Using a CMOS Camera Sensor for Visible Light Communication

Christos Danakis*, Mostafa Afgani*, Gordon Povey*, Ian Underwood*† and Harald Haas*†
*PureVLC Ltd., ETTC, Kings Buildings, Mayfield Road, Edinburgh, EH9 3JL, UK
Email: {christos.danakis, mostafa.afgani, gordon.povey}@purevlc.com
†School of Engineering, The University of Edinburgh, UK
Email: {I.Underwood, H.Haas}@ed.ac.uk

***Abstract*—In this paper, a novel scheme for data reception in a mobile phone using visible light communications (VLC) is proposed. The camera of the smartphone is used as a receiver in order to capture the continuous changes in state (on-off) of the light, which are invisible to the human eye. The information is captured in the camera in the form of light and dark bands which are then decoded by the smartphone and the received message is displayed. By exploiting the rolling shutter effect of CMOS sensors, a data rate much higher than the camera frame rate is achieved.**

I. Introduction

Visible light communication is considered as a promising future wireless communications technology. Progress in light emitting diode (LED) technology has driven researchers to go one step further from illumination and use this technology for indoor wireless data transmission [1].

Over the last decade, mobile phones with embedded cameras have become common. The majority of new generation smartphones have built-in Complementary Metal-Oxide-Semiconductor (CMOS) cameras providing the ability to capture photos and videos. Fig. 1 shows the schematic of a typical CMOS image sensor. Each sensor array of pixels is triggered row-sequentially – a process known as rolling shutter. This effect allows us to transmit with data rates that far exceed the frame rate of the camera.

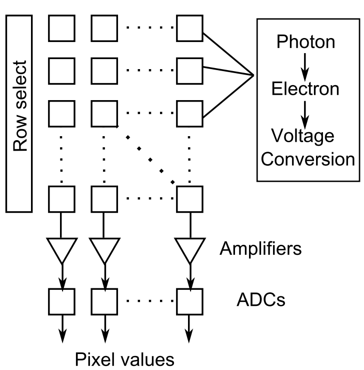

Fig. 1. CMOS camera schematic: The CMOS pixel sensors are exposed sequentially and convert the incident photons to a voltage which results in the pixel value.

In this paper we develop a model for the transmission of data from an LED to a smartphone using the built-in camera as a receiver. The model (Fig. 2) involves the transmitter protocol and an Android application for the signal reception and decoding at the smartphone. Using this model, data can be transferred reliably to a mobile device.

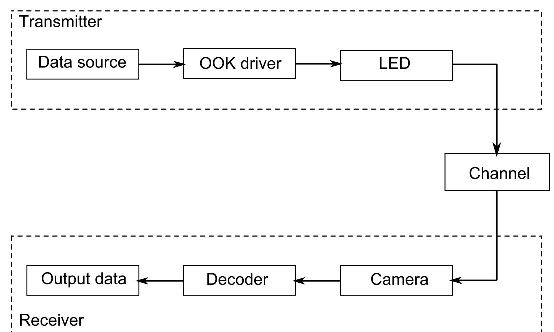

Fig. 2. System block diagram: The upper part shows the transmitter (which consists of an On-Off Keying driver, and an LED) and the lower part shows the receiver (mobile phone).

The paper is organized as follows. In Section II the rolling shutter effect and its use in visible light communication is described. In Section III the transmitter is described. The description includes a brief discussion of the transmitter circuit and a detailed description of the encoder. Section IV describes the receiver and its properties. In Section V, the Android App and the decoder included in it, are presented. The performance analysis is discussed in Section VI. This is followed by the conclusion in Section VII.

II. Rolling Shutter

The rolling shutter mechanism is a method of image acquisition used by CMOS sensor cameras. The sensor operation is shown in Fig. 1. A CMOS pixel converts incident photons into electrons which are then converted to a voltage, from which the pixel value is obtained. The level of signal generated by the image sensor depends on the amount of light incident on the imager, in terms of both intensity and duration [2].

