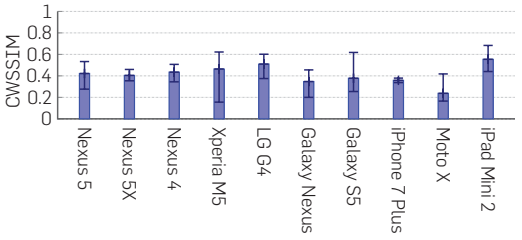


Figure 6. Image quality levels on a benchmark image.

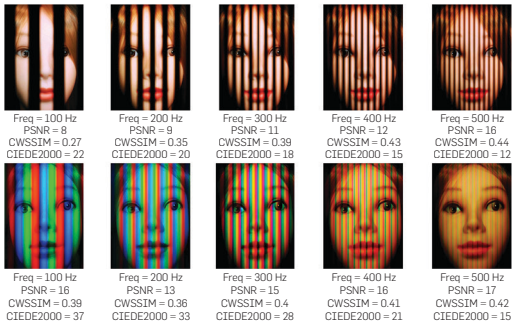
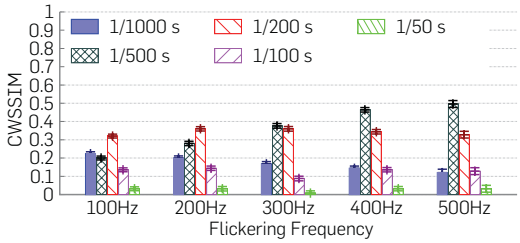


Figure 7. Quality with fix-exposure camera.



Two *bonus effects* from our RGB LED are observed: (i) The structural distortion from the stripes disrupts the camera's auto-focus function, often *making the captured scene extremely blur*. This is because under LiShield, contrast of bands no longer depends on focusing accuracy, which breaks the assumption of auto-focus mechanism. *(ii) The color bands also mislead the automatic white balance function across all five different scenes*, since the camera can no longer identify a clean region in the image to calibrate itself and thus hesitates.

Impact on dynamic scenes. To create a dynamic scene, we use the motor to rotate the smartphone, creating relative motion at three different speeds (45, 100, and 145 degrees/second). Our experiment shows the average CW-SSIM among all three speeds further decreases by 0.1, which indicates that *dynamic scene experiences worse quality under LiShield* due to motion blur. Moreover, if the exposure time is larger than 1/100 s, then overexposure and motion blurs together further reduce the quality (PSNR < 6, CW-SSIM < 0.1). Thus, *dynamic objects further decrease the adjustment range of exposure time and make manual exposure attack more ineffective*.

6.2. Effectiveness of user authorization

We developed an app (Section 5) that allows a user to capture critical frames on static scene protected by our RGB LED and then recover the scene following Section 3. The resulting image quality (PSNR = 25dB, CW-SSIM = 0.9, CIEDE2000 = 5) is comparable to the ideal setting when we disable LiShield's LED modulation (Figure 8 shows example frames extracted from a recorded video). In contrast, the attacker suffers intolerable image corruption (PSNR = 13dB, CW-SSIM = 0.56, CIEDE2000 = 34) by combining same number of randomly selected frames ("Illumination intensity randomization" section).

For the dynamic scene, we set f_{intra} = 1 kHz and f_{inter} = 300 Hz (Section 3.1). From Figure 9, we can see the authorized user has much higher quality (PSNR = 30dB, CW-SSIM = 0.98 in average) compared with attacker (PSNR = 10dB, CW-SSIM = 0.6 in average). This can be seen by resulting image frames in Figure 8, where attacker suffers from both intra-frame and inter-frame stripes. Thus *LiShield's authorization scheme is effective in unblocking specific users while maintaining protection against attackers*.

6.3. Effectiveness of barcode embedding

We first determine general optimal parameters for LiShield's barcode detector in Section 4, based on the following metrics. (i) False alarm rate. We run the detector on 200 images (random real-world scenes) and measure the probability that a barcode is detected from clean image. (ii) Detection rate. We embed monochrome barcodes with different f_1 from 400 Hz to 10 kHz with 200 Hz switching frequency. For each f_1, we embed three frequencies with Δf = 200 Hz interval and capture 300 images with these barcodes over a benchmark scene to obtain detection rate. Considering the trade-off between false alarm and detection, we choose T_b = 0.05 to bound the false alarm rate below 5%, while ensuring around 90% detection rate for monochrome barcode (Figure 10).

Using the above configuration, we found the detection rate for RGB barcode is close to 100% with or without manual exposure attack, while being slightly below

Figure 8. Frames observed by authorized users and attackers.

