# Low-cost, Low-Power and Long-range Image Sensor for Visual Surveillance

Congduc Pham
University of Pau, LIUPPA Laboratory
congduc.pham@univ-pau.fr

## ABSTRACT

We present a low-cost, low-power, long-range image sensor built from off-the-shelves components and enabling fully autonomous and out-of-the-box remote visual surveillance applications. The image sensor implements a packet loss-tolerant image compression technique that can run on very limited memory platforms. Long-range support is provided with Semtech LoRa technology. We detail the performance and energy consumption of the image sensor and describe how the system can be deployed in real-world scenarios using battery-operated image nodes.

## CCS Concepts

•**Hardware** → **Wireless integrated network sensors;** *Sensor applications and deployments;*

## Keywords

Wireless image sensor networks; long-range transmission; low-power; long-range surveillance

## 1. INTRODUCTION

The concept of the Internet of Things (IoT) in which objects are capable of interacting with the physical world and with other objects is federating more people on a large variety of applications mostly related to monitoring or surveillance tasks. Going beyond simple physical measures such as temperature or luminosity the possibility to provide visual information can greatly enhance a number of surveillance applications. Obviously, given the low level of resources of IoT platforms, implementing an image sensor with fast but still efficient image processing is a challenge. Image sensor boards such as Cyclops [1], MeshEyes [2], Citric [3], WiCa [4], SeedEyes [5], Eye-RIS [6], Panoptes [7], CMUcam3-FireFly [8, 9], CMUcam4 and CMUcam5/PIXY [9] and iMote2 with IMB400 [10] have been proposed during the last decade by the research community. Unfortunately, these platforms are either mostly based on ad-hoc development of the visual part (i.e. camera board with dedicated micro-controller) or rely on powerful micro-controller/Linux-based platforms. Most of them also lack an efficient image encoding and compression scheme adapted to low-bandwidth radios. In [11], we built our first image sensor prototype from off-the-shelves low-cost components by promoting maximum flexibility and modularity. Our motivations for the work described in [11] were: (1) to use only off-the-shelf components in order to provide maximum flexibility, evolutivity and reproducibility; and (2) to provide an efficient image compression algorithm which produces a packet stream tolerant to packet losses.

Regarding deployment issues, the potential of remote sensors is still held back by technical challenges such as short communication distances. Using the telco mobile communication infrastructure (e.g. GSM/GPRS, 3G/4G) to deploy remote devices is still very expensive and definitely not energy efficient for autonomous devices that must run on battery for months. Our developed image sensor initially relied on IEEE 802.15.4 communications and while such short range communications can be realized on smart cities infrastructures, it can hardly be generalized for the large majority of surveillance applications that need to be deployed in isolated, remote environments. In this paper, we propose an extreme long-range low-bandwidth radio version of our image sensor. Using recent modulation techniques such as those based on Sigfox$^{\text{TM}}$ or Semtech's LoRa$^{\text{TM}}$ technology, much longer transmission distance can be achieved without relay nodes. This interest is shared by many Machine-to-Machine (M2M), IoT and and Mobile Network Operators (MNO) actors: Low Power Wide Area Networks (LPWAN) technologies are gaining incredible interest with huge economic impacts behind. We demonstrate that the off-the-shelves component approach for our image sensor node allows for fast and efficient integration of these new technologies. Our motivations in proposing the long-range image sensor are then: (1) to avoid relying on operator-based communications; (2) to remove the complexity and cost of deploying a multi-hop short range infrastructure; and (3) to offer out-of-the-box visual surveillance facilities.

The image sensor works with raw 128x128 8-bbp gray scale image and integrates a simple-differencing technique with negligible processing time. The image can be compressed with various quality factors for reducing the bandwidth usage. This is especially important with long-range radio technologies where the longer distance is made possible at the cost of operating at very low data rates, i.e. less



*SMARTOBJECTS'16, October 03-07, 2016, New York City, NY, USA*



DOI: http://dx.doi.org/10.1145/2980147.2980156

than 10kbps in most cases. The rest of the article is organized as follows. Section 2 presents the main long-range radio technologies. Section 3 describes our low-cost, long-range image sensor design. Image processing performance and energy consumption measures will be presented. We then describe in Section 4 how out-of-the-box surveillance applications can be deployed with our system. In Section 5 we will show preliminary long-range image transmission tests. We conclude in Section 6.

