LoRa-based Visual Monitoring Scheme for Agriculture IoT

Mookeun Ji*, Juyeon Yoon†, Jeongwoo Choo‡, Minki Jang§, Anthony Smith¶,

*School of Computer Science and Engineering, Chung-Ang University
†School of Computing, KAIST
‡School of Computer Science and Engineering, Kyungpook National University
§School of Global Media, Soongsil University
¶Computer and Information Technology, Purdue University

***Abstract*—LoRa is a low-power wide-range wireless networking technology suitable for low-rate long-range applications in the Internet of Things (IoT). For example in the agriculture industry, LoRa-based environmental sensing system enables farmers to remotely monitor the status of a large farm in near real-time. However, there had been only a few explorations to transfer multimedia data such as images or video using LoRa because of its low data rate and restricted bandwidth. To this end, we introduce a novel system to transmit continuous images taken from a camera on a static environment through LoRa. The key challenge is to reduce the amount of transmitted data while preserving the image quality and the quality of service delivered to the application. We develop a technique that splits image to grid patches, and transmits only the modified area of an image based on their dissimilarity measure. We implement and evaluate our scheme on a real LoRa device to show its performance and image quality.**

***Index Terms*—LoRa, Smart Agriculture, Visual Monitoring**

I. Introduction

With the increasing interest and demand in smart agriculture, there have been a lot of attempts to adopt Internet of Things (IoT) technologies on a farm [1]. IoT technology enables farmers to monitor the field conditions with connected sensors through the internet from anywhere. There are a lot of monitoring systems on various IoT platforms [2], but there have been only attempts of sending small environmental data such as the temperature or humidity. Although these systems are useful to farmers, it can be worthless if the crops are not in the right place or gone for the unexpected reasons, such as a natural disaster or intruders. Therefore, a surveillance system is an important factor in agriculture in order to secure and reduce the loss of productivity in crops. Providing visual monitoring to farmers can prevent crops from getting damaged by intruders and ensure the field conditions. However, it is challenging to deploy a real-time visual monitoring system to observe the whole farm in a wide agriculture area because connecting the farm to the wired network is impracticable. Therefore suitable wireless technology should be combined with a scheme of transmitting visual data.

In recent years, there has been a lot of experiments and attempts to send big multimedia data such as image, sound, and video using IEEE 802.15.4 technology [3], [4], [5], [6]. It is also called as Zigbee, which has been acknowledged as the standard for the commercial wireless sensor network (WSN) technology. As mentioned before, even though IEEE 802.15.4 is the widely known network, it has a drawback of short transmission range that is not appropriate for the vast agricultural area. Besides, there are numerous wireless network technologies that can be used in smart agriculture. Sigfox, Weightless-N, and LoRa can be examples of Low-Power Wide Area Networks (LPWAN). LoRa technology [7], which stands for ‘Long Range’, guarantees radio coverage over a very large area, but is known to be unsuitable for transmitting multimedia data because of its low data rate. This paper presents a visual monitoring scheme via LoRa. To overcome limited bandwidth, the images are divided into small grid patches and the changed patches are the only ones transmitted. Hence, the end user can monitor visual image continuously. Through experiments and simulations, the optimal dissimilarity measurement method to achieve exclusive quality of monitoring is discussed.

The rest of this paper is organized as follows. Section II gives an overview of LoRa technology, describes its core specification and introduces former works about transmitting multimedia data using LoRa network. Section III proposes our design scheme with a grid system, the prototype architecture and protocol implemented on the top of LoRa physical layer. The evaluation of system performance is discussed and analyzed in Section IV. Finally, the paper is concluded in Section V.

II. Related work

In this section, we provide brief overview of LoRa technology and former works on transmitting multimedia data via LoRa.

A. LoRa

Lora is a noteworthy technical solution used for the LPWAN design and protocols that is promoted by the LoRa Alliance.

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & communications Technology Promotion)(20170001000031001)

978-1-5386-7713-1/19/$31.00 ©2019 IEEE

Modulation for LoRa is based on spread-spectrum techniques and a variation of a chirp spread spectrum (CSS) [8] scheme. CSS scheme uses wideband linear frequency modulated pulses whose frequency increases or decreases based on the encoded information. This modulation is a proprietary technology by Semtech.

