



A low-rate encoder for image transmission using LoRa communication modules

Kevin Guerra¹ · Jhon Casavilca¹ · Samuel Huamán² · Lidia López² · Alonso Sanchez¹ · Guillermo Kemper¹

Received: 17 May 2022 / Accepted: 17 August 2022 / Published online: 19 September 2022

© The Author(s), under exclusive licence to Bharati Vidyapeeth's Institute of Computer Applications and Management 2022

Abstract The present work proposes an encoder for image transmission via LoRa communication modules. These enable long-range, low-power transmission schemes and are ideal for monitoring in places with no mobile network connectivity. Nonetheless, this technology has a low transmission bitrate, which limits its use to high bandwidth applications. The state-of-the-art has numerous image encoders, but few achieve an adequate balance between image quality, compression, sequential decoding, and computational complexity. The proposed encoder uses the YCoCg color model and chromatic subsampling followed by wavelet subband decomposition, which extracts relevant subbands in the image to then reconstruct it sequentially. Each subband is quantized independently and then enters an adaptive entropic encoder. This encoder is compared to the JPEG2000 encoder using the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) quality metrics. Results show that the proposal obtains a reconstructed image quality close to that of JPEG2000 with a higher compression rate. Moreover, it improves the transmission time of images through a LoRa link by 99.09%.

Keywords Image compression · LoRa module · Huffman coding · Wavelets transform · Subband decomposition · YCoCg

✉ Guillermo Kemper
guillermo.kemper@upc.pe

¹ Faculty of Engineering, School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas, Av. Prolongación Primavera 2390, Monterrico, Santiago de Surco 15023, Lima, Peru

² National Institute of Research and Training at Telecommunications (INICTEL-UNI), Universidad Nacional de Ingeniería, Av. Túpac Amaru 210, Rímac 15333, Lima, Peru

1 Introduction

Nowadays, LoRa technology is present in many Internet of Things (IoT) applications which use wireless sensor networks [1, 2]. Compared to other technologies such as Wi-Fi, Zigbee, GSM or Bluetooth, LoRa has a longer range, lower energy consumption and can be used in unlicensed frequency bands. Nonetheless, due to its low bitrate, it is restricted to telemetry applications [3, 4]. It cannot be employed in monitoring systems with greater capabilities, such as those which transmit more information in short time bursts via images [5]. In this context, the proposed image encoder is designed to be used in low bitrate transmission modules such that they can handle digital image transmission, thus enhancing their monitoring capabilities. This work presents a use case in which the proposed encoder, implemented in a single-board computer, transmits images of hydrometric rulers. These are measuring tools to indicate relevant water levels in remote regions. Quick visualization of rulers' images aid monitoring centers in remote regions with flooding hazards due to seasonal El Niño events. The proposed encoder achieves this feature by using a progressive transmission, where lower-resolution images are quickly visualized first in the receiver and the visualization is steadily improved in subsequent moments. Thus, the encoder design and chosen transmission mode aim for a minimum latency in the image transmission when compared to other encoders such as the JPEG2000 while maintaining the quality of the reconstructed image.

2 Related work

This section presents two groups of selected articles which focus on image transmission in low bandwidth

communications systems. On one hand, Table 1 describes the first group, which contains proposals that use low bandwidth IoT modules. On the other hand, Table 2 presents the second group, which has articles that seek to improve information transmission efficiency considering a wider range of applications and not necessarily IoT settings.

The image transmission methods in Table 1 mostly handle only low resolution, grayscale images. Clearly, these images cannot carry information otherwise available in the color bands. The transmission schemes in Table 2 can work with higher resolutions and color images efficiently, but neither group of proposals evaluate implementations on embedded devices. These devices require balancing an acceptable quality of the uncompressed image, decent image visualization delays and manageable computational loads.

With these considerations, the proposed encoder presents a successful alternative which balances the use of color images, high compression rates and reduced transmission times with progressive decoding and a low complexity. The encoder slightly outperforms the JPEG2000 encoder, which serves as a benchmark to highlight this work's contribution.

3 Proposed methodology

The proposed encoder uses wavelet subband decomposition and reconstruction with multiresolution analysis, which gives it the unique characteristic of being able to progressively reconstruct the image while each subband is transmitted and decoded. Thus, it achieves comparable quality values [validated via Evaluating the peak signal-to-noise ratio (PSNR) at structural similarity index (SSIM) fixed compression rates] as JPEG2000 at slightly better compression rates, for RGB images. Moreover, it enables a rapid first, low-quality visualization of the decoded image, whose quality then progressively increases as the reception and decoding of the subsequent subbands takes place. The encoder and decoder were implemented in LoRa modules, where image transmission times were evaluated with and without the encoding. A maximum transmission time reduction of 99.09% was achieved. For these reasons, the proposed image encoder and decoder constitute the largest contribution in this work.

The following sections describe the encoding stage, data transmission stage and decoding stage.

3.1 Design of the proposed encoder

Figures 1 and 2 summarize the proposed image compression scheme, presenting the image encoder and decoder respectively. This section details the encoding and decoding steps, which aim to balance an acceptable image quality and a low computational load.

The proposed encoder and decoder were implemented in a Raspberry Pi 3B + single-board computer [16]. Image acquisition was performed either via a webcam or a Raspberry Pi Camera Module 2, using the RGB color format in both cases.