Most CMOS sensors contain pixel that are arranged in sequentially activated rows (scanlines) and therefore do not capture the entire image at once. On activation, each scanline of the sensor array is exposed, sampled and stored sequentially [3]. When this procedure is completed the scanlines are merged together in order to form a single image. "Rolling

978-1-4673-4941-3/12/$31.00 ©2012 IEEE

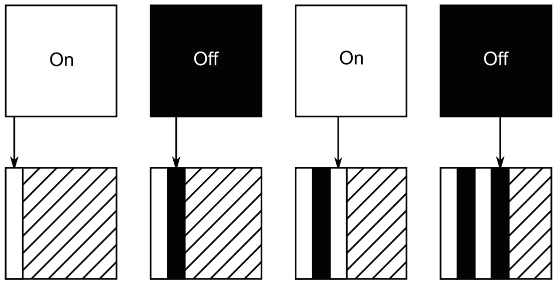

Fig. 3. Rolling shutter operation: The top squares indicate the state of the LED light. The bottom squares show the output of the CMOS camera as it enables one scanline at a time, forming the final image.

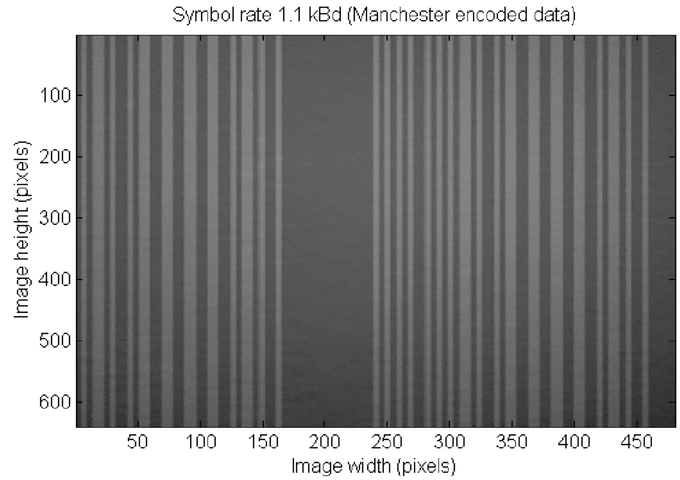

Fig. 4. Sample image of CMOS camera output – The result of the rolling shutter as taken from the preview mode of the Dell Streak mobile phone (at 1.1 kBd transmitted baud rate).

shutter" is the term used to describe this process. Fig. 3 shows the procedure of capturing the lines of the image one at a time.

Various effects can be observed due to the rolling shutter operation such as skew seen in images of a moving object. While this may seem undesirable, this property of the CMOS cameras can actually be utilized in optical wireless communication in order to transmit data from an LED to a mobile phone. When the flashing frequency of the LED is lower than the rolling shutter's scanning frequency but higher than the frequency of the preview display of the camera (frames per second or fps), bands of different light intensity appear in the image (Fig. 3). When the LED is on, the camera sees a bright frame and the CMOS sensor exposes one array of this image, which is shown in Fig. 3, as the first white line in the image. The transmitter then changes to the off state and the second scanline is enabled, which results in the first black line in the image. This procedure continues until all the scanlines are exposed and the image is completed.

The width of these bands is proportional to the symbol rate of the transmitter and the rate the camera captures preview images (fps). By adjusting these values, an array of images containing bands with different width and intensity can be obtained (Fig. 4). Using simple image processing techniques these bands can be converted to a binary array from which useful information can be extracted.

While the camera is limited to a capture rate of approximately 20 to 30 frames per second, the rolling shutter effect allows the capture of multiple information bits (LED states) inside every frame, which leads to speeds significantly higher than the frame rate of the camera.

III. Transmitter

An LED, connected to a simple on-off keying (OOK) transistor switch circuit, is used as a transmitter. The encoding procedure at the transmitter is as follows.

A. Encoder

The data is split into blocks of a given number of bits and the block sequence number (BSN) is added in front of every block. Then, each block is encoded using Manchester coding which follows the IEEE 802.3 convention [4]. The result is two chips for every bit. Therefore, each bit is transformed into a single Manchester coded symbol. A header is inserted at the beginning and a trailer at the end of every block, which carries the information needed for start of the block (SB) detection.