## 2. LONG-RANGE RADIOS

Recent so-called Low-Power Wide Area Networks (LPWAN) definitely provide a better connectivity answer for remote devices as several kilometers can be achieved without relay nodes to reach a central gateway or long-range base station (LR-BS). Fig. 1 shows a typical long-range 1-hop connectivity scenario where the LR-BS is the single interface to Internet servers through cellular/ADSL/WiFi technologies depending on what is available locally. These long-range technologies, mainly represented by Sigfox$^{TM}$ and Semtech's LoRa$^{TM}$ technologies can achieve 20km or higher range in line-of-sight (LOS) condition and about 2km in dense, urban, in-door none-LOS [12] conditions.

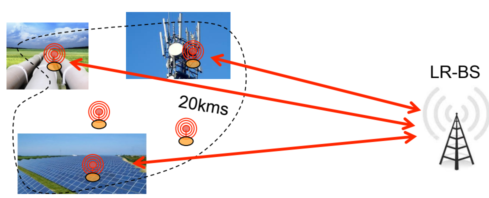


**Figure 1: Extreme long-range application**

LoRa technology, which can be privately deployed in a given area without any operator, has a clear advantage over Sigfox which coverage is entirely operator-managed. For public, large-scale deployment scenarios, The LoRa Alliance proposes the LoRaWAN [13] specification supporting multi-gateways scenario and full network/application servers architecture. However, it is also possible to use LoRa in a completely ad-hoc manner, using simpler end-device protocol stack and lower cost gateways to implement specific application use-cases. This is the approach we take for our long-range image sensor.

LoRa basically achieves longer distance by means of higher spread spectrum factors and much more robust modulation techniques. The LoRa modulation can be tuned by setting 3 main parameters [14]: BW, CR and SF. BW is the physical bandwidth for RF modulation (e.g. 125kHz). Larger signal bandwidth allows for higher effective data rate, thus reducing transmission time at the expense of reduced sensitivity improvement. CR is the coding rate for cyclic error coding to perform forward error detection and correction. Such error coding incurs a transmission overhead and the lower the coding rate, the higher the coding rate overhead ratio, e.g. with CR=4/5 the overhead ratio is 1.25 which is the minimum value. Finally SF is the spreading factor that can be set from 6 to 12. The lower the SF, the higher the data rate transmission but the lower the immunity to interference thus the smaller is the range. The longer range (higher receiver sensitivity) comes at the cost of a much lower data rate giving much higher transmission time (or so-called Time-on-Air, ToA). We show in Fig. 2 this ToA for various BW and SF combinations. Mode 4 to 6 provide quite interesting tradeoffs for longer range, higher data rate and immunity to interference. Mode 1 is the mode that can maximize the packet reception probability at longest range.

| | | | | time on air in second for payload size of | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LoRa mode | BW | CR | SF | 5 bytes | 55 bytes | 105 bytes | 155 Bytes | 205 Bytes | 255 Bytes |
| 1 | 125 | 4/5 | 12 | 0.95846 | 2.59686 | 4.23526 | 5.87366 | 7.51206 | 9.15046 |
| 2 | 250 | 4/5 | 12 | 0.47923 | 1.21651 | 1.87187 | 2.52723 | 3.26451 | 3.91987 |
| 3 | 125 | 4/5 | 10 | 0.28058 | 0.69018 | 1.09978 | 1.50938 | 1.91898 | 2.32858 |
| 4 | 500 | 4/5 | 12 | 0.23962 | 0.60826 | 0.93594 | 1.26362 | 1.63226 | 1.95994 |
| 5 | 250 | 4/5 | 10 | 0.14029 | 0.34509 | 0.54989 | 0.75469 | 0.95949 | 1.16429 |
| 6 | 500 | 4/5 | 11 | 0.11981 | 0.30413 | 0.50893 | 0.69325 | 0.87757 | 1.06189 |
| 7 | 250 | 4/5 | 9 | 0.07014 | 0.18278 | 0.29542 | 0.40806 | 0.5207 | 0.63334 |
| 8 | 500 | 4/5 | 9 | 0.03507 | 0.09139 | 0.14771 | 0.20403 | 0.26035 | 0.31667 |
| 9 | 500 | 4/5 | 8 | 0.01754 | 0.05082 | 0.08154 | 0.11482 | 0.14554 | 0.17882 |
| 10 | 500 | 4/5 | 7 | 0.00877 | 0.02797 | 0.04589 | 0.06381 | 0.08301 | 0.10093 |