The encoding process is described below:

Step 1 Acquire the image in its primary RGB components and convert to YCoCg color space to separate the luminance and chrominance components, with a digital resolution of M rows and N columns. Each component can be expressed by $I_Y(x, y)$, $I_{Co}(x, y)$ and $I_{Cg}(x, y)$ [17].

Step 2 The Co and Cg chrominance components were subsampled with the 4:2:0 color format [18]. The resulting subsampled chrominance components are $I_{SCo}(x, y)$ and $I_{SCg}(x, y)$.

Step 3 The luminance component is processed by 3-level deep tree, while the chrominance components are processed by a 4-level deep tree [19]. Evaluating PSNR and SSIM of the different subbands between the original and reconstructed images, the approximation subband $I_{Y,3,1}(x, y)$, the horizontal detail subband $I_{Y,3,2}(x, y)$ and the vertical detail subband $I_{Y,3,3}(x, y)$ are chosen for the luminance component. The rest of the subbands are never encoded nor transmitted.

For both chrominance subbands, only the approximation subbands ($I_{SCo,4,1}(x, y)$ and $I_{SCg,4,1}(x, y)$) are chosen and the rest subbands are neither encoded nor transmitted. The different subbands will subsequently be expressed as $I_{subband}(x, y)$. This notation will be employed in the rest of the article.

Different families of wavelets were analyzed and compared to choose low-pass and high-pass filters [20]. A total of 35 hydrometric ruler images were used in the evaluation process, which considered computational loads and PSNR and SSIM metrics [21]. Table 3 shows the comparison of the filters' impulse response length in each family. This parameter is directly related to the computational load of the convolutions performed during the subband decomposition and reconstruction.

The biorthogonal family has the best trade-off between quality and computational load, resulting in choosing the bior3.9 wavelet family for this encoder.

Step 4 A uniform scale factor quantizer is used which is expressed in Eq. (1) [22]. Each quantizer's resolution was chosen considering the human visual tolerance to distortion, through testing and comparison of subjective quality, average PSNR and SSIM of diverse images. Finally, luminance subbands were

Table 1 List of similar works using low-bandwidth IoT modules

References	Approach	Application	Strengths	Weaknesses
Ruslan et al. [6]	The method uses the JPEG2000 compression algorithm	Multimedia transmission from a UAV via LoRa modules	Achieves acceptable image and audio quality in the transmission	Does not handle color images. Images can only be 160×120 pixels Image visualization is not progressive The maximum tested transmission distance is 5 km
Ching-Chuan et al. [7]	The algorithm uses Webp compression and a Base64 encoder	Efficient image transmission using LoRa modules	Includes an algorithm which re-sends lost packets	Only works with low resolution, 200×150 pixels images The maximum tested transmission distance is 1.5 km
Tonghao et al. [8]	Uses the JPEG image compression algorithm	Image transmission via LoRa modules for agricultural monitoring	Implements MPLR protocol to reduce image transmission time	The maximum tested transmission distance is of 300 m Paper focuses on reducing transmission time, so there is no image quality evaluation
Mookeu et al. [9]	The method tracks intensity changes in pixel fragments	Visual monitoring using LoRa, for agricultural applications	Only pixel fragments with intensity variations are transmitted	Does not handle color images Tests were done only in controlled settings with low ambient variability The method introduces distortion to the images
Haron et al. [10]	FFT applied to image compression	Image compression for industry 4.0 applications	Variable compression is applied, considering 4 different sizes	Only handles 512×512 pixel grayscale images. The maximum compression factor is 1.66
Pan et al. [11]	Uses deep neural networks and a Huffman encoder	Image transmission for IoT systems	The use of neural networks makes the system robust against packet loss	Since the neural network has a high computational load for embedded system use, it is limited to 224×224 pixel images

Table 2 List of similar works with general purpose

References	Approach	Application	Strengths	Weaknesses
Ahmad et al. [12]	Uses the dual tree complex compression algorithm and the EBCOT encoder	Image and multimedia applications	Achieves a 0.8 PSNR and 0.9 dB improvement over JPEG2000	Has a higher computational load which limits its use in embedded systems
Al-Azawi et al. [13]	Uses AMBTC encoding algorithms and a bit plane compressor	Image compression	Reduces the number of arithmetic operations to achieve a lower computational load	The bitrate reduction heavily reduces the PSNR. For example, a 0.1 bpp (bit per pixel) bitrate achieves less than 5 dB PSNR
Prades et al. [14]	Uses adaptive module-PCM	Video applications	Low complexity encoder with adaptable computational load	Although it achieves a lower computational load, the reconstructed quality is lower than in a JPEG encoder
Telles et al. [15]	Combines subband decomposition with predictive multispectral analysis	Image transmission for small satellites	Higher PSNR and SSIM than SPIHT, EZW and STW. For a compression factor of 20, the method achieves a PSNR of 30 dB and a SSIM of 0.8	Higher compression rates considerably reduce the PSNR. A rate of 88 achieves a PSNR of 11 dB

quantized with $r = 5$ bits per sample and chrominance subbands, with $r = 4$ bits per sample.

$$I_{subband}^Q(x, y) = \text{round}\left(\frac{I'_{subband}(x, y)}{fe_{subband}} \cdot (2^r - 1)\right) \quad (1)$$

where $I'_{subband}(x, y) = I_{subband}(x, y) - Imin_{subband}$ which ensures that the subbands only have positive values.