The BSN bits are encoded together with the data using Manchester coding. The SB symbols are uncoded and consist of four chips (ones) at the header and two chips (one-zero) at the trailer, as shown in Fig. 5. Finally, every block is transmitted twice. This method guarantees that at least one of the two identical blocks is included in every preview frame. When the final block is transmitted, the block sequence repeats from the start, for reasons that will be covered in Section V.

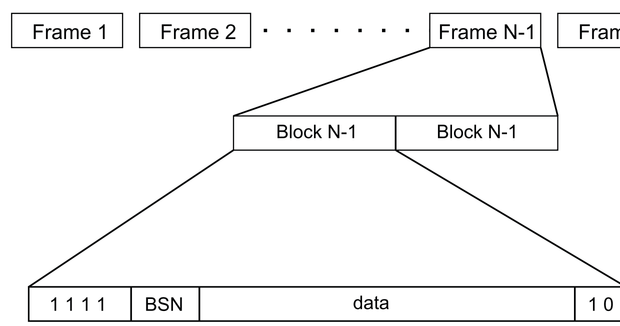

Fig. 5. Block structure of the transmitted signal. Each transmitted frame consists of a block and a copy of it. Each block contains the header (SB detection bits and BSN), the data and the trailer (SB detection bits).

There is an upper limit on the achieveable data rate imposed by the rolling shutter operation which is dependent on the hardware. At the upper limit, the width of the band is one pixel, whereas if higher rates are used, overlapping between adjacent bands will occur. At the other extreme, when the transmitter operates in low baud rates, flicker is observed. The flicker is caused by the SB detection pattern frequency. The pattern is four chips long (four ones – *i.e.,* 2 ms at 1 kBd) and appears twice per photo frame (48 ms at 1 kBd). The frequency is approximately 40 Hz which is in the visible range and thus flicker is observed.

The flicker effect is alleviated by adding a DC bias to the transmitted signal. The DC bias increases the mean of the signal and reduces its distance from the on state value. Although this measure affects the light intensity range, the distinction between the light and dark bands in the image remains reliable.

IV. Receiver

The light from the transmitter is used to illuminate a surface and the reflection is received by the camera of the smartphone. The preview mode of the camera is used to capture a continuous array of frames and then, a decoder based on Java is applied frame by frame.

As previously mentioned, the application takes advantage of the rolling shutter effect of CMOS cameras in combination with the ability of the mobile phone to store temporarily the continuous preview images in a buffer. In order to achieve maximum performance, a buffer queue of three is used. Every time a preview frame is captured, it is stored in a buffer in order to be processed by the decoder. After processing, the buffer is released and returns to the queue. This process continues until the data sent from the LED is received.

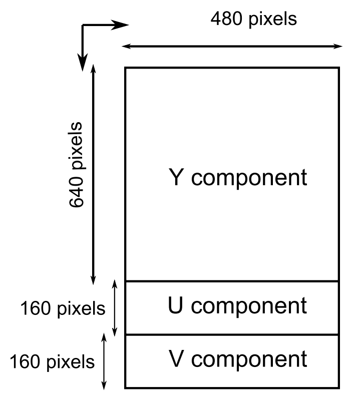

Fig. 6. Example YUV file structure for a 640×480 pixel image. Each YUV file consists of three parts: the Y luminance and the U & V chrominance components.

A. Data extraction from YUV format file

The preview images are stored in a buffer using the YUV format, the output of which is a matrix consisting of three components: the Y luminance and the U and V chrominance components (Fig. 6). The first one carries information regarding the brightness whereas the other two contain the colour differences. Since the U and V components include only information regarding the colours of the image, they can be discarded from the decoding procedure. Thus, the Y component carries the information needed for the decoder, since the transition between black and white is clearly indicated and provides enough information for the decoding process.