**Figure 2: ToA as payload size is varied**

Given the much longer transmission time of an image with long-range radios, the surveillance application profile is different than with traditional short range radios. While using short range 802.15.4 can allow for almost real-time mission-critical intrusion detection applications, the long-range radio version is more suitable for visual surveillance systems such as periodic situation-awareness or visual detection of physical phenomenon such as cracks, leakages. . . .

## 3. LOW-COST, LONG-RANGE SENSOR

### 3.1 Hardware components

We use the Teensy3.2 [15] board as the host micro controller to drive the CMOS uCamII camera from 4D systems. The uCamII is shipped with a $56^o$ angle of view lens but we also use $76^o$ and $116^o$ lenses depending on the application needs. The uCamII is connected to the Teensy board through an UART interface at 115200 bauds. The uCamII is capable of providing both raw and JPEG bit streams but we are not using this last feature as it is impossible from the delivered JPEG bit stream to build a packet stream tolerant to packet losses. As a result, we retrieve raw 128x128 8-bpp grey scale images from the uCamII then we operate image compression on the board. The Teensy3.2 is a small form factor micro-controller board based on the MK20DX256 Cortex-M4. Its rated speed is 72MHz but it can be overclocked at 96MHz or slowed down to 48MHz and 24MHz. The board has 64KB of SDRAM.

Our first prototype used an XBee S1 module from Digi to provide short-range IEEE 802.15.4 connectivity. In this new version, we connect through the SPI bus a LoRa$^{TM}$ long-range radio module (an inAir9 from Modtronix) built upon Semtech's LoRa SX1276 chip [14]. Compared to legacy modulation techniques, Semtech's LoRa permits an increase in link budget and increased immunity to in-band interference. Fig. 3 shows the schematic of the developed image sensor built from off-the-sheves components. 2 leds are connected to indicate the operation mode of the camera and a 4-AA-battery pack is used for power supply. The small form factor of the Teensy board allows for easy integration in a small waterproof box for outdoor deployment as illustrated in Fig. 4.

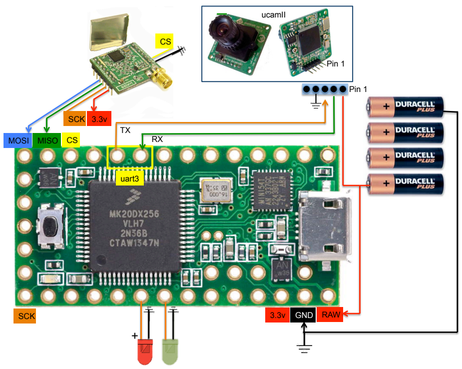


**Figure 3: Teensy3.2, LoRa radio and uCamII**

From the 1-camera system it is not difficult to have a multiple camera system: the Teensy has 3 UART ports therefore 3 uCamII can be connected. We have a 3-camera prototype with $116^o$ lenses that provides almost omni-directional coverage: the cameras are set at $120^o$ from each other and are activated in a round robin manner. When images need to be transmitted, each camera can be configured with a different image quality factor if necessary. Figure 4 shows the FoV of the 3 types of lenses and the 3-camera prototype.