Every scale factor and minimum scaling value for each luminance and chrominance subband must also be quantized and encoded. These numbers were quantized with 24 bits to avoid hindering the obtained image quality. On the other hand, the minimum scaling values in every subband are always negative, so that quantization is done to the absolute value. The scaling factor and minimum scaling value are expressed in Eqs. (2) and (3) respectively.

$$fe_{subband}^Q = \text{round}(fe_{subband} \cdot 10^5) \quad (2)$$

$$Imin_{subband}^Q = \text{round}(|Imin_{subband}| \cdot 10^5) \quad (3)$$

Step 5 The quantized subbands are further coded with a Huffman entropy coder [23]. Static codebooks were generated from the subbands evaluated in 100 images. Two types of distributions were predominantly obtained: those with occurrences at all numerical values for a given bit quantization depth; and distributions with missing values. The second type of distribution generated codebooks with an average of 5% higher coding gain. The total number of codebooks is 44, so a 6-bit number ("Number codebook") is transmitted in the packet header to indicate which codebook encodes a particular subband.

Before Huffman coding is applied to a subband, an algorithm determines which option occupies the least number of bits in the transmission packet. Whether or not Huffman coding has been used is indicated in the packet header by the 2-bit number "Huffman Enable". If it is "00", it indicates that no coding has been used; if it is "11", it means that Huffman coding has been applied. Finally, every Huffman encoded subband will subsequently be denoted by $I_{subband}^{QH}(x, y)$.

3.2 Data transmission and reception

The images and encoded data are transmitted in the LoRa packet payload (see Fig. 3), which contains a maximum of 146 bytes. The explicit packet type is used which allows payloads of variable size [24]. The "Data type identifier" field (1 byte) indicates whether the packet transmits header data

Fig. 1 Block diagram of the proposed encoder

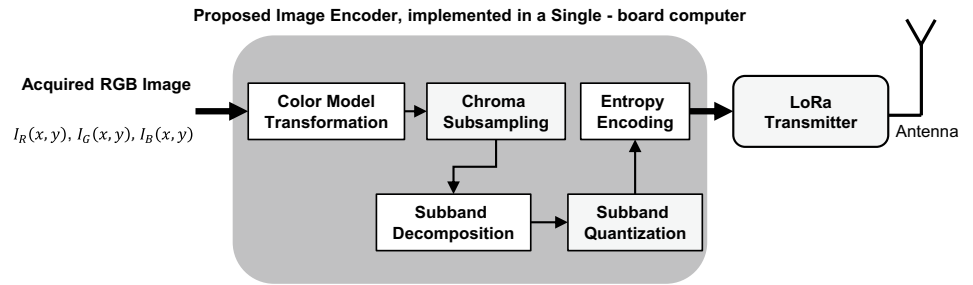


Fig. 2 Block diagram of the proposed decoder

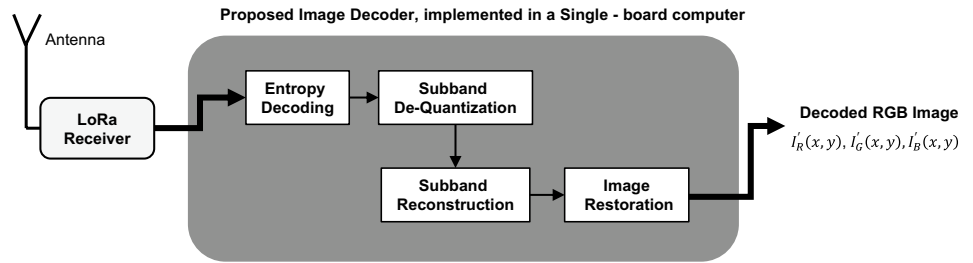


Table 3 Evaluation of wavelet filters on hydrometric ruler images

Wavelet filter	Number of coefficients	PSNR (dB)	SSIM
Daubechies 40	80	25.2	0.71
Symlet 20	40	25.3	0.74
Coiflet 5	30	25.2	0.74
Biorthogonal 3.9	20	25.2	0.75

or a coded subband. The "Counter" field (1 byte) contains the count of packets sent, and the "Information" field (up to 144 bytes) contains the header or coded subband data. When transmitting header data, the "Data type identifier" has a value of 0; when transmitting a coded subband, it has a value of 255.