The LoRa physical layer consists of many parameters which can be configured into a number of different settings to offer a wide range of choices for ensuring a good quality link or consuming less energy. The most important parameters of the LoRa physical layer are the carrier frequency, spreading factor [9], bandwidth, transmission power, and coding rate. LoRa operates in unlicensed frequency bands: 915MHz in the USA, 868MHz in Europe, and 433MHz in Asia. Also, the technology provides throughput of 50 bps to 37 kbps under a variety of different parameters of modulation. The function of fragmentation and collection of valuable information is executed at the level of application.

B. *Multimedia via LoRa*

LoRa is generally considered unsuitable to transmit high bit rate data , such as image or voice, because of its limited bandwidth available for physical layer modulation. Despite this weakness, there have been a few attempts to transmit multimedia via LoRa.

In [10], various compression techniques are used for transmitting image or voice. For image transfer, widespread JPEG and JPEG2000 have been compared. It was shown that the format of JPEG2000 compression has less distortion after transmission; therefore it is more suitable for LoRa technology. Along with image transmission, A-law and U-law compression method were evaluated in voice transmission. This experiment indicated higher grade for English and Vietnamese by subjective evaluation, but lower grade when using Arabic.

Meanwhile, [11] presents a low-cost, low-power visual surveillance system based on image compression and change detection mechanism. This is implemented with simple-differencing of pixels. Also, its technique reduces the total number of transmission by omitting unchanged frames. By conducting a long-range image test, an image of about 2.4 Kbytes could be transferred up to 1.8 km. In the perspective of energy consumption, the image sensor node could last for about 258 days on 4 AA batteries, in a scenario where the image sensor wakes up every hour to take an image, encode, and transmit. Although this work is similar to our scheme in aspects of using change detection, it transmits the whole image when each small change is detected and simply compares each pixel. This is hard to be deployed for an outside environment with constantly changing brightness and moving background objects. Also, [12] shows feasibility of image transmission via LoRa modulation (*i.e.* LoRa physical layer) by demonstrating transmission of image with various ranges.

III. Proposed Scheme

This section features how to transmit images continuously through LoRa modulation. Prior researches have shown that transmitting full images sequentially is hard to achieve due to the property of LoRa, which includes limited bandwidth and data rate. However, we noticed that most agricultural areas are in a static environment where considerable changes are infrequent. This outlines full images are not required to be transmitted, which can reduce bandwidth usage on LoRa. We took advantage on the static property of agriculture by proposing a new monitoring scheme that divides each image into small grid patches. Each grid patches are only transmitted if any notable changes exist. This saves a lot of link budget to surveil static agriculture environment and enables better performance.

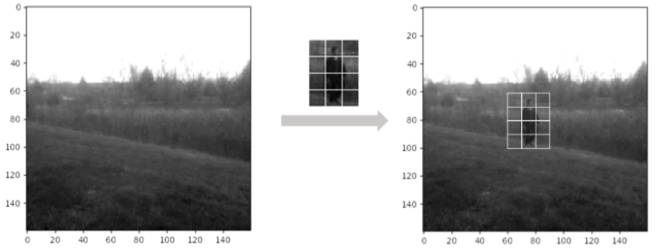


Fig. 1: Scheme overview. If end node transmit modified area of image, next frame can be obtained

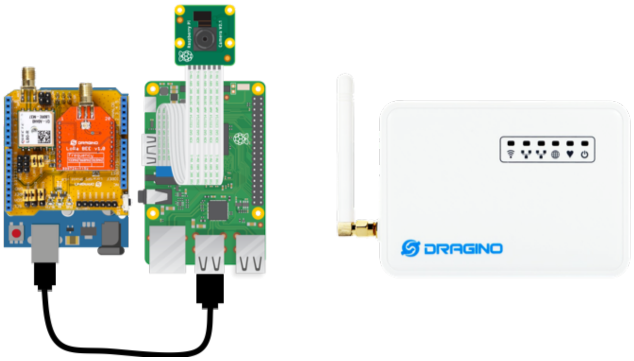


Fig. 2: LoRa end node (left), LoRa gateway (right)

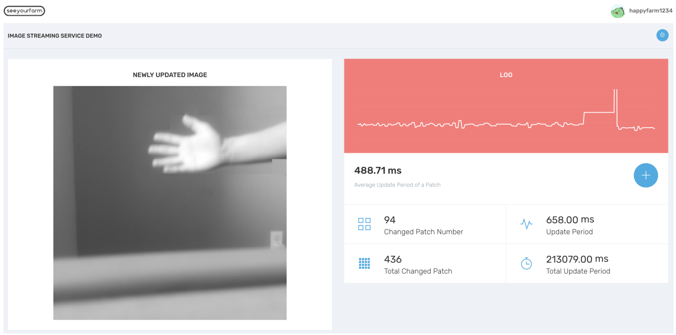


IMAGE STREAMING SERVICE DEMO

NEWLY UPDATED IMAGE

LOG

488.71 ms
Average Update Period of a Patch

94
Changed Patch Number

658.00 ms
Update Period

436
Total Changed Patch

213079.00 ms
Total Update Period

Fig. 3: Web page for streaming images

A. *Platform*

Our prototype consists of three parts: end node, gateway, and back-end server.