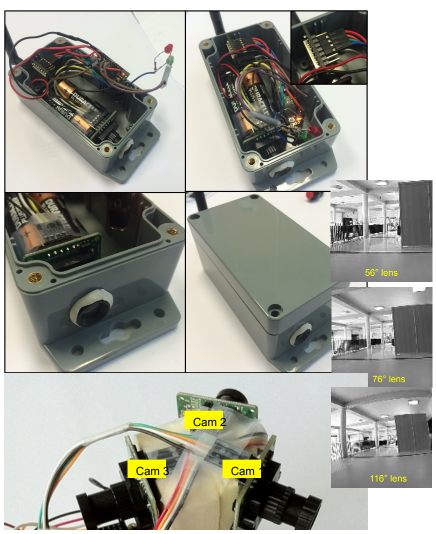


**Figure 4: Outdoor deployment**

## 3.2 Image compression/encoding

When using long-range radio the encoded bit stream needs to be particularly tolerant to packet losses. We use an optimized encoding scheme proposed in [16] which features the 2 following key points: (*i*) image compression must be carried out by independent block coding in order to ensure that data packets correctly received at the sink are always decodable and, (*ii*) de-correlation of neighboring blocks must be performed prior to packet transmission by appropriate interleaving methods in order to ensure that error concealment algorithms can be efficiently processed on the received data. The compression scheme is a JPEG-like coder and operates on 8x8 pixel blocks with advanced optimizations on data computation to keep the computational overhead low. The combination of the fast JPEG-like proposed encoder with an optimized block interleaving method [17] allows for an efficient tuning, the so-called Quality Factor (Q), of the compression ratio/energy consumption trade-off while maintaining an acceptable visual quality in case of packet loss.

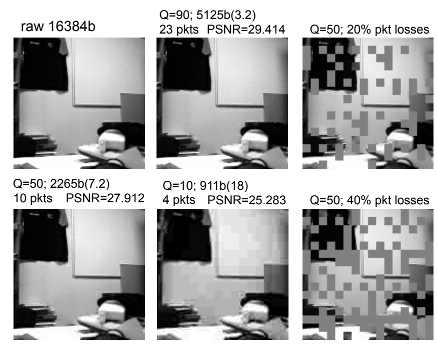


**Figure 5: Image at various quality factor**

Fig. 5 shows the original raw 128x128 image taken with the image sensor and encoded with various quality factors: Q=90 (high quality), Q=50 (medium quality) and Q=10 (low quality). The total size of the compressed image, the compression ratio (in bracket) the number of generated packets and the PSNR compared to the original image are shown. The number of packets depends on the image maximum segment size (IMSS) allowed per packet. We set it to 240B and in practice the produced packet size will slightly vary according to the packetization process. 9 bytes need to be added to the image payload: these are framing bytes (2B), 16-bit source image sensor address (2B), packet sequence number (1B), quality factor (1B), image payload size (1B) and offset of the first block of image data in the packet (2B). Then, when adding the 5 bytes of the radio protocol header, each produced packet size is close to the 255B limit of LoRa. Fig. 5 also shows the impact of packet losses on the image quality. We can see that the produced encoded bit stream is particularly robust to packet losses. Also, the decoding process can be realized with any number of received packets. This feature is particularly useful for very low-bandwidth and high-latency technologies such as long-range radios.

## 3.3 Image change detection

We implemented an image change detection mechanism based on "simple-differencing" of pixel: each pixel of the image from the uCam is compared to the corresponding pixel of a reference image, taken previously at startup of the image sensor and stored in memory. When the difference between two pixels, in absolute value, is greater than PIX_THRES we increase the number of different pixels, N_DIFF. In order to take into account slight modifications in luminosity due to the camera, when N_DIFF is greater than NB_PIX_THRES we additionally compute the mean luminosity difference between the captured image and the reference image, noted LUM_DIFF. Then we re-compute N_DIFF but using PIX_THRES+LUM_DIFF as the new threshold. If N_DIFF is still greater than NB_PIX_THRES we conclude for an image change and trigger the transmission of the image. For the intrusion detection test shown in Fig. 6, we set PIX_THRES to 35 and NB_PIX_THRES to 300. In doing so, we were able to systematically detect a single person intrusion at 25m without any false alert. Additionally, if no image change occurs, the sensor takes periodically a new reference image to take into account light condition changes.

