SMARTPHONE BASED VISIBLE LIGHT COMMUNICATION BEACON SYSTEM FOR INTERNET OF THINGS

Marek Novák
Doctoral Programme (2017), DREL FEEC BUT
E-mail: xnovak0m@vutbr.cz

Supervised by: Peter Barcík
E-mail: barcik@feec.vutbr.cz

Abstract: The article describes an application of Visible Light Communication (VLC) for Internet of Things (IoT) nodes authentication and metadata exchange. A practical use-case of presented communication system is provided and benefits of using such technique is explained. A smartphone camera based platform and advanced image processing is used for data reception and as a proof of concept implementation of presented communication technique. The future direction of the research is provided.

Keywords: VLC, Internet of Things, OCI, Image Processing, GPU, AR, Android, OpenGL, GLSL

1 INTRODUCTION

One of advantages of VLC besides potentially high bandwidth is the possibility to leverage existing infrastructure. In almost any today's device requiring human interaction there is at least one status light emitting diode (LED) or an LED indicating power state of the device. This paper proposes to multiplex their purpose with VLC to allow an unidirectional communication for distribution of the device related metadata. As receiver of these metadata, which may consist of passphrase in case of a WiFi access point, a camera mounted on a smartphone will be used.

Cameras are present practically on any contemporary smartphone. Therefore, it is very easy to make the VLC technology and its application proposed herein accesible to a wide range of users.

Further in this paper, related work will be presented, consequently the communication chain of the proposed system will be explained, followed by description of reception chain with focus on image processing of data from the camera. In the last section, experimental results and direction of the research is announced.

2 STATE OF ART

The usage of so-called human controlled LED-Camera channel was already announced in [1]. The author combines radio frequency (RF) and VLC channels for authentication. VLC is used as an out-of-band (OOB) channel in the authentication process. This paper proposes a different application consisting of the device metadata broadcasting, which represents a generalization of [1].

A patent proposing a beacon system based on VLC was already published in [2]. This paper proposes however using status or power LEDs as transmitters. This differs from [2], where authors propose leveraging illumination sources instead. Furthermore in the presented system, the broadcasted data are not intended for indoor positioning as in [2].

A design of VLC beacon system for augmented reality (AR) was published in [3]. Authors create a dedicated CMOS sensor and a field programmable gate array (FPGA) digital signal processing (DSP) system to create an augmented reality vision. Our paper, however, targets cameras mounted in contemporary smarphones and on-device GPU for image processing rather than providing a custom hardware solution.

3 COMMUNICATION CHAIN

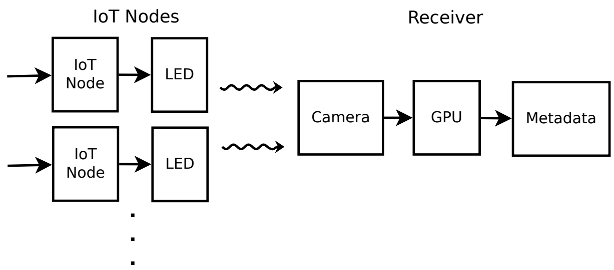

Figure 1: Communication chain diagram

As shows Fig. 1, the communication system consists of single or multiple IoT nodes (such as temperature sensors, humidity sensors, WiFi access points etc.) and a single or multiple smart devices with a camera and graphical processing unit (GPU). The number of IoT nodes being captured by a receiver is given by the number of devices being visible by the device camera.

The communication task can be split in two parts - first part is the detection of devices in the picture (in the signal from the camera) and the second consists in demodulation and decoding of the transmitted data.

The most critical part of the chain is the receiver camera. A common frame rate is 30 fps, which in case of a single red or green LED with on-off keying (OOK) provides only 15 bps. In more recent smartphones, there are however cameras capable to shoot slow-motion video with the frame rate of up to 240 fps with the resolution of 720 p (e.g. Samsung Galaxy S7 or Apple iPhone 6S). This would enable a maximum bit rate of 120 bps.

4 RECEIVER IMAGE PROCESSING

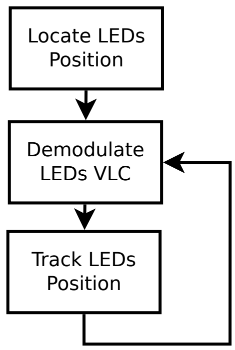

Figure 2: Image processing algorithm high level flowchart

Fig. 2 represents actions which need to be taken for a successful demodulation. First, the LED or LEDs need to be located in the picture. If the camera is not completely stationary (user holds the smartphone simply in his hand), this needs to be repeated periodically. Several different approaches can be taken based either on correlation or high pass filtering or both [1].