Header data is sent three times in three consecutive LoRa packets. This redundancy aids in avoiding reception errors for header data, which could have a large negative impact on

Fig. 3 Explicit LoRa packet format and data structure carried in the payload

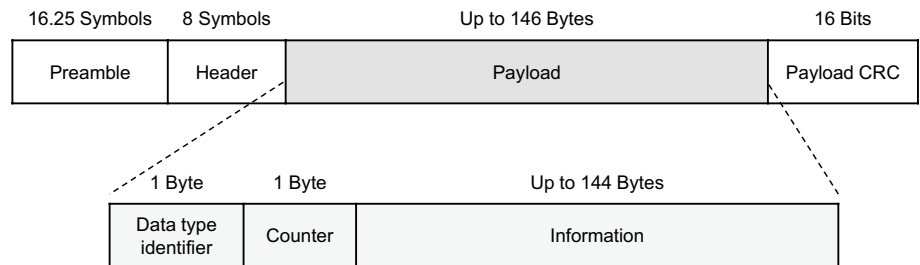


Fig. 4 Header data frame

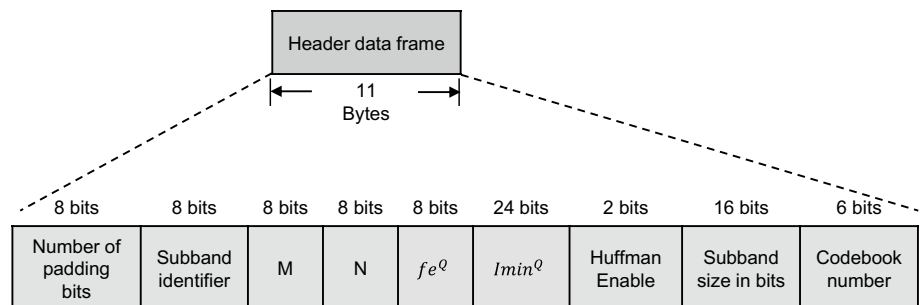


Fig. 5 Encoded subband data frame

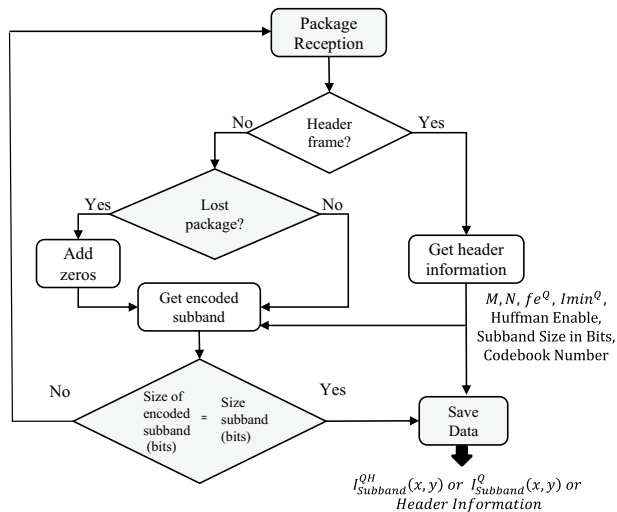
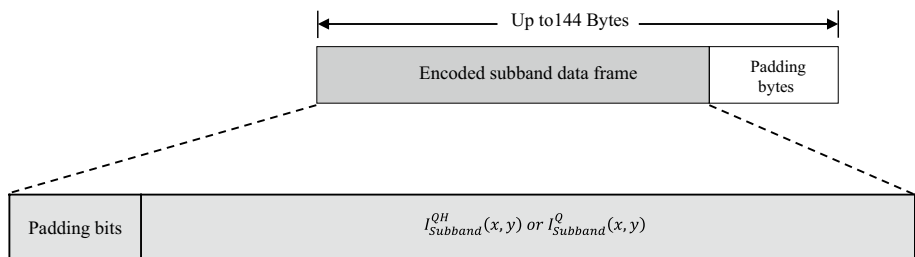


Fig. 6 Data reception process

image reconstruction. Figure 4 shows the header data frame sent in the “Information” field.

The 8-bit “Number padding bits” field indicates the number of bits added to a field that contains an encoded subband to always have an integer number of bytes. The 8-bit “Subband identifier” field indicates which subband is being encoded. Each of the 5 encoded subbands has a unique 8-bit identifier code. The “M” and “N” 8-bit fields correspond to the number of rows and columns of the matrix that represents the subband. The 8-bit fe^Q corresponds to the quantized scalefactor. The 24-bit $Imin^Q$ is the minimum scaling value used during subband quantization. The 2-bit “Huffman Enable” indicates whether Huffman encoding was applied. The 16-bit “Subband Size in bits” store the number of bits that are used to store the codified subband. Finally, the 6 bit “Codebook number” field indicates the codebook used for Huffman encoding, if applicable.

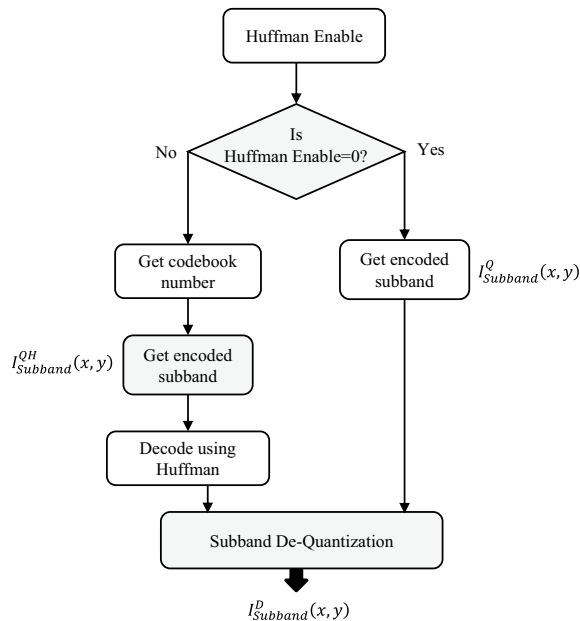


Fig. 7 Decoding and subband de-quantization process

Figure 5 shows the data frame of the encoded subband data, sent in the “Information” field in Fig. 3. “Padding bits” holds the 0-valued bits required so that the frame has an integer number of bytes. Finally, $I_{subband}^{QH}(x,y)$ or $I_{subband}^Q(x,y)$ represent the subband, encoded with or without Huffman respectively.