**Figure 6: reference image (left); intrusion (right)**

The "simple-differencing" method is very light-weight: it requires for each pixel of the image only 1 addition (to compute the pixel difference) and 1 comparison (to compare with PIX_THRES) and 1 variable incrementation in case the difference is greater than PIX_THRES. This process is done on-the-fly while reading data from the uCam. Note that the "simple-differencing" process is performed on the raw image. Once a change is detected, a lower quality version of the image can be transmitted using the lossy compression scheme.

## 3.4 Image processing performance measures

Fig. 7 shows for various quality factors and CPU frequencies the image encode time and the packetization time. The last 2 column summarizes the number of produced packets and the encoded image size with the compression ratio when IMSS=240. The time to read the raw image data from the uCam is constant and takes 1512ms. All these measures are taken without transmission of packets and are expressed in ms.

| | | | | | | | | | MSS=240 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 96MHz | | 72MHz | | 48MHz | | 24MHz | | N | S |
| Quality Factor Q | encode time | packetization time | encode time | packetization time | encode time | packetization time | encode time | packetization time | number of packets | size in bytes (compression ratio) |
| 100 | 204 | 220 | 234 | 281 | 309 | 420 | 565 | 813 | 47 | 9982 (1.64) |
| 90 | 226 | 86 | 263 | 109 | 349 | 164 | 644 | 322 | 23 | 5090 (3.21) |
| 80 | 226 | 58 | 262 | 76 | 348 | 112 | 640 | 218 | 16 | 3595 (4.55) |
| 70 | 225 | 48 | 261 | 63 | 347 | 92 | 640 | 178 | 13 | 2842 (5.76) |
| 60 | 224 | 44 | 261 | 56 | 347 | 82 | 639 | 162 | 11 | 2461 (6.65) |
| 50 | 222 | 39 | 258 | 51 | 342 | 75 | 630 | 150 | 10 | 2129 (7.69) |
| 40 | 224 | 37 | 260 | 47 | 346 | 69 | 637 | 139 | 9 | 1898 (8.63) |
| 30 | 224 | 33 | 260 | 44 | 345 | 64 | 637 | 127 | 7 | 1608 (10.19) |
| 20 | 223 | 31 | 260 | 39 | 345 | 58 | 636 | 115 | 6 | 1279 (12.81) |
| 10 | 223 | 26 | 260 | 31 | 345 | 50 | 636 | 99 | 4 | 824 (19.88) |
| 5 | 223 | 23 | 259 | 31 | 344 | 45 | 635 | 89 | 3 | 503 (32.57) |

**Figure 7: Image processing performance**

We can see that the encoding time is quite constant, except for Q=100. With the maximum quality factor value, the encoding stage has little to do, which explains the smaller encoding time. In this case, however, as the image size is large, the packetization time is much higher. Practically, the quality factor can be set to 20 or smaller in order to reduce the image size for transmission using the long-range radio.

## 3.5 Energy consumption measures

To make the energy consumption (EC) measures we inserted additional power consumption by toggling a led to better identify the various phases of the image sensor operations. For all the energy tests, the image was encoded using a quality factor of 20 and about 6 to 7 packets were produced at the packetization stage.

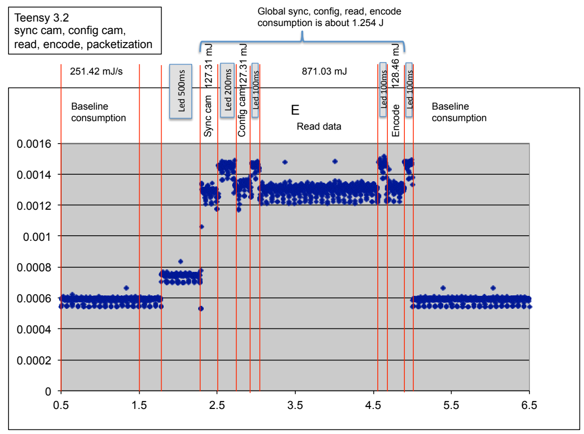


**Figure 8: EC for a cycle**

Fig. 8 shows an entire cycle of camera sync, camera config, data read, image encode and packetization, without transmission, when the CPU frequency is set to 96MHz. After removing the EC of the led, we found that an entire cycle for image acquisition, encoding and packetization consumes about 1.254J (our first prototype based on an Arduino Due needed 4.58J).