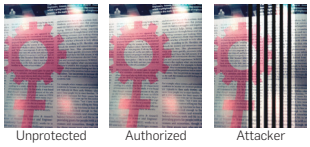


Figure 9. Video quality with and without authorization.

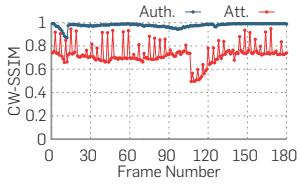
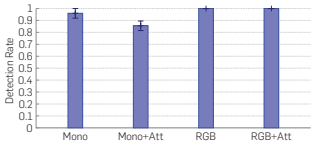


Figure 10. Detection rate of monochrome and RGB barcode design.



90% for monochrome barcodes if attacked. We conclude that *LiShield's barcode detector provides reliable detection, whereas RGB barcodes are more detectable and robust than monochrome ones*, thanks to extra redundancy provided by color channels.

6.4. Robustness against attacks

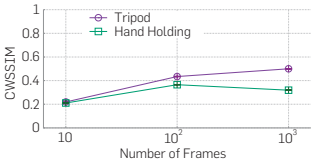
Manual exposure attack. One possible attack against LiShield is to manually set the exposure time t_e to smooth out the flickering patterns (Section 2.2). Our experiment shows that although the image quality first increases with t_e, it drops sharply as overexposure occurs. Therefore, *LiShield traps the attacker in either extremes by optimizing the waveform* (Section 2.1) and *thwarts any attempts through exposure time configuration*.

We also tested the effectiveness of randomization with auto-exposure (except for attacker). We set $f_1 = f_B$ = 200, 300, …, 600 Hz, Δf = 50Hz, and M = 2, 3, ..., 6 to scramble multiple frequencies. We found that the image degradation with scrambling is comparable with single frequency setup, thus *frequency randomization does not harm LiShield's protection*.

Multiframe attack. Figure 11 plots the recovered scene's quality under the multiframe attack. Here, we set t_e to be 1/500 s to avoid overexposure and then record a video in 30 fps. CW-SSIM remains low at 0.5 using 1000 frames, which means *the impact of stripes on structure of scene is still strong, making quality still unacceptable for professionals who spend such a great cost*. We also ask five volunteers to hold the smartphone as stable as they can on a table, and Figure 11 shows the quality is even lower, because it is impossible to completely avoid dithering with hands. Extending the recording duration increases disturbance and probability of being identified by the protected user, making it impractical for the attack to occur.

Image recovery processing attack. We evaluate the image quality after postprocessing with denoising or debanding (Section 5). The denoising methods fail to improve the quality significantly (CW-SSIM $\approx$ 0.3–0.4) as the disruption pattern of LiShield does not fit most known Gaussian noise model. The deformation removal methods (i.e., debanding and unstriping) do not help too much (CW-SSIM $\approx$ 0.4–0.5), since interpolation process cannot bring back the exact pixel values. The CIEDE2000 color metric also shows a low quality (around 35). Thus, *it is hard to fully remove LiShield's impact by simple image restoration*. More advanced computer vision techniques may provide better recovery, but even they will not recover the *exact original scene* since information is already lost at capture time.

Figure 11. Image quality by multiframe ensemble.



Impact of ambient light. We have also evaluated LiShield's performance under different types of ambient lights. We found the stripes are almost completely removed under direct sunlight due to its extremely high intensity. Flash light can increase the quality slightly thanks to its close distance to the scene, but the improvement is marginal and far from unprotected. In addition, we only found a marginal decrease of barcode detection rate in every case. Thus, we conclude that *LiShield is robust against most ambient lights*.

7. RELATED WORK

Camera recording of copyright screen-displayed videos (e.g., in a movie theater) accounts for 90% of pirated online content.[21] Since screen refresh rate is much higher than video frame rate, Kaleido[21] scrambles multiple frames within the frame periods to deter recording, while preserving viewing experience by taking advantage of human eyes' flicker fusion effects. Many patented technologies addressed the same issue. In contrast, the problem of automatic protection of private and passive physical space received little attention. Certain countries dictate that smartphone cameras must make shutter sound to disclose the photo capturing actions, yet this does not enforce the compliance, cannot block the photo distribution, and cannot automatically protect against video recording.

Certain optical signaling systems can remotely ban photography in concerts, theaters, and other capturing-sensitive sites. For example, BlindSpot[17] adopts a computer vision approach to locate retro-reflective camera lenses and pulses a strong light beam toward the camera to cause overexposure. Such approaches fail when multiple cameras coexist with arbitrary orientations.