For the reception process, the binary data is reconstructed from the header and the received subband frames. Figure 6 shows a flowchart of this process. The frames contain coded subbands, the data is concatenated until a complete subband is obtained. If a subband packet is lost,

zeros are concatenated to the subband so that the image can still be reconstructed.

3.3 Image reconstruction and restoration

The header data serves to reconstruct the 2-D matrices in each decoded subband, and with these to progressively reconstruct the image. Figure 7 illustrates the flowchart of the employed decoding [23] and de-quantization process [22]. This process may or may not utilize Huffman binary decoding, according to the header data. The process is described below:

Step 1 The de-quantization process is executed. Equations (4)–(6) describe the de-quantization, scale-factor $fe_{subband}^D$ decoding and $Imin_{subband}^D$ value computation for each subband, respectively.

$$I_{subband}^D(x, y) = \left(\frac{I_{subband}^Q(x, y)}{2^r - 1} \cdot fe_{subband}^D \right) - Imin_{subband}^D \quad (4)$$

$$fe_{subband}^D = \frac{fe_{subband}^Q}{10^5} \quad (5)$$

$$Imin_{subband}^D = \frac{Imin_{subband}^Q}{10^5} \quad (6)$$

Step 2 The components of the YCoCg color model are reconstructed from the subbands using the wavelet family and reconstruction trees described in the Sect. 3.1 [19]. The reconstruction trees employ the inverse wavelet transform, where the subbands that were neither encoded nor sent are set to 0 during the reconstruction. The resulting components of the reconstruction are $I_Y^D(x, y)$, $I_{SCo}^D(x, y)$ and $I_{SCg}^D(x, y)$.

Step 3 This last stage restores the image from the reconstructed components, YCoCg. First, it is necessary to interpolate the chrominance components, since they were undersampled with a 4:2:0 scheme in the encoder. Due to their computational burden, zero-order interpolation was applied [25]. The resulting chrominance components are $I_Y^{D'}(x, y)$, $I_{SCo}^{D'}(x, y)$ and $I_{SCg}^{D'}(x, y)$.

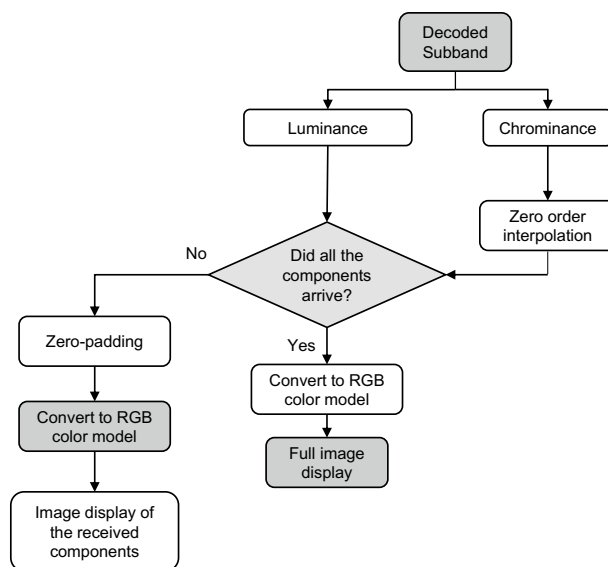


Fig. 8 Image restoration process

With the recovered chrominance components, the YCoCg image is converted to the RGB primary color space [17], where the numerical values are rounded and scaled to a range from 0 to 255. Finally, the reconstructed RGB primary components $I_R'(x, y)$, $I_G'(x, y)$ and $I_B'(x, y)$ are shaped into a 3D array for displaying in any graphical user interface. The complete process is summarized in Fig. 8.

4 Experimental results and discussion

The PSNR (Peak Signal-to-noise Ratio) and SSIM (structural similarity index) metrics were used to validate the proposed encoder [21]. Both metrics are always related to the obtained compression factor, such that different image encoders can be compared at similar compression levels. Firstly, the PSNR measures the ratio of the maximum signal power to average encoding noise, in dB (decibels), and is useful for low-frequency, uniform image regions [26]. It is computed as:

$$PSNR(dB) = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (7)$$

where

$$MSE = \frac{1}{3 \times M \times N} \left(\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_R'(x, y) - I_R(x, y)]^2 + \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_G'(x, y) - I_G(x, y)]^2 + \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_B'(x, y) - I_B(x, y)]^2 \right) \quad (8)$$

