Tracking and Disabling Smartphone Camera for Privacy

Qurban Memon¹, Khawla Al Shanqiti², AlYazia Al Falasi², Amna Al Jaberi² and Yasmeen Amer²

¹Associate Professor, EE Department, UAE University, Al Ain, 15551
²BSc student, EE Department, UAE University, Al Ain, 15551

Email: qurban.memon@uaeu.ac.ae

Abstract: With people's easy access to various forms of recent technologies, privacy has decreased immensely. One of the major privacy breaches nowadays is taking pictures and videos using smart phones without seeking permission of those whom it concerns. This work aims to target privacy in the current mobile environment. The main contribution of this work is to block the smart phone camera without damaging the smart phone or harming people around it. The approach is divided into stages: body area detection and then camera detection in the frame. The detection stage follows pointing of laser(s) controlled by a microcontroller. Tests are conducted on built system and results show performance error less than 1%. For Safety, the beams are devised to be harmless to the people, environment and the targeted smart phones.

1. Introduction

With the revolution of technology, cameras are becoming more and more popular. From a technical perspective, it is only just starting, as it is becoming more modern with different designs, different sizes mostly smaller, and better resolution, and the price is continuously decreasing. The camera popularity has seen a constant growth in the past twenty-five years, and seemingly, it will not stop any time soon.

Reading this as an individual that thrives for better technology and a modern future, it does not sound bad at all, in fact, this is considered great news, as camera has made a powerful impact, both in a smaller private scale, and definitely on a wider scale. However, like almost everything in the world, this quick spread of cameras around the world has taken a dark turn, by invading the privacy of many people.

While maintaining one’s privacy is more important to some people, it is globally known to be a dearly valued concept to all people. It is something to be protected, in some cases, by all possible means. And with the current environment that is overflowing with various types of technologies, people are given the accessibility to many forms of personal or public information, as well as many forms and means to intrude someone else’s private life, which resulted in a huge decrease in privacy in general. One of the major privacy breaches is taking pictures and videos in a photography-prohibited environment or without seeking permission of those whom it concerns.

In public places, like parks, sidewalks, and malls, one is free to take pictures as one pleases. However, some places, like sensitive government buildings (military bases, nuclear facilities), for example, can deem a threat to national security. In addition, private parties have very limited rights to detain you against your will but are considered situations where privacy is to be protected at any cost, especially as a Middle Eastern culture and specifically in UAE, with most of the population using...
smartphones, the restrictions on privacy are very sensitive regarding unauthorized photography in weddings and private gatherings. All that leads to important questions that pass through most minds: How can that be avoided? Is it possible to preserve the privacy that is considered one of the basic rights of each human being?

Maintaining privacy is the main objective of this work. The idea focuses on tracking and disabling smartphone camera, while keeping one’s privacy. This provides a technical solution that does not require asking people verbally not to take pictures while maintaining their well-being and the safety of their properties. This can be done using a fast image-processing algorithm to detect the smart camera specifications such as its shape and size, and then disable it using a jamming source, without having any impact on other functionalities and applications of the smart phone device. Jamming a device or network has been addressed in literature at length like [1], where authors report that off-the-shelf smartphones can be used to selectively jam Wi-Fi networks in the 2.4 and 5 GHz bands using arbitrary waveforms stored in IQ sample buffers. However, this jamming ability creates a risk of misuse initiated by a malicious attacker to jam Wi-Fi networks in a selected area. This technique exposes capability of a smartphone to jam network(s), but does not stop any one to use camera to take unsolicited pictures. For tracking, the reader is referred to [2-3].

A number of solutions have been attempted [4-5], but they were either partial or imperfect to handle challenges during smartphone camera jamming, as addressed in this work. More recent approaches [6-7] report disabling camera with some success. In [6], the authors propose an approach, which is based on Near Infrared labelling. The labelling is dependent on camera and people in surroundings are unaware of this label, when an object wears it. Further, the authors propose a policy label recognition and enforcement system implemented on Android. The system then recognizes these labels and then enforce privacy policies using a Gaussian Blur, for example. Similarly, the author in [7] use a smart LED on-off lighting beyond human eye sensitivity to generate a flickering pattern to interfere with camera sensors and create a vertical striped pattern effect in the photo taken by digital smartphone-grade camera in an indoor environment. However, the authors emphasize use of multiple LEDs for entire area to enforce privacy, but synchronization between multiple LEDs is still a problem to be fixed. In the next section, a proposed approach is presented followed by experimental results in section 3, and conclusions are in section 4.

2. Proposed approach

This section discusses details of the design: hardware and software. It addresses component specifications, and criteria for each part, and the stages needed to reach final design target.

