

Simulation of Rolling Shutter Acquisition in Optical Camera Communications

V. Matus, V. Guerra, C. Jurado-Verdu, J. Rabadan, and R. Perez-Jimenez
IDeTIC-ULPGC, PCT Tafira, 35017, Las Palmas de Gran Canaria, Spain.
Email: {vmatus, vguerra, cjurado, jrabadan, rperez}@idetic.eu

Abstract—The rolling shutter (RS) acquisition mechanism performed by the majority of the digital cameras included in portable devices has been exploited to make receivers for visible light communications (VLC). They are able to decode signals of frequencies higher than the frame rate of the camera, which is a limitation when using global shutter sensors. Based on experience with an experimental optical camera communications system, a simulation of rolling shutter receiver for OCC has been developed. This simulator uses a color-multiplexed on-off keying (OOK) modulation, along with a radiance pattern for the LED source, to generate transmitted signals. Then, a simplified camera optics model translates them into received signals, to which the RS simulation is applied, delivering synthetic camera frames.

Index Terms—Optical Camera Communications, Rolling Shutter

I. INTRODUCTION

The opportunity of exploiting mobile camera technology as receivers for Visible Light Communications (VLC) has been explored in the recent years since the publication of seminal paper [1]. The principle is to take frames with the camera sensor and decode different states of modulated light sources after processing these frames.

The video-recording capabilities of cameras, yielding up to hundreds of frames per second (fps), along with the ability to run heavy image processing algorithms, have enabled researchers to implement links between low-speed VLC lamps and digital cameras. The decoding stages are purely based on software, making Optical Camera Communications (OCC) a technology that can take advantage of already deployed end-user devices such as Light Emitting Diodes (LED) and smartphones.

In this stage of OCC technology, simulating systems is a crucial step for designing and testing new devices and algorithms. CamComSim by A. Duque *et al.* [2] gives a tool for the estimation of bit and packet performance of OCC links, allowing the test of higher layer protocols. That work was experimentally validated with a test bed in which the alignment between transmitter and receiver is assumed perfect. In CamComSim, the distance between transmitter and receiver can be up to 40 cm. It allows to use only a circular shape for the LED (the diameter can be changed). In the level of the optical phenomena, [3] proposes a geometrical model for studying non-aligned LED and camera positions, along

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 764461

with the use of a Point Spread Function (PSF) to model the blurriness of the picture by using parameters obtained from camera radiometric response database, obtaining simulated images of multiple LED sources.

In this paper a simulator of the rolling shutter mechanism of an OCC setting it presented. The work comprises computational models for transmitter radiance, channel propagation, camera optics, and the process of RS. The main objective is to reproduce the line-wise scanning of VLC sources and to output synthesized image frames. This simulator is a useful tool for testing and validating OCC algorithms and hardware since it allows to set up a variety of different scenarios while preserving the signal synchronization between the source and the receiver.

The next sections of this manuscript are distributed as follows. Section II shows the basic principles of the communications chain that is going to be simulated. Section III explains the way each block is simulated and what the computation delivers. Finally, Section IV exposes the conclusions of this work.

II. SYSTEM DESCRIPTION

A. Transmitter

The transmitters consist of red-green-blue (RGB)-LED light sources with arbitrary shape and radiation pattern, emitting OOK-encoded light signals independently over the three color channels. In the literature, the radiation pattern of an LED is assumed to be Lambertian [4], but it is important to notice that the commercial LED-based lamps usually integrate a variety of optical pieces, such as reflectors and lenses, that modify this pattern consequently. Nonetheless, it is possible to obtain radiation measurements of lighting devices from manufacturers under the Illuminating Engineering Society (IES) or EULUMDAT formats. These files provide a numerical definition of the lamps. Although in this work it has been assumed that the radiation pattern is uniform within the emitter area, generally it is defined as $I(x, y, \theta, \phi)$.

The shape of the emitter is important in OCC, since its projection over the sensor of the receiver determines the pixels that is going to illuminate. In this work, the shape of LED transmitters is assumed a flat area, and it has been defined by the perimeter line of it Γ_{Tx} , which is a tuple of vertices. Typical VLC systems consider the emitters as point sources since their areas are much smaller than the projected area of the receiver's Field Of View (FOV) at the emitter's plane.