V. Android Application and Decoder

The Android application enables the preview mode of the camera and the resulting images are stored in the buffer. Then, the decoder is called every time a preview is obtained. As previously described, the decoder input is the Y luminance component of the YUV format image and the output is the binary array representation of the received data. This method involves seven main stages as shown in Fig. 7:

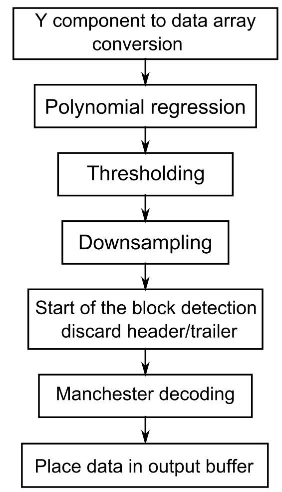

Fig. 7. Decoder Flow chart

1) Y component to data array conversion: The raw data from the image must be converted into an acceptable single dimension waveform in order to be decoded. The Y component of the continuous stream of raw data in the YUV file is reshaped to meet the image dimensions and the mean of every column is calculated. For example, using a 0.3 megapixel camera, the YUV raw data array is 960×480 pixels long (Fig. 6) and the Y component is the first 307200 values. An example image and the resulting one dimensional sequence are shown in Fig. 8.

2) Polynomial regression: The image is not homogeneous in terms of brightness mostly because the receiver is not stable (the distance and relative position between transmitter and mobile phone varies) and thus the intensity levels of black and white in a single image are not normalized (Fig. 8). As a result, the data stream is not normalized as well and a clear distinction between ones and zeros cannot be readily achieved.

A polynomial fitting (represented by the dashed blue line in Fig. 8) is applied on the data stream and its output is subtracted from the latter, resulting in a normalized version of the data sequence with the same reference level for all values. Simulations showed that a 3^{rd} degree polynomial is the most appropriate approximation.

The polynomial regression normalizes the data sequence by bringing it down to the same reference level. More specifically, by subtracting the polynomial fitting values from the data stream the data is now centered at zero.

3) Thresholding: Afterwards, a threshold is applied to the normalized data sequence in order to distinguish the two levels (1 and 0) of the transmitted signal. Since the polynomial regression method results in a data stream centered at zero, the threshold is set to be zero. Thus, every value that is above

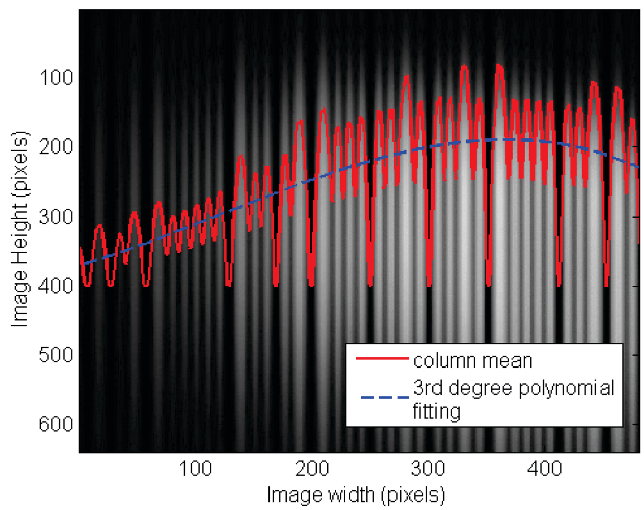

Fig. 8. A single preview frame. The red (continuous) line indicates the mean of the pixel values per column and the blue (dashed) line shows the output of the polynomial fitting.

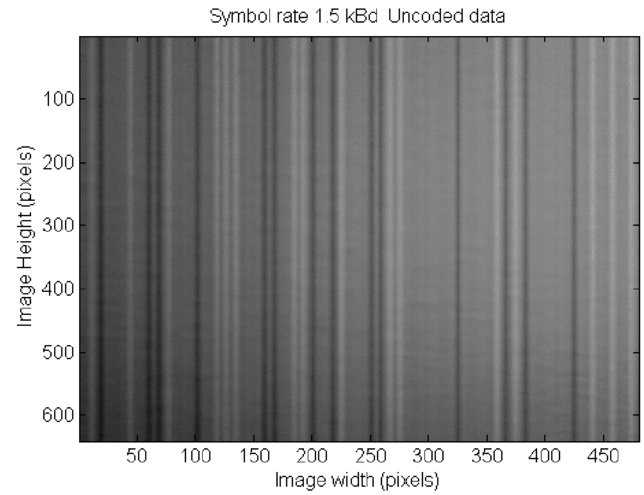

Fig. 9. Example of the reception of uncoded data at 1.5 kBd. There are multiple gray levels which affect the hard decision between 1 and 0.

zero is considered as a one whereas every value below zero is considered as a zero.