2.1 Component Specifications

First, the details of the components that have been chosen to achieve the targeted goal of this work and the criteria of selecting these components are discussed below:

Camera: The camera is needed to continuously capture the frames that are processed for smartphone camera detection. It is connected to processor of the system. Its specifications are: C270 HD Webcam, video capture up to 1280 x 720 pixels, video capture (software enhanced) up to 5 Megapixels resolution, interface of Hi-speed USB 2.0 and 1 Mbps upload speed or higher. Required resolution was found considering the maximum possible height of a person (2m) and the area around him (2m$^2$) with respect to an area smaller than that of the camera lens (1mm$^2$), using the following equation:

$$\text{Resolution} = \frac{\text{Object Size}}{\text{Size of the detail to be inspected}} = \frac{2m^2}{1mm^2} = 2000\, \text{px}$$

Microcontroller: In the prototype, the microcontroller is working as the interface between lasers matrix and the main processor, which is the laptop. After capturing the frame and detecting the coordinates of the smartphone cameras using image processing, the corresponding lasers of the targeted coordinates should be ‘ON’, and this can be achieved using the microcontroller as the interface.
**Laser Matrix:** Lasers are the jamming source that point toward the camera lens to jam it temporarily, safe to the surroundings and without damaging the camera lens. In the prototype, a 2x1 matrix is used, which means that two coordinates will be possible to target and jam. Safety of this laser was set as number one criterion. Based on this consideration, the lasers of class 2 were used, since the total power output of class 2 laser should not exceed 1mW [8], which is the maximum allowable power output in the GCC region. The wavelength of the required lasers vary between Red (620 to 780 nm) and Green (492 to 577 nm).

The lasers with the wavelengths mentioned above are the lasers that were obtained and used in the testing stage. The specifications and criteria for the final decision is based on the success rate of blurring photos, safety, the path visibility and the cost. For positioning, the lasers are critical with the frame they are fixed within. That is, the lasers should be perpendicular to the surface of the frame, or, in other words, as parallel to the ground as possible. That will eliminate the dependency on the depth that might result when the detected smartphone camera is found in different distances from the jamming source. A seemingly small deviation in the laser angle with the surface will be very noticeable in large distances from the laser source, which will make its coordinates in the frame taken from the webcam vary with depths.

**Processing:** A laptop with Intel core i7 processor running MATLAB is used to get the images from the camera, process these images to determine the targeted coordinates where the smartphone cameras are found and turn the lasers ON or OFF through the microcontroller interface.

**Prototype Architecture:** Based on these components, the prototype is envisaged as shown in the Figure 1. The processor receives the frame, and processes it. In prototype, the lasers matrix consists of two red lasers interfaced with the processor using an Arduino Uno microcontroller board. The webcam will provide the frame to be processed, and is interfaced with the processor via a USB port. This is illustrated in Figure 2.

![Figure 1. Prototype Architecture](image1)

![Figure 2. Connections to lasers through PCB](image2)

**Database Design:** For testing, database tables were developed for the smartphones that the device is capable to jam. Moreover, some ratios and measurements are made as conditions in image processing routines to determine if the targeted circles are smartphone camera lenses or not. The Table 1 shows the published database for cameras’ area on the Internet by manufacturing companies, while Table 2 shows calculated ratios of these areas with those of box areas corresponding to smartphones used for testing.

**Table 1. Ratio of Camera Area**

| Phone   | Height (mm) | Width (mm) | Camera Area                        | Area ratio |
|---------|-------------|------------|------------------------------------|------------|
| IPhone7 plus | 158.2      | 77.9       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 440.135    |
| IPhone7  | 138.3       | 67.1       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 331.43     |
| IPhone6s plus | 158.2      | 77.9       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 440.135    |
| IPhone 6s | 138.3       | 67.1       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 331.426    |
| IPhone SE | 123.8       | 58.6       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 277.93     |
| IPhone 4  | 115.2       | 58.6       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 241.097    |
| IPhone 3  | 115.5       | 62.1       | 28mm, f/1.8, OIS & 56mm, f/2.8    | 256.162    |
| Samsung Galaxy S3 | 136.6   | 70.7       | 8 MP, f/2.6, 1.9 MP, f/2.6         | 311.5      |
In this section, logical steps related to mainly image processing are explained.