Nevertheless, in OCC, since each pixel presents equivalent FOV in the range of several milliradians, the situation exposed above does not occur.

B. Optics

Position and orientation of transmitters (Tx) and receivers (Rx) are modeled using linear algebra in order to create a realistic scene. The distance between Tx and Rx is assumed long enough to perform the differential solid angle approximation ($d\Omega \approx A_{lens}/d^2$).

The optical system has been modeled using Ray Transfer Matrix (RTM) equations. The main limitation of this approach is that its accuracy is severely diminished for non-paraxial rays (incident angles above 30°). RTM is based on 2×2 matrices which model how optical rays travel through the optical system. Since RTM assumes that $\sin \theta \approx \theta$, and the refraction (or reflection) plane is conserved ($\phi_{in} = \phi_{out}$), the output radial deviation y_o and angle θ_o can be expressed as Eq. 1.

$$\begin{pmatrix} y_o \\ \theta_o \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} y_i \\ \theta_i \end{pmatrix} \quad (1)$$

This ABCD matrix is the product of all the individual matrices which compose the optical system (Eq. 2).

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \prod_i \begin{pmatrix} A & B \\ C & D \end{pmatrix}_i \quad (2)$$

C. Receivers

Cameras are essentially an imaging sensor attached to an image-forming array of lenses. It is assumed that the lens array of the camera can be reduced to one equivalent lens. Furthermore, all the incident power is projected over the sensor without losses. Equation 3 describes the incident power as follows:

$$P_{Rx} = P_{Tx} \frac{m+1}{2\pi} \cos \theta \cdot \frac{A_{lens} \cos \Psi}{d^2}, \quad (3)$$

where P_{Tx} is the transmitted power, m is the degeneration factor of the Lambertian pattern (directivity), θ is the emission angle, A_{lens} is the main lens area, Ψ is the incident angle, and d the link range.

The projected area in pixels N_{px} of a distant source depends on the receiver's FOV and the sensor resolution. Several works treat this mathematically [5], but in this work, a generic simulation framework has been defined, and the projected area has been numerically calculated using Eq. 1 constrained to the optical axis ($y_i = 0$). The resulting projection of Γ_{Tx} has been binarized, defining a mask of ones within the projected area and zeroes otherwise. Due to the conservation of the *etendue*, the optical power at each pixel is:

$$P_{px} = \frac{P_{Rx}}{N_{px}}. \quad (4)$$

The Point Spread Function (PSF) of an optical system is its spatial impulse response, which describes how energy is spread by the system. If no losses are taken into account,

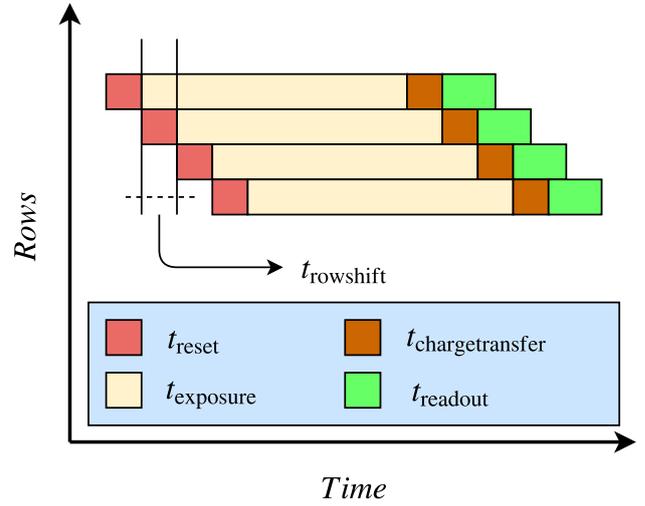


Fig. 1. Illustration of row-by-row timings in RS acquisition mode.

the energy of the PSF must be unitary. From the locality principle, it can be assumed that the PSF of each point of the surface is the same. Therefore, the resulting energy distribution $I_{Rx}(x, y)$ is the convolution of the Mask $M(x, y)$ with the PSF $h(x, y)$, weighed by the pixel power (Eq. 5).