4) Downsampling: After applying the threshold, the data stream must be downsampled in order to give a single bit for every single band and thus a continuous bit sequence.

5) Start of the block detection - discard header/trailer: A sliding window is then applied to the sampled sequence, in order to detect the start of the block. The detection becomes more reliable by using the trailer from the previous block, since the sliding window searches for a sequence of four ones following a single one and a single zero. Once the start of the block is detected, the header and the trailer are discarded and the remaining parts of the block (the BSN together with the data) proceed to the next stage of the decoder.

6) Manchester decoding: The Manchester coding scheme provides a robust method for error detection. In high data rates, the transmission of uncoded data in combination with the rolling shutter effect results to the appearance of various gray levels in the preview image (Fig. 9). These levels make distinction between one and zero states difficult and thus a decision threshold cannot be applied.

Manchester encoding ensures that the maximum sequence of similar chips can only have a length of two, compared to the uncoded case where unlimited symbols of the same value can appear next to each other. Therefore with Manchester encoding, the different widths of the bands in a preview image can only contain one or two chips of the same kind (Fig. 8). Any other case leads to an erroneous reception causing the received data to be discarded.

The only exception is the proposed start of the block detection scheme, which uses a chip sequence of four ones. At the start of the block detection stage, the sliding window detects a bright band with a width larger than the two expected values, which indicates the start of a frame.

7) Adding data to buffer using the BSN: The BSN indicates the position of the block in the output buffer. The data is placed in the buffer and the decoder moves on to the next preview image. The procedure is repeated until the output buffer is filled. Once this is achieved, the binary sequence is converted to ASCII characters and the message is displayed on the smartphone screen.

As a result of the various functions and Java methods used by the decoder, it takes approximately 50 ms on average for the mobile phone to complete the procedure, per image. In addition, a plethora of other processes run in the background when the current application is active, which causes changes in their prioritization. Thus, the reception is not continuous and some frames are inevitably lost. Fig. 10 shows the distribution of the intervals between the capture of the preview frames.

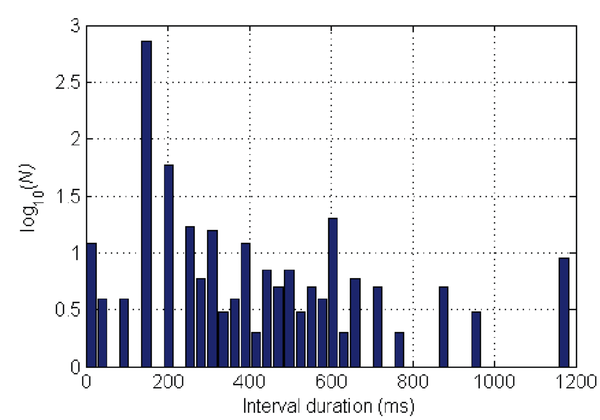

Fig. 10. Interval distribution (1 kBd transmitted baud rate): The figure shows the $\log_{10}(N)$ of the number of the intervals between adjacent frames (in ms).

Changing the priority status of the running applications does not affect the discontinuity of the reception of the packages significantly. Furthermore, one of the major characteristics of the communication link is that it is simplex only (the mobile

TABLE I
DELL STREAK CONFIGURATION

Dell Streak Configuration	
Image format	YUV (NV21)
Scene mode	Sports
FPS	set to maximum (20 fps)
Preview image size	640 × 480 pixels
b_p (bits per pixel)	12
Buffer size	460800 bits
Minimum baud rate	1 kBd
Maximum baud rate	3.1 kBd

phone cannot send any data to the transmitter). Thus, the transmitted signal is repeated continuously so that all blocks are received successfully. Since there are processes of crucial importance running in the background, any changes at the reception stage is unlikely to provide any further improvement.