**Calibration:** Calibration is the first step for the device to determine laser coordinates within the frame, to be later compared with the targeted circle coordinates (of smartphone cameras lenses). The main step used is the red objects detection method, since lasers are red. Before starting using the device, lasers will be “ON” one by one on a non-red background, where they will be detected and coordinates are assigned to variables, which are later turned into ranges to consider any resulting uncertainties in the main detection code. The calibration code makes use of the fact that red lasers are used and that a code for red objects detection can be developed. The code written creates first a camera, Arduino objects and configure pins D6 and D7 as digital outputs. Then, the laser is turned on and the camera starts taking a short video of 50 frames as set in the outer loop. The outer loop will detect the red laser while the inner will bound it in a rectangle and show its center coordinates. Then, these coordinates are assigned to variables, which then are turned into ranges. This is repeated for the second laser after the first is turned off. All variables in this code will remain after the code terminates except the camera object.

### Table 2. Area to box ratios, example of iPhone modules

| Type        | Box area | Camera lens area | Area ratio | Box area | Camera lens area | Areas ratio | Areas ratio average |
|-------------|----------|------------------|------------|----------|------------------|-------------|---------------------|
| iPhone 3    | (405)^2  | Π (9.8)^2        | 543.9      | (416)^2  | Π (9.9)^2        | 559.7       | 551.84              |
| iPhone 4    | (389)^2  | Π (8.3)^2        | 695.8      | (363)^2  | Π (7.8)^2        | 680.6       | 688.24              |
| iPhone 5    | (375)^2  | Π (7.9)^2        | 702.9      | (411)^2  | Π (9.8)^2        | 559.8       | 620.78              |
| iPhone 6    | (430)^2  | Π (9.9)^2        | 592.1      | (420)^2  | Π (10.2)^2       | 538.6       | 565.37              |
| iPhone 7    | (405)^2  | Π (9.2)^2        | 619.5      | (325)^2  | Π (6.9)^2        | 700.1       | 659.81              |

2.2 **System Development**

In this section, logical steps related to mainly image processing are explained.

 Calibration: Calibration is the first step for the device to determine laser coordinates within the frame, to be later compared with the targeted circle coordinates (of smartphone cameras lenses). The main step used is the red objects detection method, since lasers are red. Before starting using the device, lasers will be “ON” one by one on a non-red background, where they will be detected and coordinates are assigned to variables, which are later turned into ranges to consider any resulting uncertainties in the main detection code. The calibration code makes use of the fact that red lasers are used and that a code for red objects detection can be developed. The code written creates first a camera, Arduino objects and configure pins D6 and D7 as digital outputs. Then, the laser is turned on and the camera starts taking a short video of 50 frames as set in the outer loop. The outer loop will detect the red laser while the inner will bound it in a rectangle and show its center coordinates. Then, these coordinates are assigned to variables, which then are turned into ranges. This is repeated for the second laser after the first is turned off. All variables in this code will remain after the code terminates except the camera object.
Figure 3. Flowchart of the system

**Detection Steps:** After capturing the frame, three things are detected to finalize detection of smartphone camera lens: (1) face detection, which is the process of identifying one or more people in images or videos. After face is detected, the bounding box is expanded enough to cover all the upper body; (2) eyes are detected using ‘vision.CascadeObjectDetector’ function based on Viola-Jones Classifier [9]; (3) all of the circles in the frame are detected using Hough transform technique [10]. This detects bright sensitivity circles and dark sensitivity circles. The bright circles are saved in [cen1, rad1], where cen1 is an array for the centers of the circles and rad1 is an array for the radii of the circles respectively. So if X circles were detected we will have X rows in cen1 array and X rows in the radius array for the radii. This follows eliminating circles of eyes. The first elimination step is to eliminate any circles outside the box, which is the bounding box of the upper body, because smartphones are held by persons, and any circle that is not around a person, will not be a smartphone camera. The flowchart in Figure 3 summarizes and combines the steps that are just explained. As shown, getting the frame is followed by detection phase, then elimination process, and finally controlling lasers to be “ON” or “OFF”. Continuous tracking requires repetitive detection, and that calibration is not repeated as it is done during set stage explained before.

3. **Experimental Results**

For calibration, a code was tested separately. The calibration test was done in several lighting situations with best results in normal to dimmed lighting. This was acceptable since it is a step that is done only once for the device. In order to test the approach, a code was written in Matlab. A number
of trials were run on different persons. Table 3 shows circle detection results for 10 consecutive tests on frames. The last testing process included integration of microcontroller and laser matrix into the main detection code. The outputs of detection code are the coordinates of the smartphone camera. After that step, these values are compared with coordinates of the lasers in the laser matrix. The laser is turned ON, if these values match. This test was done several times with outputs similar to detection outputs shown in Table 3.