When active, the largest consumed energy part comes from polling the serial line to get the image data from the uCamII. At 96Mhz, the encoding process actually consumes 6.8 time less that amount of energy. The cost of periodic image change detection, without encoding and transmission is similar to the "Read data" cost. Therefore, we found that the "Global sync, config, read&compare" consumption is 1.125J (drawn current of about 115$mA$ during about 2s). Fig. 9 summarizes for various CPU frequencies the EC of the image sensor for reading uCamII data and encoding the image.

| | baseline (mJ/s) | baseline, hibernate (mJ/s) | read (mJ) | encode (mJ) | Read+encode (mJ) |
|---|---|---|---|---|---|
| 96MHz | 251.42 | 0.834 | 871.03 | 128.46 | 999.49 |
| 72MHz | 219.54 | 0.834 | 834.97 | 143.58 | 978.55 |
| 48MHz | 211.19 | 0.834 | 813.30 | 185.58 | 998.88 |
| 24MHz | 160.95 | 0.834 | 719.09 | 302.48 | 1021.57 |

**Figure 9: EC as CPU frequency is varied**

We also measured the baseline EC of the board without and with advanced power saving mechanism (advanced power saving mechanisms, deep sleep & hibernate mode, is provided by the `Snooze` library on the Teensy). Without power saving, the baseline EC is 251.42mJ/s at 96MHz (drawn current of about 50$mA$). In hibernate mode, the EC drops to about 0.834mJ/s (drawn current of about 167$\mu$A) regardless of the CPU frequency. We can then see that using the rated clock speed of 72MHz minimizes the global

EC. In a scenario where the image sensor wakes up every hour to perform an image change detection, without encoding nor transmitting, then it can theoretically run for about 453 days on 4 AA batteries (2500mAh).

Regarding the EC when transmitting, the drawn current is about 68mA (transmit power set to 14dBm). Therefore, in a scenario where the image sensor wakes up every hour to take an image (about 2s), encode it (about 290ms with Q=10) and transmit the image packets using mode 4 (about 8s for 4 packets), it can run for about 268 days on 4 AA batteries.

# 4. OUT-OF-THE-BOX SURVEILLANCE

## 4.1 Low-cost LoRa gateway

Within the H2020 WAZIUP project on low-cost IoT deployment in sub-Saharan Africa, we developed our low-cost LoRa gateway built around a Raspberry PI. Advanced data post-processing tasks are performed after the radio stage by using Unix redirection to post-processing scripts written in high-level language such as `Python`. Image data can be saved and decoded into BMP format. The gateway program can also run on an Arduino board equipped with a LoRa radio module acting as a transparent RF-USB-serial bridge, and connected to a Linux computer. Then data received from the serial port can be write back to Unix standard output and the `Python` post-processing script can be run unchanged on the Linux machine; see Fig. 10.

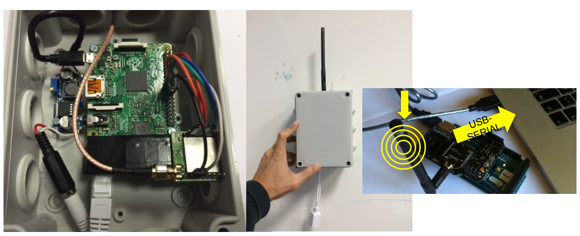

**Figure 10: Low-cost LoRa gateway**

## 4.2 Deploying the image sensor

The battery-operated image sensor is programmed to be completely autonomous once powered on. It can either run a periodic image transmission behavior for situation-awareness or environmental monitoring applications, or run a periodic image change detection task that will send images to the gateway in case of image change. We show in Fig. 11 a simple display application that dynamically discover new image nodes and assign for each node a column index in increasing order.