Next, the SSIM is an objective evaluation metric ranging from 1 to 0, where 0 translates to a null similarity. The metric is useful for high-frequency, highly varying image regions [27]. The SSIM has a better performance with respect to subjective metrics such as the Mean Opinion Score (MOS). The SSIM computation is done between an original image, $F(x, y)$, and a reconstructed image, $V(x, y)$. Both images must be monochrome. The metric is computed for image blocks or regions, and it is then averaged to obtain the mean SSIM (MSSIM) in each image block l [27]:

$$MSSIM_{FV} = \frac{1}{L} \sum_{l=0}^{L-1} SSIM_{FV,l} \tag{9}$$

where

$$SSIM_{FV,l} = \frac{(2 \times \mu_{F,l} \times \mu_{V,l} + C_1) \times (2 \times \sigma_{FV,l} + C_2)}{(\mu_{F,l}^2 + \mu_{V,l}^2 + C_1) \times (\sigma_{F,l}^2 + \sigma_{V,l}^2 + C_2)} \tag{10}$$

for $l = 0, 1, 2, \dots, L - 1$

$$\mu_{F,l} = \sum_{x=0}^{P-1} \sum_{y=0}^{P-1} w(x, y) F_l(x, y) \tag{11}$$

$$\mu_{V,l} = \sum_{x=0}^{P-1} \sum_{y=0}^{P-1} w(x, y) V_l(x, y) \tag{12}$$

$$\sigma_{F,l}^2 = \sum_{x=0}^{P-1} \sum_{y=0}^{P-1} w(x, y) (F_l(x, y) - \mu_{F,l})^2 \tag{13}$$

$$\sigma_{V,l}^2 = \sum_{x=0}^{P-1} \sum_{y=0}^{P-1} w(x, y) (V_l(x, y) - \mu_{V,l})^2 \tag{14}$$

$$\sigma_{FV,l} = \sum_{x=0}^{P-1} \sum_{y=0}^{P-1} w(x, y) (F_l(x, y) - \mu_{F,l}) (V_l(x, y) - \mu_{V,l}) \tag{15}$$

Here, $w(x, y)$ is a circular symmetry Gaussian filter of size 7×7 ($P = 7$) [28], so that the SSIM computation is done over $P \times P$ blocks. The l -th image block is $F_l(x, y)$ and the reconstructed block is $V_l(x, y)$. Following the recommendations in [27], $C_1 = 6.5025$ and $C_2 = 58.5225$. For color images, such as in the proposed encoder, the final SSIM value is the average of the SSIM of each color component.

Another metric of interest is the compression factor (F_c) which is the ratio between the number of bits in the original image and the number of bits in the compressed image after the encoder. It indicates how much smaller is the compressed image size with respect to the original image size. It is computed by:

$$F_c = \frac{Nbo}{Nbc} \tag{16}$$

where Nbo is the number of bits of the original image and Nbc is the number of bits of the encoded image.

4.1 Comparative evaluation of compression results

To evaluate the proposed encoder, PSNR and SSIM metrics were first calculated for different image compression factors. They were calculated for both the proposed encoder

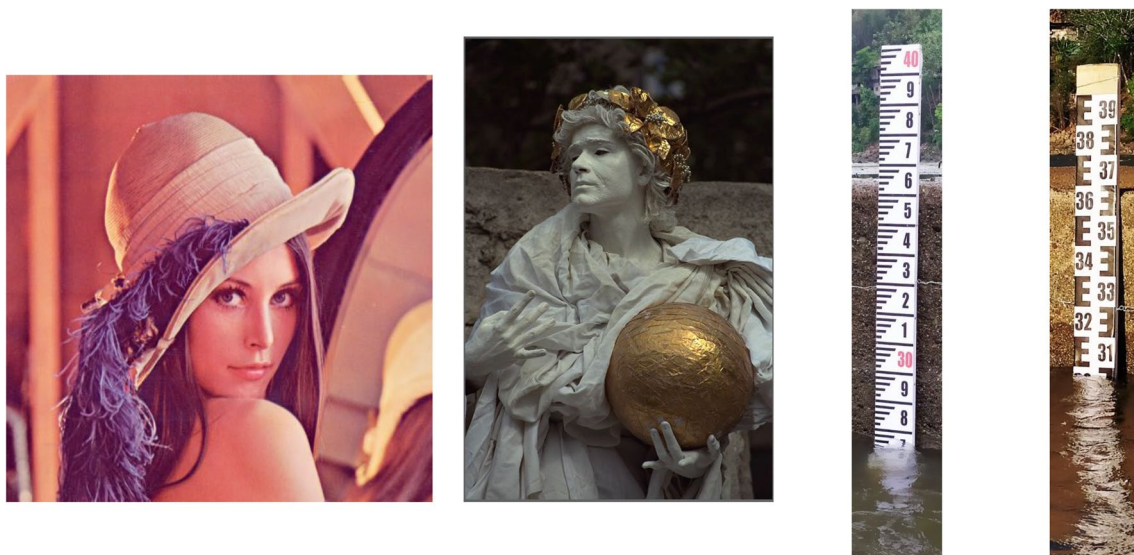


Fig. 9 Images of “Lena”, “Monument” and hydrometric rulers 1 and 2 ordered from left to right