Conventional visual privacy-protection systems have been replying on postcapture processing. Early efforts employed techniques like region-of-interest masking, blurring, mosaicking, etc.,[11] or re-encoding using encrypted scrambling seeds.[3] There also exists a vast body of work for hiding copyright marks and other information in digital images/videos (e.g.,[10]). LiShield's barcode protection is inspired by these schemes, but it aims to protect physical scenes before capturing.

8. CONCLUSION

Privacy protection in passive indoor environment has been an important but unsolved problem. In this paper, we propose LiShield, which uses smart-LEDs and specialized intensity waveforms to disrupt unauthorized cameras, while allowing authorized users to record high quality image and video. We implemented and evaluated LiShield under various representative indoor scenarios, which demonstrates its effectiveness and robustness. We consider LiShield as a first exploration of automatic visual privacy enforcement and expect that it can inspire more research along the same direction.

Acknowledgment

This project was partially supported by the NSF under Grant CNS-1506657, CNS-1518728, and CNS-1617321.

References

1. Barni, M. *Document and Image Compression*, CRC Press, Boca Raton, FL, 2006.
2. Denning, T., Dehlawi, Z., Kohno, T. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2014.
3. Dufaux, F., Ebrahimi, T. Scrambling for privacy protection in video surveillance systems. *IEEE Trans. Circuits Syst. Video Technol. 18*, (2008), 8.
4. Ferrari, F., Zimmerling, M., Thiele, L., Saukh, O. Efficient network flooding and time synchronization with glossy. In *Proceedings of the ACM/IEEE IPSN*, 2011.
5. Google. android.hardware.camera2.
6. Hoyle, R., Templeman, R., Armes, S., Anthony, D., Crandall, D., Kapadia, A. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the ACM UbiComp*, 2014.
7. Jiang, Y., Zhou, K., He, S. Human visual cortex responds to invisible chromatic flicker. *Nat. Neurosci. 10*, 5 (2007), 657–662.
8. Liang, C.K., Chang, L.W., Chen, H.H. Analysis and compensation of rolling shutter effect. *IEEE Trans. Image Processing 17*, 8 (2008), 1323–1330.
9. Luo, M.R., Cui G., Rigg, B. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Res. Appl. 26*, 5 (2001), 340–350.
10. Naor, M, Shamir, A. Visual cryptography. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1995.
11. Newton, E.M., Sweeney, L., Malin, B. Preserving Privacy by De-Identifying Face Images. *IEEE Trans. Knowl. Data Eng. 17*, 2 (2005), 232–243.
12. Rallapalli, S., Ganesan, A., Chintalapudi, K., Padmanabhan, V.N., Qiu, L. Enabling physical analytics in retail stores using smart glasses. In *Proceedings of the ACM MobiCom*, 2014.
13. Sampat, M.P., Wang, Z., Gupta, S., Bovik, A.C., Markey, M.K. Complex wavelet structural similarity: A new image similarity index. *IEEE Trans. Image Processing 18*, 11 (2009), 2385–2401.
14. Shao, L., Yan, R., Li, X., Liu, Y. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. *IEEE Trans. Cybern. 44*, 7 (2014), 1001–1013.
15. Social Pilot. 125 Amazing Social Media Statistics You Should Know, 2016.
16. Templeman, R., Rahman, Z., Crandall, D., Kapadia, A. PlaceRaider: Virtual theft in physical spaces with smartphones. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2013.
17. Truong, K.N., Patel, S.N., Summet, J.W., Abowd, G.D. Preventing camera recording by designing a capture resistant environment. In: *Proceedings of ACM International Conference on Ubiquitous Computing (UbiComp)* (2005), 73–86.
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing 13*, 4 (2004), 600–612
19. Winterhalter, W., Fleckenstein, F., Steder, B., Spinello, L., Burgard, W. Accurate indoor localization for rgb-d smartphones and tablets given 2D floor plans. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015.
20. Zhang, C., Zhang, X. LiTell: Robust indoor localization using unmodified light fixtures. In *Proceedings of the ACM MobiCom*, 2016.
21. Zhang, L., Bo, C., Hou, J., Li, X.-Y., Wang, Y., Liu, K., Liu, Y. Kaleido: You can watch it but cannot record it. In *Proceedings of the ACM MobiCom*, 2015.
22. Zhu, S., Zhang, C., Zhang, X. Automating visual privacy protection using a Smart LED. In *Proceedings of the ACM MobiCom*, 2017.

Shilin Zhu, Chi Zhang, and Xinyu Zhang
{shz338, c4zhang, xyzhang}@eng.ucsd.edu,
University of California San Diego, CA, USA.