Here column index 0 (left-most) is for node 0x01. Node 0x02 has column index 1. As node 0x02 has 3 cameras, the image taken by each camera appears on a different line. The top line is for camera 0. We can also see how the display window indicates which image is the last one, and which image for a given image node is the last received one: the blue frame indicates for a given image node which image is the last received while the red frame (only one red frame at any time) is the last received image. Additionally, received images stored in a folder could be shared in real-time with a smartphone through a cloud application such as Dropbox™ as shown in figure 11 with the Dropbox application for iPhone™.

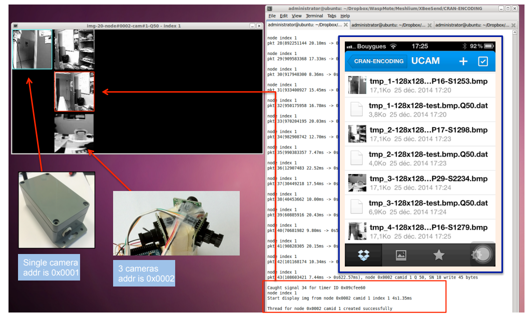


**Figure 11: Images displayed at a base station**

## 4.3 Operating the image sensor

The image sensor can also be further configured remotely by sending ASCII commands (from the base station or from any other Linux computer with a LoRa gateway). Commands start with "/@" and are separated by "#". The list of commands is shown below. For instance, The address of the destination gateway can be changed dynamically using command "D". By default image packets are sent to address 1. Note that several gateways can be used if available.

| Command | Camera command |
|---|---|
| T | immediately captures, encodes and transmits the image |
| C1 | selects camera 1 for next T/S/I/Q command |
| F30000 | sets inter-snapshot time to 30000ms, i.e. 30s, fps is then 1/30 |
| S0/S1 | starts a snapshot, make the comparison with the reference image and transmit the image if S1 |
| I | makes a snapshot and stores a new reference image |
| Z150 | sets the MSS size, default is 240 for LoRa |
| Q30 | sets the quality factor, default is 20 for LoRa |
| D8 | sets the 8-bit destination address to node 8. Default is 1. Use D0 for broadcast |
| Command | LoRa specific commands |
| LORAM1 | sets to Libelium LoRa mode 1 |
| LORAC12 | uses channel 12 |
| LORAPL/H/M | sets power to Low, High or Max |
| LORAA9 | sets node address to 9 |

**Figure 12: Configuring the image sensor**

The time between 2 image change detection or periodic image transmission can be changed with command "F" according to the surveillance profile. Another useful parameter is the quality factor. The lower the quality factor, the smaller the image size. In a multi-camera system, the quality factor for each camera can be defined. For instance command "/@C1#Q10#" will set for camera 1 a quality factor of 10. The maximum image payload can also be changed up to the maximum size of 240 bytes for LoRa radio with "Z" command. The image sensor can also send an image on demand with command "T". There are additionally 4 LoRa-specific commands to change the LoRa mode, the frequency channel, the transmit power and the image node's address.

# 5. IMAGE TRANSMISSION TESTS

## 5.1 Range tests

Many LoRa range tests have been performed by Semtech and others, and more than 20kms have been reported for LOS conditions. Semtech's tests also included transmission from pits. We show below some image transmission tests we did in our university area, close to city downtown in a dense urban area with many buildings (NLOS conditions) between the gateway located in front of the science faculty building and the image sensor. Both receiver and transmitter are at 1.5m height. Libelium LoRa mode 4 is used. We set the IMSS to 240 bytes and the quality factor to 20. Between 8 and 10 packets per image were generated. With mode 4, we could not received at 1010m as indicated in Fig. 13. 1 packet was lost in image 5. By using mode 1 which provides the longest range, we could increase the distance to 1.8km.