VI. PERFORMANCE ANALYSIS

For the tests, a Dell Streak Android mobile phone with a 5 megapixel camera is used as a receiver. A demonstration of the tests can be seen in [6]. The Android App enables the preview mode of the mobile phone and also calls the Java decoder. The camera settings used for running the application on the specific smartphone are shown in TABLE I.

The system is evaluated by transmitting text messages from an LED to the Dell Streak smartphone. In this Section, the results for the minimum and maximum baud rate are discussed. Taking Manchester coding into consideration, this is interpreted as 1 and 3.1 kbps respectively.

In the first case (1 kBd) the transmitted text message is 16 ASCII characters long. Three BSN bits are used which allow the transmission of eight frames in total. Thus, two ASCII characters are included in every frame. The repeatable experimental conditions set the distance from the LED to the illuminated surface to 35 cm and the distance from the surface to the camera lens as 9 cm. However, greater distances can be achieved in practice. Calculations show that on average the image decoder requires 182 ms to operate. The decoder time is higher than expected because of delays caused by the background running applications which affect its performance, as described previously. The discontinuity of the reception affects the number of frames that need to be received before the message is completed at the receiver. Thus, 14.5 images are needed on average. This also affects the average total time for a successful reception, which is approximately 2.5 s. Furthermore, the overall time for a complete continuous transmission of the message is 384 ms. Thus the transmitter needs to repeat sending the message at least seven times.

In the second case (3.1 kBd) the text message is 72 characters long. The SB detection scheme, as well as the number of BSN bits do not change. Thus, we are able to send eight frames of nine ASCII characters each. In this case, the distance between the LED and the surface is set to 12 cm while the distance between the surface and the mobile phone is set to 8 cm. In this experiment, the decoder time is slightly increased (186 ms) and the number of frames needed for the reception is now 23.1. The average total time is 3.9 s and by calculating the time needed for the transmission of the message (1.136 s), the message needs to be sent at least five times.

The configuration values for the smartphone that are used for the experiments are not a general limitation for the algorithm. The results described in this Section are smartphone dependent. Any hardware or software changes may result in different figures. In addition, the distance between the LED and the surface, as well as the distance between the surface and the smartphone may vary. These variations result in different light intensity values reaching the CMOS camera sensors. Furthermore, using a different type of reflected surface (*i.e.*, a surface with a darker or more complex background) may cause changes to the received light intensity also. Finally, the level of ambient light of the surrounding environment is of crucial importance for the clear distinction between the on and off states, since it affects the image intensity during the off state of the LED.

VII. CONCLUSION

In this paper the potential of using visible light communications for data transmission to a mobile phone is investigated. A system, consisting of an LED transmitter and the mobile phone camera as a receiver, is presented. A proof-of-concept has been established, demonstrating that the rolling shutter effect can be constructively exploited to achieve data rates multiple times faster than the frame rate using an embedded CMOS camera sensor.

As far as future work is concerned, in order to increase the data rate, the application of the Luby Transform (LT) codes is to be investigated [5]. In addition, reliable data extraction from a complex background is desirable.

ACKNOWLEDGMENT

The authors would like to thank Dr. Wasiu O. Popoola for his valuable comments and contributions and Saket Kumar for his contribution to the project.

REFERENCES

[1] Elgala, H., Mesleh, R. and Haas, H., "Indoor Broadcasting via White LEDs and OFDM", *IEEE Transactions on Consumer Electronics*,vol. 55, No. 3, pp. 1127–1134, 2009.

[2] Kodak Image Sensor Solutions, "Shutter operations for CCD and CMOS image censors", *Application note: MTD/PS–0259*, Rev. 3, June 2011.

[3] Chia-Kai Liang, Yu-Chun Peng and Homer Chen, "Rolling shutter distortion correction", *SPIE Visual Communications and Image Processing*, vol. 5960, pp. 59603V–1, 2005.

[4] "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Networks–Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Section One", *IEEE Std 802.3-2008 (Revision of IEEE Std 802.3-2005)*, pp. c1–597, 2008.

[5] Luby, M., "LT codes", *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pp. 271 - 280, 2002.

[6] pureVLC, LightMessage from pureVLC, http://youtu.be/IaxD4SF3jsA