4.1 LOCATION OF LED POSITION

As already mentioned, there are two methods of LED position detection.

The first one is based on correlation of the image with a known image pattern. This pattern represents the transmitting LED. A common method to calculate the correlation is using the convolution and the fast Fourier transform (FFT) [7].

The second method analyses the image as a part of sequence of images, rather than analysing it isolated. The advantage of this method is that the pattern representing the transmitting LED does not have to be known and the number of mathematical operations required to detect the position of the LED is lower. This approach however requires the transmitter and the LED to be static or at least moving slowly compared to the FPS. When this is verified, substracting two consecutive images pixel by pixel gives either a black image or a black image with bright areas. In first case, the LED has transmitted on both images high logic value. In the other case, the value has changed from low to high or from high to low. As the baudrate is chosen to be as close to the half of FPS as possible only up to 3 consecutive images are needed to obtain the bright areas. LEDs are located in the center of gravity of those bright spots in the resulting image. This method therefore requires only N operations of substraction to be done in order to process the whole image, where N is the number of pixels in the image.

If the transmitter and the receiver were expected not to be mutually static, the first method would be chosen. In application proposed by this paper, the scene is supposed to be quasi-static, so the second method can be preferably used or herein proposed image processing technique can be adopted (Listing 3) to compensate the effects of non-static scene.

4.2 TRACKING LED POSITION

Once the transmitting LEDs are detected, their position is stored in an inventory. If the scene is not completely static, then their position needs to be tracked and LEDs detected in a new image needs to be matched to those detected in the previous images. This is especially needed in a scenario with multiple transmitting LEDs, otherwise the transmitted symbols' streams could be mixed.

All LEDs present in the camera field of view are tracked, so the task of the application would be to determine, which LEDs are transmitting metadata and which ones are just regular LEDs (and ignore them). This only applies when the correlation-based method is used for LEDs' detection. In case method based on substraction of subsequent frames is used (temporal high pass filtering), LEDs which are not blinking are not detected and automatically ignored.

4.3 DEMODULATION OF OOK SYMBOLS

As already announced, the demodulation is not the most complicated part of the reception chain. When the position of the LED is known across all images in the sequence of images creating the video stream, then the brightness of the pixel representing a transmitting LED is compared to a brightness threshold. The result of the comparison is the detected symbol.

5 EXPERIMENTAL RESULTS

5.1 SOFTWARE IMPLEMENTATION

The image processing algorithm used for VLC data reception was developed using OpenGL fragment shader, which is programmed in OpenGL Shader Language (GLSL). The processing is done in two steps - first part consists of edges detection in a frame and the second part consists of motion detection.

The edges detection is done using first-order derivative of the image in both axes direction of the image. The value of the derivative function is compared with a fixed threshold and based on the result of this comparison, a white or a black pixel is generated in the resulting edge detection image mask. The GLSL code is provided in Listing 1. The $dFdx()$ and $dFdy()$ functions provide respectively value of derivative in direction of axis x and axis y. The function $length()$ provides length of a vector and $step()$ function is used for comparison of selected threshold (0.02) with the length of the vector containing values of derivative. This equation is evaluated for each pixel of the image in parallel using GPU. The *edges* vector contains red, green and blue components for the given pixel being evaluated.

Listing 1: Edge detection frame shader source code

```
vec3 edges = vec3(step(0.02, length(vec2(dFdx(gray), dFdy(gray)))));
```

The motion detection is done by substracting the value of luminosity component of the current frame from the value kept from the previous frame. If the scene changes, the value is non-zero. Listing 2 provides equivalent GLSL source code. *grayL* is the value of luminosity from the previous frame and *gray* is the value from the current frame. The vector *motion* contains red, green and blue components for a pixel forming the resulting image.

Listing 2: Motion detection frame shader source code

```
vec3 motion = vec3(abs(grayL-gray));
```

By substracting the pixels from edge detection from the pixels of motion detection, only the areas containing blinking LEDs and their reflections remain white. Listing 3 contains the source code.