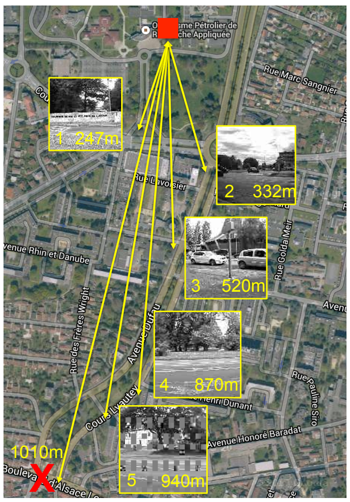


**Figure 13: Long-range image tests (NLOS)**

## 5.2 Improved channel access

A LoRa network is still basically a simple ALOHA system. Given the longer transmission time of an image packet, collisions have higher probability to occur than with traditional sensor where the data payload is much shorter. To increase network reliability when deploying multiple image sensors, a CSMA-like mechanism with SIFS/DIFS has been implemented using the Channel Activity Detection (CAD) function of the LoRa chip. A DIFS is defined as 3 SIFS and prior to begin an image transmission (first packet) a DIFS period free of activity should be observed. If "extended IFS" is activated then an additional random number of CAD followed by a DIFS is required. If RSSI checking is activated then the RSSI should be below -90dB for the packet to be transmitted. Once the first image packet has been successfully transmitted, all the remaining image packet transmissions use the SIFS timer to get higher transmission priority.

# 6. CONCLUSIONS

We presented an out-of-the-box low-cost long-range image surveillance system built from off-the-shelves components. The image sensor integrates an image change detection method and a packet loss-tolerant image compression technique that can run on very limited memory platforms. Encoded images can be transmitted with extreme long-range low-bandwidth radio such as the new LoRa technology by Semtech. Targeting periodic situation-awareness or visual detection of physical phenomenon in hard to access areas we showed that long-range transmission of encoded images is possible while keeping the energy consumption low for autonomous image sensors running on batteries. Further works will add advanced adaptation techniques both at the MAC and application layer to comply with radio activity time constraints.

# 7. ACKNOWLEDGMENTS

This work is support by the WAZIUP project with funding from the EU's H2020 research and innovation program under grant agreement No 687607.

# 8. REFERENCES

[1] M. Rahimi et al. "Cyclops: In situ image sensing and interpretation in wireless sensor networks," in *ACM SenSys*, 2005.

[2] S. Hengstler et al. "Mesheye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *IPSN*, 2007.

[3] P. Chen et al. "Citric: A low-bandwidth wireless camera network platform," in *ACM/IEEE ICDSC*, 2008.

[4] R. Kleihorst et al., "Camera mote with a high-performance parallel processor for real-time frame-based video processing," in *AVSS*, 2007.

[5] Evidence Embedding Technology, "Seed-eye board, a multimedia wsn device".

[6] Á. Rodríguez-Vázquez et al. "The Eye-RIS CMOS vision system," in Analog Circuit Design, H. Casier, M. Steyaert, and A. V. Roermund, Eds.

[7] W.-C. Feng et al., "Panoptes: Scalable low-power video sensor networking technologies," *ACM TMCCA*, vol. 1(2), May 2005.

[8] A. Rowe, D. Goel, and R. Rajkumar, "Firefly mosaic: A vision-enabled wireless sensor networking system," in *IEEE RTSS*, 2007

[9] Evidence Embedding Technology, "CMUcam: open source programmable embedded color vision sensors".

[10] S. Paniga et al. "Experimental evaluation of a video streaming system for wireless multimedia sensor networks," in *IEEE/IFIP MedHocNet*, 2011.

[11] C. Pham, "Low cost wireless image sensor networks for visual surveillance and intrusion detection applications," in *IEEE ICNSC*, 2015.

[12] S. Jeff McKeown, "LoRa$^{TM}$- a communications solution for emerging LPWAN, LPHAN and industrial sensing & IoT applications. http://cwbackoffice.co.uk/docs/jeff 20mckeown.pdf," accessed 13/01/2016.

[13] LoRa Alliance, "LoRaWAN specification, v1.01," Oct. 2015.

[14] Semtech, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver". rev.4-03/2015.

[15] PJRC, "Teensy31/32. https://www.pjrc.com/teensy/teensy31.html," accessed 22/05/2016.

[16] V. Lecuire, L. Makkaoui, and J.-M. Moureaux, "Fast zonal DCT for energy conservation in wireless image sensor networks," *Electronics Letters*, vol. 48(2), 2012.

[17] C. Duran-Faundez and V. Lecuire, "Error resilient image communication with chaotic pixel interleaving for wireless camera sensors," in *ACM RealWSN*, 2008.