An elimination step is done for eye circles to ensure safety, even though lasers specifications are safe. The last step is for remaining circles inside the box, where areas are calculated and then ratio of these areas to the ones in box (that they lie in) are calculated. These ratios are compared with ratios in Table 2 to know whether it is a smartphone camera lens or not.

From the results of the tests, that the percentage of error and the duration of the detection process were determined to be an average of 0.74% and two seconds respectively. For the percentage of error, it is important to note that this error was of type false positive. Never was it a false negative where the smartphone camera was there and left undetected. The percentage is very acceptable. As for the duration of detection, while it is considered a delay, there is always room for improvement in coding process. Currently, we are aiming for a one-second duration, but since a person does not move much while taking a picture, a two-second delay is still acceptable to some degree, especially for prototyping.

**Table 3. Percentage errors**

| Test No. | Number of detected circles before filtering | Number of circles besides the camera left after filtering | % Error |
|----------|---------------------------------------------|-------------------------------------------------------|---------|
| 1        | 256                                         | 4                                                     | 1.56    |
| 2        | 174                                         | 1                                                     | 0.57    |
| 3        | 287                                         | 3                                                     | 1.05    |
| 4        | 429                                         | 5                                                     | 1.16    |
| 5        | 99                                          | 0                                                     | 0.00    |
| 6        | 126                                         | 2                                                     | 1.58    |
| 7        | 328                                         | 0                                                     | 0.00    |
| 8        | 189                                         | 1                                                     | 0.53    |
| 9        | 64                                          | 0                                                     | 0.00    |
| 10       | 211                                         | 2                                                     | 0.95    |

Average % Error 0.74

4. Conclusions
The main objective in this work was to prevent unauthorized use of smartphone cameras in places where privacy is prioritized. The approach requires image processing steps that include face detection, eye detection, and circle detection along with corresponding coordinates. The code works with microcontroller to point laser(s) beam(s) at the detected camera circle(s), with error testing results of less than 1%.

The proposed solution is improvable and can be enhanced from several aspects. The duration of this project was a limitation to do some improvements. Following are the some points to enhance: (1) discrete coordinates: Lasers in the matrix can point to definite coordinates, so a smartphone camera that lies in a coordinate that does not belong to the lasers coordinates will be detected but not jammed. Having more lasers and getting them closer will reduce the blind spots. Since the laser coordinates will always be definite and discrete, so a fast rotating motor can be used as an alternative to use of fixed matrix; (2) maximum distance: The maximum distance that can be covered is actually a variable that depends on the selected camera resolution and range of radii for the detected circles. A higher resolution and a wider range of radii can increase the maximum distance but it will increase the cost and more circles will be detected. This is expected to lead to more challenges in the detection methodology to enhance precision; (3) the detection process needs a well-lighted environment to work best. For dark places, like theaters, the suggested solution is to add a “Dark operation mode”, where detecting camera runs on a night vision mode.
References

[1]. M. Schulz, F. Gringoli, D. Steinmetzer, M. Koch, and M. Hollick, “Massive reactive smartphone-based jamming using arbitrary waveforms and adaptive power control,” ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 111-121, doi: 10.1145/3098243.3098253

[2]. Q. Memon, “Smarter Healthcare Collaborative Network,” Building Next-Generation Converged Networks: Theory and Practice, CRC Press, 2013.

[3]. Q. Memon, S Khoja, “RFID–based Patient Tracking for Regional Collaborative Healthcare,” International Journal of Computer Applications in Technology, 45 (4), pp. 231-244, 2012

[4]. The Pirate Eye Anti-Piracy Solution. (n.d.). Retrieved from http://www.pirateeye.com/pirateeye/

[5]. Kate Greene, (June 22, 2006). MIT Technology review, a prototype device seeks out cameras and blocks them from taking pictures and video. Retriever From https://www.technologyreview.com/s/405968/lights-camera-jamming/

[6]. D. Li, D. Guo, W. Han, H. Chen, C. Cao and X. S. Wang, “Camera-Recognizable and Human-Invisible Labelling for Privacy Protection,” International Conference on Mobile Ad-Hoc and Sensor Networks, Hefei, 2016, pp. 365-369., doi: 10.1109/MSN.2016.066

[7]. J. Hsu, “LiShield Can Block Smartphone Cameras for Privacy's Sake,” IEEE Spectrum, Oct. 2017.

[8]. T. Wheatley, “Laser pointer prohibition: improving safety or driving misclassification,” International Laser Safety Conference, 2013, pp. 48-54

[9]. Viola, P., Jones, N., “Robust Real Time Object Detection,” International Journal of Computer Vision, pp. 137-154, 2004.

[10]. Hough, P.V.C. Method and means for recognizing complex patterns, U.S. Patent 3,069,654, 1962