$$I_{Rx}(x, y) = P_{px} \int_X \int_Y M(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (5)$$

Finally, the input power is filtered by the Bayer filter array, separating the image spectrum into R, G, and B components. In addition, the optoelectrical conversion is carried out generating a photo-current dependent on the input spectrum. For each pixel, Eq. 6 is applied:

$$i(x, y, c) = I_{Rx}(x, y) \int_{\lambda} \bar{S}(x, y, \lambda) F_c(\lambda) R(\lambda) d\lambda, \quad (6)$$

where $i(x, y, c)$ is the output pixel current at the c -th channel, $\bar{S}(x, y, \lambda)$ is the pixel's energy-normalized input spectrum, $F_c(\lambda)$ is the c -th channel filter response, and $R(\lambda)$ is the substrate's responsivity (typically Silicon).

The majority of Complementary Metal-Oxide Semiconductor (CMOS) based imaging sensors for digital cameras operate in RS acquisition mode. In this mode, the sensor scans row-by-row of pixels (line-wise) the entire image, with a delay between each row. This scan process is tied to the system clock tick and limited by the sampling rate of the Analog to Digital converter, A/D. In this mode, the whole time the camera is on, the pixel sensors continuously integrates the light that falls into their surface, until a reset signal discharges them. This reset signal acts over an entire row of pixels during a short t_{reset} time. After resetting, the row of pixels is exposed during a $t_{exposure}$ time. Finally every pixel is read out by the ADC, $t_{readout}$. This procedure is repeated for each row. The time between the start of one row to the consecutive one it is called $t_{rowshift}$, and is generally smaller than the exposure time, thus

producing an overlapping between the row scanning times, as shown in Fig. 1.

III. IMPLEMENTATION

The implementation of the simulator has been carried out using Python (with Numpy, Scipy and Matplotlib libraries). The experimental setup shown in a previous work [6] was modified and used to define key parameters of the simulation. Real image frames were taken using an experimental setup, shown in Fig. 3, which is comprised by a Raspberry Pi computer equipped with its camera and display, and a self-made VLC lamp with RGB LED stripes and a front-end that allows using switching modulations like OOK.

In the following of this section, the main steps of the simulator are explained. The flow diagram shown in Fig. 2 is useful to understand the whole process. From the side of the receiver (camera) an image mask and the PSF from the source (LED) are calculated and convoluted into a power-spread or power distribution frame, as explained in Section II. From the side of the source (LED), random data is generated and converted into a time domain signal that travels through the channel and reaches the sensor. Finally, the RS process takes place by integrating the signal in various times, one for each line, as shown in Fig. 1.

A. Preparation of Energy Distribution Frame

The calculation of Mask and the PSF can be understood by the following point of view: the Mask is made from projecting the source's visible area over the frame through the center point of the lens of the camera, while the PSF is made from projecting one point of the source over the area of the frame. All the projections are made using the RTM of the system, where inputs y , θ on the lens are computed using coordinates systems relative to source and destination, *i.e.*, calculating point-to-point vectors and using rotation-translation linear algebraic tools. The result is shown in Fig. 4, where Mask and PSF are plotted in units of pixels. The size of the PSF will vary depending on the focal length of the lens and the positions of Tx and Rx.

Note that both calculations use an area represented by a fine sweep of discrete points, then, to obtain pixel-sized values, an interpolation has to be carried out. Output at this point is obtained by the convolution of both Mask and PSF, and this is the power distribution over pixels of the sensor for a fixed position of Tx and Rx.

B. Transmitted Waveform

The data to be transmitted is randomly generated and then separated into three data streams, one for each channel (R, G, and B). Since the modulation used in this system is OOK and, for instance, no channel codes are being used, the binary stream is just translated into pulses of width t_{chip} . The time step width t_{step} for the simulation's global time variable at this point has to satisfy Eq. 7 in order to have enough resolution.

$$\frac{1}{t_{step}} > \max\left(\frac{2}{t_{chip}}, \frac{2}{t_{rowdelay}}, \frac{2}{t_{rowexposure}}\right) \quad (7)$$

The impulse response of the LED and its driver are neglected since the frequency t_{chip}^{-1} is small compared to a typical LED frequency response. The LED lamp used in the experimental setup is built by 5050 surface mount device (SMD) RGB chips. Their radiation pattern is available in .IES format [7]. For this format, a matrix of azimuth and elevation angles contains data of luminous flux per solid angle, or candelas. Since the distance between Tx and Rx is big enough, it can be assumed that light comes from the same solid angle.