0	1	...	14	15
16	17	...	30	31
...	...		...	...
224	225	...	238	239
240	241	...	254	255

Grid size = 16

Fig. 4: Image divided by grid patches, indexed from top left to bottom right.

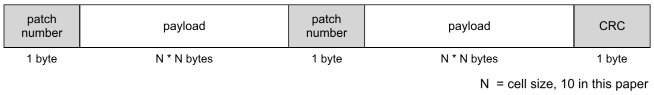


Fig. 5: LoRa packet structure

The end node is implemented by connecting Raspberry Pi 3 model B, which is comprised of Pi Camera 2 and Arduino Uno with LoRa shield. Data such as captured images and transmission parameters are exchanged through serial communication. This end node captures an image, divides the image into patches, and then applies the change detection algorithm between the patches from the previously taken image. It detects the distinct patches and finally sends the patches by using LoRa modulation.

Dragino LG-01 is a gateway that forwards the packets from the end node to the back-end server over TCP by using built-in script. Back-end server serves as TCP client for the gateway and web server that allows users to access the streamed image of the farm. For our prototype demonstration, we used AWS EC2 instance. It keeps listening on TCP port and notifies web server by HTTP request when new patch data arrives. Web server renders each patch image from the binary array and displays the whole image.

B. Scheme Design

The captured image, 160×160 8 bit grayscale, is divided into 256 grid patches that each patch is 10×10 images. (and those patches are indexed as Fig. 4)

Also, packet structure shown in Fig. 5 is used to deliver those patches from the end node to the gateway. An index of the patch being transmitted is stored in the first byte followed by the patch data. 2 patches are contained in a packet and at the end of the packet CRC-8 (8 bit cyclic redundancy check) is attached for error detection. The total packet size is 203 bytes in our prototype. Gateway responds to these packets with acknowledgment. End node retransmits up to MAX_RETRANSMISSION times unless it receives the acknowledgment within ACK_WAIT_TIME. MAX_RETRANSMISSION is set to 2 and ACK_WAIT_TIME is set to 3000ms.

End node has two different stages of operation: initial transmission stage and continuous transmission stage. As an end node wakes up, it turns into an initial transmission stage.

1) Initial Transmission Stage: In this stage, the end node establishes connection with Pi camera and Arduino. And then, the end node captures an image and sends all patches to the gateway to make sure that the backend server has a full image. Moreover, it saves the image for further similarity comparison. The gateway responds with an acknowledgment and forwards the received packets to the back-end server over TCP. Every time the back-end server receives a packet from the gateway, it renders PNG image using Javascript PNG library and generates base64 encoded string from the image. The patches compose the full image again and display it on a web application. Although it takes a relatively long time, this is vital as it is a prerequisite step of our system because transmitting the whole image after this initial stage is not considered. Once all patches are sent to the back-end server, initial transmission state is terminated and end node enters continuous transmission stage.

2) Continuous Transmission Stage: At this stage, an end node keeps sending the changed patch by repeating the following cycle. First, the end node captures an image and calculates the dissimilarity (or distance, error) between the current patch and a patch in the previous image that has same index (*i.e.* end node calculates error between two patches at the same position, one from the current and one from the previous image). If the calculated dissimilarity is greater than the threshold, the end node sends the patch to the gateway. Similar to initial transmission stage, the gateway forwards each received packet to the backend server and the server updates the corresponding patch on the web application immediately.