Table 4 Results of comparison between JPEG2000 encoder and the proposed encoder

	Encoded	Compression factor	PSNR	SSIM	Compressed size (bytes)	Dimensions (pixels)
Lena	JPEG2000	111.298	27.069	0.7365	7066	512×512
	Proposed	116.027	26.384	0.7263	6778	
Monument	JPEG2000	139.918	27.596	0.7509	8431	512×768
	Proposed	139.951	27.039	0.7563	8429	
Ruler 1	JPEG2000	135.404	22.341	0.5974	8264	248×1504
	Proposed	139.558	21.708	0.6141	8018	
Ruler 2	JPEG2000	135.305	21.889	0.5603	8270	248×1504
	Proposed	139.541	21.511	0.5917	8019	

Table 5 Transmission time test results for three distances using the proposed encoder

Distance (m)	Module settings			Total packages	Progressive subband transmission					Time header (s)	Total time (s)
	Spreading factor (SF)	Power (dBm)	Fc		First band (s)	Second band (s)	Third band (s)	Fourth band (s)	Fifth band (s)		
2068	7	14	139	59	34	28	24	3	3	19	110
4848	8	16	139	59	45	37	32	4	4	27	148
16,150	11	18	139	59	66	55	47	6	6	36	216

and the JPEG2000 encoder. The following test images were used: "Lena" [28] and "Monument" [29]. As mentioned in the introduction, other tests were performed using images of hydrometric rulers which measure water levels in remote areas. Figure 9 shows these images. The acquired images were encoded and transmitted to a monitoring station with LoRa transceivers. The numbers on the rulers could not be distorted by the encoder, to avoid errors in the determination of water levels.

Table 4 shows the results for the "Lena", "Monument" images and the hydrometric rulers. In all cases, the PSNR and SSIM quality metrics for JPEG2000 and the proposed encoder are very similar, but the proposed encoder achieves a slightly better compression factor and can be adaptable to different situation.

4.2 Evaluation of progressive transmission results

Transmission timing tests were done with hydrometric ruler images, using the SX1272 LoRa transceptor [24]. Testing was done with a total of 20 images of size 248×1504 pixels. Table 5 show the obtained results. There are time measurements for each received subband and additional time measurements for the header frames. The transmission distances of the links were 2068, 4848 and 16,150 m. The maximum test distance required a minimum transmission time of 3 min and 33 s.



Fig. 10 Progressive reconstruction stages, where each image (a–e) corresponds to the sequential reception of each subband. Left to right: first to fifth subband

The subband decomposition and reconstruction technique allows a progressive display of the image and the quality of the reconstruction improves as the subbands are decoded at the receiver. The first decoded subband is the approximation subband, which gives a general idea of the characteristics of the transmitted image.

Table 6 Comparative table of transmission time between the proposed encoder and without the encoder

Distance (m)	Transmission characteristic			Data type		Minimum transmission time	
	SF	Power (dBm)	Original size (bytes)	Number of packages	Number of headers	Without encoder	Proposed encoder
2068	7	14	1,118,976	7773	8	194 min and 32 s	1 min and 48 s
4848	8	16	1,118,976	7773	8	259 min and 22 s	2 min and 26 s
16,150	11	18	1,118,976	7773	8	389 min and 03 s	3 min and 33 s

The proposed encoder can achieve a low-quality image transmission in 34 s for a 2069-m link, and in 66 s for a 16,150-m link. Yet, the quality will improve as the rest of subbands reach the decoder. Figure 10 illustrates this progressive quality improvement.

4.3 Comparative evaluation of encoded and non-coded transmissions

Table 6 presents the results of encoded and un-encoded image transmission tests. Clearly, there is a significant improvement in the transmission time of the encoded images compared to the un-encoded ones. For a link of 2068 m and using the same transceiver configuration as for coded images, there is a 99.07% reduction in transmission time compared to the time required to transmit coded images; for a 16,150-m link, the reduction is of 99.09%. Therefore, the benefit of using the proposed encoder is evident: there is a faster transmission time and, consequently, a lower energy usage and a longer autonomy of the device.

5 Conclusions and future scope

When compared to the standard JPEG2000 image encoder, the proposed encoder achieved similar quality levels, measured by the PSNR and SSIM. Yet, the proposal in this work did this with a higher compression level, while JPEG2000 has compression limits. This is summarized in Table 4, where the proposal results in similar PSNR and SSIM than JPEG2000 but with slightly higher compression rates.

Moreover, the proposed encoder obtained a reduction in transmission times of up to 99.09% with regards to transmitting un-encoded images. Increasing the transmission distances did not have a significant impact on this time reduction, which could translate into lower energy consumption and a greater device autonomy when the device is powered by batteries. Furthermore, the largest transmission distance was of 16,150-m, and it could be increased by using the maximum capacity of the LoRa transceptor.

As another concluding remark, this work demonstrates that the proposed encoder firstly achieves an acceptable quality image visualization with the reconstruction of the

first received subband (approximation subband). Then, this quality progressively improves as the rest of the subbands are received and decoded. This progressive transmission, reception and decoding accommodates the encoder to the limitations of LoRa technology. Hence, there is a significant reduction in transmission and visualization times of the information of interest.

Finally, a particular application for this encoder was explored, transmitting images of hydrometric rulers installed in water sources, as part of a project where the environmental parameters and level of these sources was remotely monitored. This application, where water overflows after El Niño events can severely harm the surrounding population, requires the complementary information which these images bring to the information transmitted by the water level sensors. Moreover, these sensors are prone to failure. Hence, the complementary information brought by the hydrometric ruler images is even more impactful. The proposed image encoder enables adding these redundancies in low-connectivity regions via LoRa networks.