Finally, the power impinging the camera's main lens is calculated using Eq. 3, and by using the empirical radiation pattern data as shown in Eq. 8:

$$P_{Rx} = P_{Tx}(\theta_{LED}, \phi_{LED}) \cdot \cos \theta_{camera} \cdot \frac{A_{lens}}{d^2}, \quad (8)$$

where, $P_{Tx}(\theta_{LED}, \phi_{LED})$ is the power obtained from the LED radiation pattern at inclination and azimuth angles θ_{LED}, ϕ_{LED} , and θ_{camera} is the incident inclination angle at the camera.

C. Rolling Shutter Acquisition

The experimental setup was built using a Raspberry Pi equipped with its original Camera V2 (Sony IMX219PQH5-C CMOS sensor [8]), and touchscreen display for easy input. For simulation, the power spreading or power distribution frame is amplified by ISO gain G_{ISO} by using $G_{iso} = 10^{iso/10}$, where iso is the ISO value.

The RS process is then carried out by generating frame trigger times. It can be considered that a frame is taken every $1/fps$ seconds where fps is the frame rate. In a real situation, the frame trigger times are conditioned by the output buffer readout operation performed by the operative system. This leads to non-deterministic frame trigger times, which are modeled following a normal distribution.

Line acquisition triggers depend only on the camera's delay between row expositions $t_{rowdelay}$. In the case of the Raspberry Camera V2 $t_{rowdelay} = 18.904 \mu s$. With these triggers, the received signal is integrated by windows of width $t_{rowexposure}$ that can be set arbitrarily for the capture. It can be also thought as a moving average of length $t_{rowexposure}$ and a stride of $t_{rowdelay}$. This operation has been performed numerically by convolving the transmission waveform with the exposure window, and then decimating the resulting signal by the appropriate factor (determined by $t_{rowdelay}$). Equation 9 describes the moving average respect to the camera's parameters and the transmitted waveform $f(t)$.

$$v_{x,y,c} = G_{iso} \cdot i(x, y, c) \int_{y \cdot t_{row}}^{y \cdot t_{row} + t_{exp}} f(t) dt \quad (9)$$

To model different noise sources, such as shot noise, and thermal noise, it's assumed that all can be modelled by Normal distributions. A frame-size matrix of additive noise values per pixel N is defined by the sum of each noise's standard deviation as follows:

$$\sigma_{x,y,c} = \sqrt{G_{iso}(\sigma_{th}^2 + \sigma_{sh}^2)}, \quad (10)$$

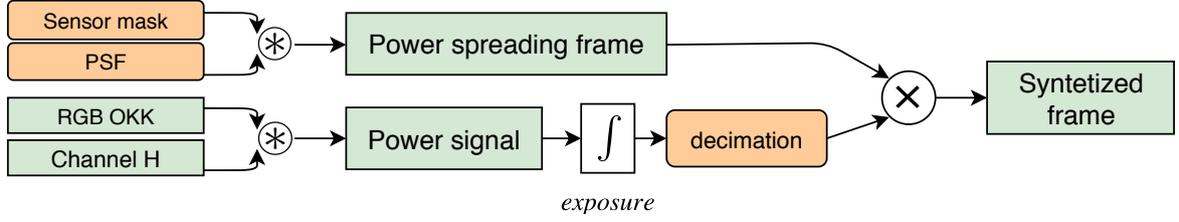


Fig. 2. Flow diagram of the simulator implementation.