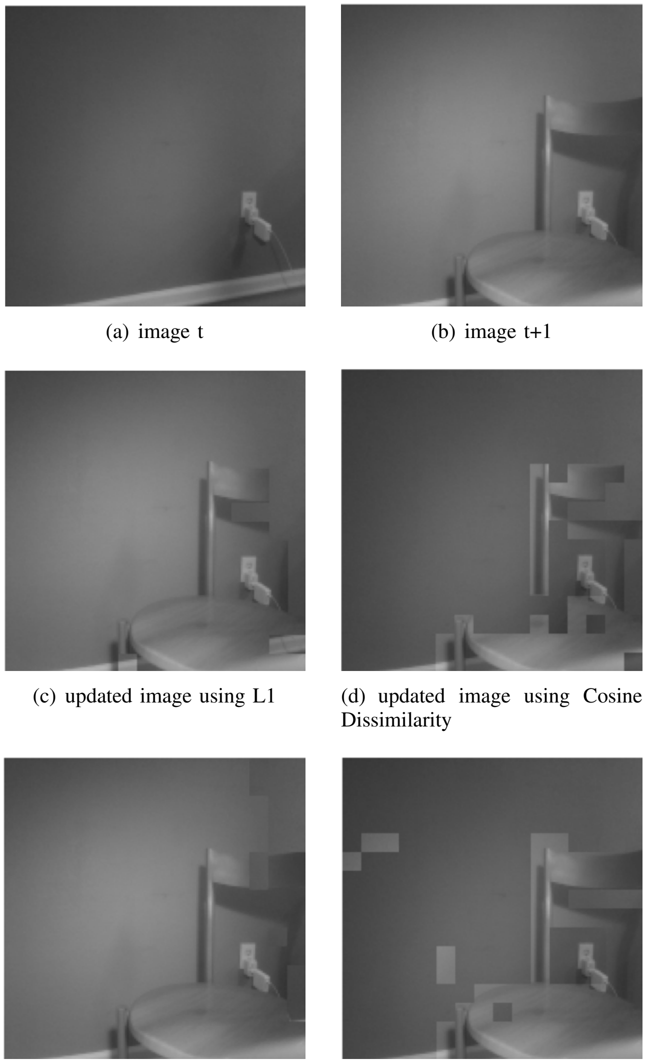
C. Dissimilarity Measurement

Dissimilarity measurement method between two patches and threshold is a key to our scheme. This is highly influential in performance because it determines how many patches to be transmitted.

To choose the best dissimilarity measurement for our scheme, we compared four different measurement methods in our system: L1 distance (Manhattan distance), L2 distance (Euclidean distance) [13], Cosine Dissimilarity and Structural Dissimilarity (DSSIM) [14]. For two vectors p and q in n-dimensional vector space, four dissimilarity metrics are as follows:

$$L1 = \sum_{i=1}^{n} |p_i - q_i| \tag{1}$$

$$L2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2}$$


(a) image t

(b) image t+1

(c) updated image using L1

(d) updated image using Cosine Dissimilarity

(e) updated image using L2

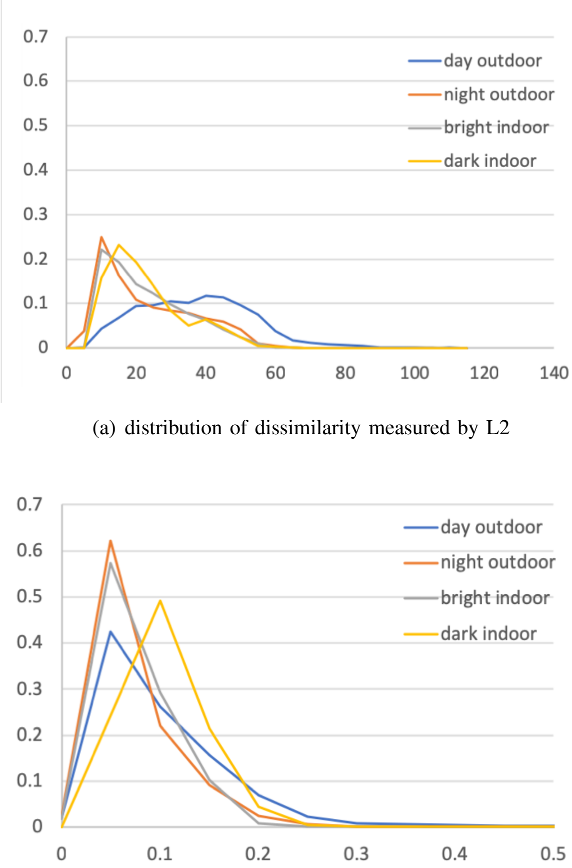
(f) updated image using DSSIM

Fig. 6: comparison of four dissimilarity measurement methods. (a) and (b) are spontaneously captured image. (c), (d), (e) and (f) are updated image by using L1, Cosine Dissimilarity, L2 and DSSIM

$$Cosine\ Dissimilarity = \frac{1-\cos\theta}{2} = \frac{1-\frac{p\cdot q}{|p||q|}}{2} \tag{3}$$

$$DSSIM = \frac{1-SSIM}{2} = \frac{1-\frac{(2\mu_p\mu_q+C_1)(2\sigma_{pq}+C_2)}{(\mu_p^2+\mu_q^2+C_1)(\sigma_p^2+\sigma_q^2+C_2)}}{2} \tag{4}$$