Listing 3: VLC LEDs detection shader source code

```
vec3 vlc = max(motion - edges, 0.0);
```

The function *max*() clamps the negative values to 0 (black color) resulting in desired effect.

The source codes of a proof-of-concept application which uses this image processing algorithm can be found at [9].

5.2 Image Processing Performance

Fig. 3 contains a screen captured by Nexus 5X smartphone before and after application of frame shader. After localisation of the bright area in the filtered frame, the value of data being transmitted can be read from the original frame using the same coordinates. For more robust operation a prediction of future LED position in the frame can be done and any parasitic bright areas can be therefore ignored. This makes the whole image processing also more feasible for higher frame rates. LED position prediction and tracking is however not provided in this paper.

6 CONCLUSION

The paper addresses a specific area of VLC and proposes to integrate VLC in IoT nodes and other devices containing at least a single status LED. The emphasis is put on generating as few requirements as possible to the transmitting IoT node. Architecture of a proof-of-concept implementation is provided. LED Tracking algorithm development is to be done.

Data received by the receiver (a smartphone) can be further used in systems implementing the augmented reality (AR) [3]. Real world transmitting IoT nodes can be for example temperature, humidity or pressure sensors, smart home devices such as smart TV or WiFi access points etc. The list of IoT nodes does not exclude light/illumination sources, but these are not mainly targeted by this paper.

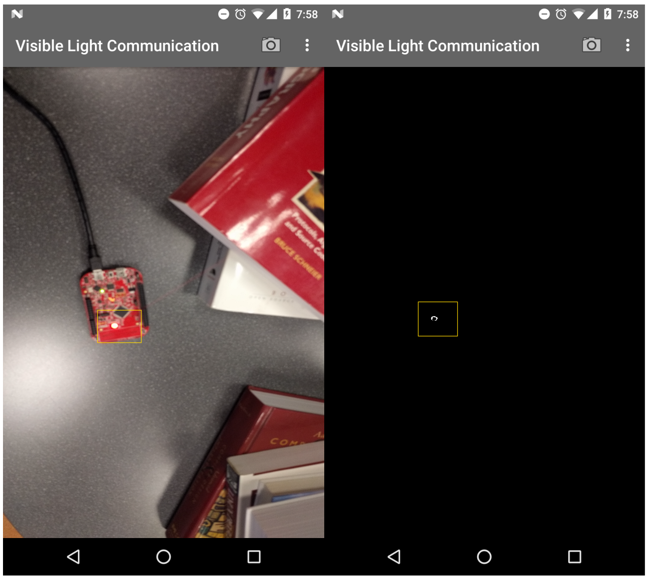

Figure 3: Android application video stream before and after application of detection algorithm

ACKNOWLEDGMENT

Research described in this paper was financed by Czech Ministry of Education in frame of National Sustainability Program under grant LO1401. For research, infrastructure of the SIX Center was used.

Herein described research was also financed by project FEKT-S-17-4707 ("Návrhy nových funkčních bloků a algoritmů pro moderní komunikační systémy").

REFERENCES

[1] SHANGYUAN, G.; Device Pairing Using Visible Light Communications, p. 56, 2014.

[2] GANICK, A.; RYAN, D.; Independent beacon based light position system. Patent No. US9287976 B2, 2016.

[3] OIKE, Y; IKEDA, M.; ASADA, K.; A Smart Image Sensor With High-speed Feeble ID-Beacon Detection for Augmented Reality System, p. 4, 2003.

[4] FERNANDIZ-LAHUERTA, J. et al.; A reliable asynchronous protocol for VLC communications based on the rolling shutter effect, p. 6, 2015.

[5] HUI-YU, L. et al.; RollingLight: Enabling Line-of-Sight Light-to-Camera Communications, p. 14, 2015.

[6] PENGFEI, H. et al.; ColorBars: Increasing Data Rate of LED-to-Camera Communication using Color Shift Keying, p. 27, 2015.

[7] CHEVALIER, L.; CALLOCH, S.; HILD, F.; Digital Image Correlation used to Analyze the Multiaxial Behavior of Rubber-like Materials, p. 38, 2001.

[8] TAKAI, I. et al.; Optical Vehicle-to-Vehicle Communication System Using LED Transmitter and Camera Receiver, p. 14, 2014.

[9] NOVAK, M.; VLC Android Application Source Code, available at `https://github.com/MarekNovakNXP/camera`