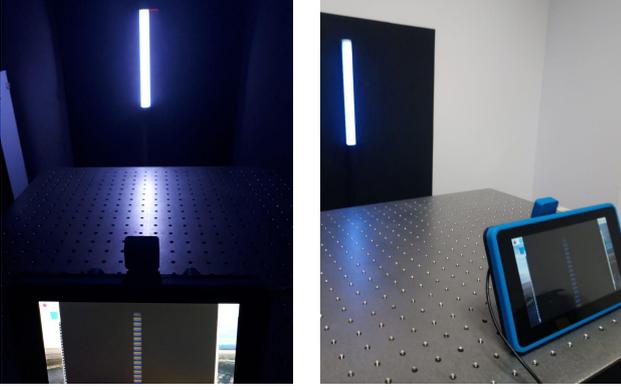


Fig. 3. Pictures of the real setup.

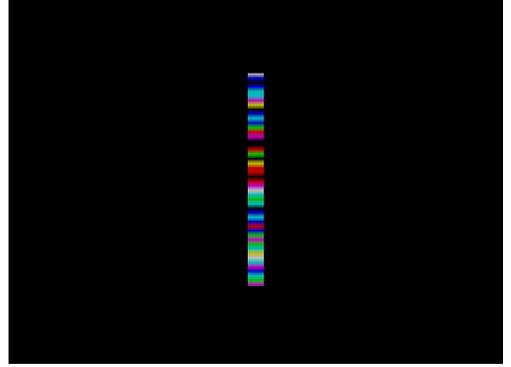


Fig. 5. Synthetic frame generated by the simulator.

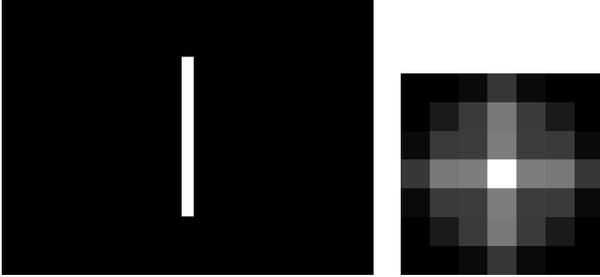


Fig. 4. Mask (left) and PSF (right) obtained to calculate power distribution.

where, σ_{th}^2 is associated to thermal noise and σ_{sh}^2 to shot noise. Thermal noise is independent to input, and it's defined by the following:

$$\sigma_{th}^2 = \frac{4k_B T_n B}{G_{iso}}, \quad (11)$$

where, k_B is the Boltzmann's constant, T_n is the noise temperature, and $B = 1/t_{rowdelay}$ is the bandwidth. The rest of the standard deviations depend on the input, and can be reduced to a single term σ_{sh} calculated by using:

$$\sigma_{sh}^2 = 2q_e (i(x, y, c) + i_d) B, \quad (12)$$

where, q_e is the elementary charge of the electron and i_d is the dark current of the camera's pixels. Finally, to obtain the synthesized frame, gamma correction is applied in the digital domain (although it can be performed also in the analog domain) to the image so that the final value of each pixel $p_{x,y,c}$ is obtained by:

$$p_{x,y,c} = T_{adc} (v_{x,y,c} + N_{x,y,c})^\gamma, \quad (13)$$

TABLE I
KEY PARAMETERS OF THE EXPERIMENT'S SETUP

Key parameter	Value
Tx t_{chip} (μ s)	300
Tx height (cm)	24
Tx width (cm)	2.2
Test distance, d (cm)	80
Sensor	Sony IMX219 [8]
Sensor resolution (pixels)	3280 \times 2464
Pixel size (μ m)	1.12
Camera	HFoV, VFoV
Focal length	3.04 mm
ISO	200
Analog gain	3.66
Digital gain	1
$t_{rowdelay}$ (μ s)	18.904
$t_{rowexposure}$ (μ s)	85, 293, 633

where $N_{x,y,c}$ is the pixel noise and T_{adc} is the ADC conversion function.

The output of the simulator at this point are synthetic frames, like the one shown in Fig. 5. Using the parameters shown in Table I, the simulator and the experimental setup were ran to obtain real pictures, shown in Fig. 6 and synthesized frames, shown in Fig. 7.