We observed each metric by applying it to our system. We noticed that the L1 distance and the cosine dissimilarity are not appropriate to identify changed patches. L1 distance is extremely sensitive to shifting and brightness changes. For example, it frequently detects changes on the patches in the background, as the sunlight changes. Also, when the wind is blowing, the static objects such as trees were often updated. Cosine dissimilarity often ignores significant changes in brightness as Fig. 6.



(a) distribution of dissimilarity measured by L2

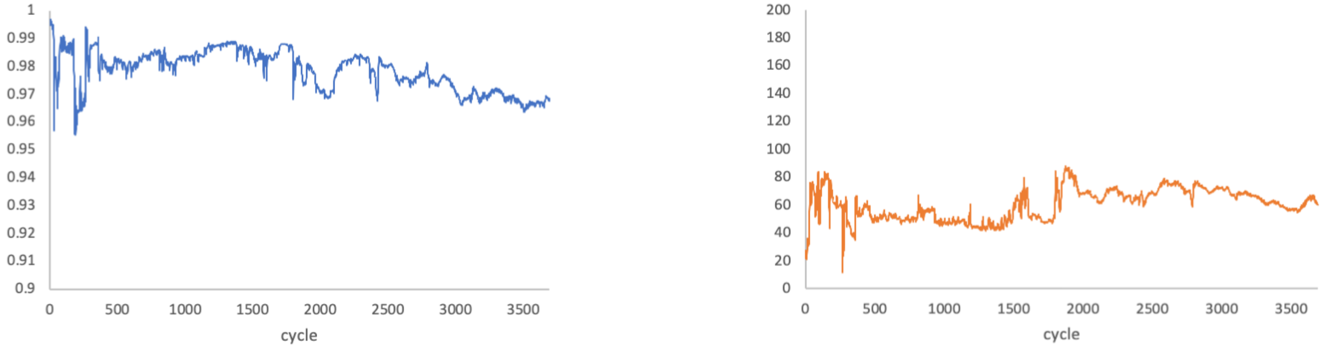
(b) distribution of dissimilarity measured by DSSIM

Fig. 7: distribution of dissimilarity measured by L2 and DSSIM on various environments.

L2 distance and Structural Dissimilarity (DSSIM), which are commonly used for image comparison, showed better performance than the above two on detecting the changed portion of the image. L2 distance is a basic dissimilarity measurement method for general data, whereas DSSIM is designed to detect structural changes on image.

To compare L2 and DSSIM in the aspect of suitability for our system, we collected more data using these two methods on a controlled environment, which is very static. Changed patches and static patches need to be distinguished by the adequate threshold value. The value can be set slightly higher than the dissimilarity of static patches. So we decided to observe the range of the dissimilarity values, expected to be relatively low, on the static patches. We chose various environments, both outdoor and indoor, day and night, for an hour to see if the distribution may vary along with sunlight or illumination. For the two methods, the distribution of the dissimilarity values for all patches is shown as Fig. 7.

Fig. 7 show that DSSIM has a much more stable background dissimilarity than L2 distance on different environments. This



(a) SSIM over cycles

(b) MSE over cycles

Fig. 8: Streaming quality measured by (a) SSIM and (b) MSE over cycles.

means that the DSSIM can detect changed patches easily from the same threshold even environment changes. Also, DSSIM was tolerant to non-structural changes in background. Hence, we adapted DSSIM to our scheme as the dissimilarity measurement method and the threshold is set to 0.15 for the experiments in this paper.

IV. Evaluation

In our work, a detailed evaluation of study that demonstrates our scheme is in two aspects: performance and quality of streaming. We have conducted a two hours experiment on the outdoor environment that is similar to agriculture surroundings. Then, we analyzed the collected log.

A. *Performance*

# of total cycles	3,695
# of cycles that skipped transmission	2,820
# of total image patches	945,920
# of transmitted image patches	3,457

TABLE I: the number of cycles and image patches

First, we measured the number of overall image patches of the captured images by the end-node and the number of transmitted image patches. As a result, 3457 patches are considered as changed during 3695 cycles and transmitted as shown as Table I.