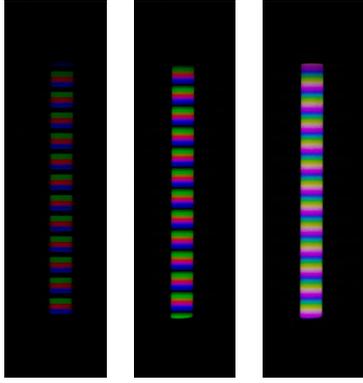


Fig. 6. Real pictures of LED Tx taken with experimental setup using $t_{\text{rowexposure}} = 85 \mu\text{s}$, $293 \mu\text{s}$, and $633 \mu\text{s}$ respectively.

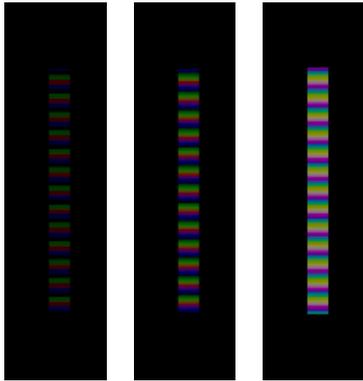


Fig. 7. Synthetic pictures of LED Tx made with simulator using $t_{\text{rowexposure}} = 85 \mu\text{s}$, $293 \mu\text{s}$, and $633 \mu\text{s}$ respectively.

IV. CONCLUSIONS

This paper shows the implementation of a simulator for the RS acquisition mechanism of the camera in an OCC setting, including a simplified model for optics of the system by means of RTM. The simulated system's parameters were tested on a real setup using Raspberry Pi Camera V2 and SMD 5050 RGB LED self-made lamp, enabling to compare the output synthesized image frames with real images.

The optical model reproduces the image formation in the camera based on the computation of the power distribution of the source's signal over the camera frame. The lens array of the camera is simplified by using RTM, which improves the computation load but reduces accuracy. To resemble the row-by-row scanning of the RS mode, the received signal is integrated by windows of wide equal to the arbitrarily selected exposure time of the camera.

The main contribution of this paper is the implementation of the RS simulation, which allows generating synthetic image frames of multiple low-speed VLC sources with arbitrary shapes. The synchronization of transmitters and receivers can be fixed while varying parameters of the simulation. This would enable, for example, to optimize the design of modulation schemes aided by machine learning.

REFERENCES

- [1] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas, "Using a CMOS camera sensor for visible light communication," in *2012 IEEE Globecom Workshops, GC Wkshps 2012*, 2012.
- [2] A. Duque, R. Stanica, H. Rivano, and A. Desportes, "CamComSim: a LED-to-Camera Communication Simulator," INSA Lyon ; INRIA Grenoble - Rhône-Alpes ; CITI - CITI Centre of Innovation in Telecommunications and Integration of services ; Rtone, Tech. Rep. RR-9114, 10 2017. [Online]. Available: <https://hal.inria.fr/hal-01625734>
- [3] M. S. Ifthekhar, M. A. Hossain, C. H. Hong, and Y. M. Jang, "Radiometric and Geometric Camera Model for Optical Camera Communications," in *2015 SEVENTH INTERNATIONAL CONFERENCE ON UBIQUITOUS AND FUTURE NETWORKS*, 2015, pp. 53–57.
- [4] Z. Ghassemlooy, L. N. Alves, S. Zvanovec, and M.-A. Khalighi, *Visible Light Communications: Theory and Applications*. CRC Press, 2017.
- [5] P. Chavez-Burbano, V. Guerra, J. Rabadan, D. Rodríguez-Esparragon, and R. Perez-Jimenez, "Experimental characterization of close-emitter interference in an optical camera communication system," *Sensors*, vol. 17, no. 7, p. 1561, 2017.
- [6] C. Jurado-Verdu, V. Guerra, J. Rabadan, R. Perez-Jimenez, and P. Chavez-Burbano, "RGB Synchronous VLC modulation scheme for OCC," in *2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2018*, 2018.
- [7] FlexfireLEDs, *Photometric Testing Data and IES File Downloads*. FlexfireLEDs, 2017. [Online]. Available: <https://www.flexfireleds.com/photometric-data-ies-files/>
- [8] Sony Corporation, *IMX219PQH5-C Datasheet*, 2014.