The end node did not transmit any patches for 2820 cycles because no changed area was detected. Thus, only 24% of the total cycles actually performed transmission.

The average of throughput is measured at 2.212 Kbps based on the collected data. The elapsed time for an initial transmission stage was total 127.5 seconds. Since the device enters continuous transmission stage, we analyzed the elapsed time per cycle when a notable change is detected in sight of a camera. In our experiment, we observed average 14.47 patches were updated per cycle and it took average 5.22 seconds while the number of transmitted patches varies from 10 to 30. The average latency of updating a new patch was 0.361 seconds.

B. *Quality of Streaming*

Due to the property of transmitting changed image patches, partial loss of the image is inevitable. In addition, there is a possibility that the old patches remains for a long time over cycles because the only patches that end node recognized change are updated. Therefore, it is necessary to evaluate the quality of image and assure that it is maintained. We compared the image displayed on the web application with the captured image using SSIM and Mean Square Error (MSE) as time goes on.

As shown in Fig. 8, SSIM is maintained over 0.955 for 3695 cycles during 7230 sec. While MSE is kept under value of 90 for the same period. This result indicates that the user-side image maintains over 95% similarity with the original image with low mean square error.

V. Conclusion

This paper presents a new visual monitoring technique specialized for the agricultural applications using LoRa technology. By sending image patches of only the modified region of an image, we successfully reduced bandwidth usage and achieved noteworthy performance enhancement on user-perceived latency. Moreover, we adapted the existing dissimilarity measurement method, DSSIM, to our system and continuously maintained a high quality of streaming. Through implementation and evaluation, we showed the feasibility of real-time visual surveillance applications on a wide-area agricultural environments. In future extensions, our scheme will be deployed on an actual farm for evaluation on battery efficiency and performance measurements according to transmission distances. Future work will focus on taking advantage of LoRaWAN which is a widely used MAC protocol for LoRa. At a later stage, our scheme will also integrate with existing sensors to deploy in the farm sector.

References

[1] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/9/1466

[2] L. Xiao and L. Guo, "The realization of precision agriculture monitoring system based on wireless sensor network," *2010 International Conference on Computer and Communication Technologies in Agriculture Engineering*, vol. 3, pp. 89–92, June 2010.

[3] R. Kirichek, M. Makolkina, J. Sene, and V. Takhtuev, "Estimation quality parameters of transferring image and voice data over zigbee in transparent mode," *Distributed Computer and Communication Networks*, pp. 260–267, 2016.

[4] G. Pekhteryev, Z. Sahinoglu, P. Orlik, and G. Bhatti, "Image transmission over ieee 802.15.4 and zigbee networks," *2005 IEEE International Symposium on Circuits and Systems*, pp. 3539–3542 Vol. 4, May 2005.

[5] J. Paek, J. Hicks, S. Coe, and R. Govindan, "Image-based environmental monitoring sensor application using an embedded wireless sensor network," *Sensors*, vol. 14, no. 9, pp. 15 981–16 002, 2014.

[6] J. Paek and J. Ko, "K-means clustering-based data compression scheme for wireless imaging sensor networks," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2652–2662, Dec 2017.

[7] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, April 2017.

[8] A. Berni and W. Gregg, "On the utility of chirp modulation for digital signaling," *IEEE Transactions on Communications*, vol. 21, no. 6, pp. 748–751, June 1973.

[9] A. Waret, M. Kaneko, A. Guitton, and N. E. Rachkidy, "Lora throughput analysis with imperfect spreading factor orthogonality," *CoRR*, vol. abs/1803.06534, 2018. [Online]. Available: http://arxiv.org/abs/1803.06534

[10] R. Kirichek, V.-D. Pham, A. Kolechkin, M. Al-Bahri, e. O. Paramonov, Alexander", S. Andreev, S. Balandin, and Y. Koucheryavy, "Transfer of multimedia data via lora," *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 708–720, 2017.

[11] C. Pham, "Low-cost, low-power and long-range image sensor for visual surveillance," *ACM SmartObjects '16*, pp. 35–40, 2016.

[12] A. H. Jebril, A. Sali, A. Ismail, and M. F. A. Rasid, "Overcoming limitations of lora physical layer in image transmission," *Sensors*, vol. 18, no. 10, 2018.

[13] J. Li and B.-L. Lu, "An adaptive image euclidean distance," *Pattern Recognition*, vol. 42, no. 3, pp. 349–357, 2009.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.