Future works could improve the performance of the encoder using artificial intelligence and machine learning in combination with the subband decomposition technique, while maintaining a low computational load in the overall algorithm. It is important to state that the encoder could be also implemented in other single-board computers, such as Toradex, Jetson Nano, Orange Pi or others.

Funding The authors would like to thank the Dirección de Investigación de Universidad Peruana de Ciencias Aplicadas for funding and logistical support with code UPC-D012-2021.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Ali I, Partal SZ, Kepke S, Partal HP (2019) ZigBee and LoRa based wireless sensors for smart environment and IoT applications. In: 2019 1st global power, energy and communication

- conference (GPECOM), Nevsehir, Turkey, pp 19–23. <https://doi.org/10.1109/GPECOM.2019.8778505>
2. Huh H, Kim JY (2019) LoRa-based mesh network for IoT applications. In: 2019 IEEE 5th world forum on Internet of Things (WF-IoT), pp 524–527. <https://doi.org/10.1109/WF-IoT.2019.8767242>
 3. Khutsoane O, Isong B, Abu-Mahfouz AM (2017) IoT devices and applications based on LoRa/LoRaWAN. In: IECON 2017—43rd annual conference of the IEEE industrial electronics society, pp 6107–6112. <https://doi.org/10.1109/IECON.2017.8217061>
 4. Saari M, bin Baharudin AM, Sillberg P, Hyrynsalmi S, Yan W (2018) LoRa—a survey of recent research trends. In: 2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO), pp 0872–0877. <https://doi.org/10.23919/MIPRO.2018.8400161>
 5. Kolobe L, Sigweni B, Lebekwe CK (2020) Systematic literature survey: applications of LoRa communication. *Int J Electr Comput Eng* 10(3):3176–3183. <https://doi.org/10.11591/ijee.v10i3.pp3176-3183>
 6. Kirichek R, Pham VD, Kolechkin A, Al-Bahri M, Paramonov A (2017) Transfer of multimedia data via LoRa. In: Lecture notes in computer science (including Subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 10531 LNCS, pp 708–720. https://doi.org/10.1007/978-3-319-67380-6_67
 7. Wei C-C, Su P-Y, Chen S-T (2020) Comparison of the LoRa image transmission efficiency based on different encoding methods. *Int J Inf Electron Eng* 10(1):1–4. <https://doi.org/10.18178/ijee.2020.10.1.712>
 8. Chen T, Eager D, Makaroff D (2019) Efficient image transmission using LoRa technology in agricultural monitoring IoT systems. In: 2019 international conference on Internet of Things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), Atlanta, GA, USA, pp 937–944. <https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00166>
 9. Ji M, Yoon J, Choo J, Jang M, Smith A (2019) LoRa-based visual monitoring scheme for agriculture IoT. In: SAS 2019—2019 IEEE sensors applications symposium conference proceedings, pp 1–6. <https://doi.org/10.1109/SAS.2019.8706100>
 10. Haron MH, Isa MN, Ahmad MI, Ismail RC, Ahmad N (2021) Image data compression using discrete cosine transform technique for wireless transmission. *Int J Nanoelectron Mater* 14(Special Issue InCAPE):289–297
 11. Hu P, Im J, Asgar Z, Katti S (2020) Starfish: resilient image compression for AIoT cameras. In: SenSys 2020—proceedings of the 2020 18th ACM conference on embedded networked sensor systems, pp 395–408. <https://doi.org/10.1145/3384419.3430769>
 12. Ahmad N, Jaffery ZA, Sharma D (2019) Low bitrate image coding based on dual tree complex wavelet transform. In: 2019 international conference on power electronics, control and automation (ICPECA), New Delhi, India, pp 1–6. <https://doi.org/10.1109/ICPECA47973.2019.8975652>
 13. Al-Azawi S, Boussakta S, Yakovlev A (2011) Low complexity image compression algorithm using AMBTC and bit plane squeezing. In: International workshop on systems, signal processing and their applications, WOSSPA, Tipaza, pp 131–134. <https://doi.org/10.1109/WOSSPA.2011.5931432>
 14. Prades-Nebot J (2011) Very low-complexity coding of images using adaptive Modulo-PCM. In: 2011 18th IEEE international conference on image processing, Brussels, pp 305–308. <https://doi.org/10.1109/ICIP.2011.6116310>
 15. Telles J, Kemper G (2019) A multispectral image compression algorithm for SmallSatellites based on wavelet subband coding. Lima
 16. Raspberry Pi Ltd (2014) Raspberry Pi 3 Model B+. Raspberry Pi. <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>
 17. Malvar H, Sullivan G (2003) YCoCg-R: a color space with RGB reversibility and low dynamic range. *Iso/Iec Jtc1/Sc29/Wg11 Itu-T Sg16 Q 6*(July):22–24
 18. Dumic E, Mustra M, Grgic S, Gvozden G (2009) Image quality of 4:2:2 and 4:2:0 chroma subsampling formats. In: 2009 international symposium ELMAR, pp 19–24
 19. Gonzales RC, Woods RE (2006) Digital image processing, 3rd edn. Prentice Hall, New York
 20. Mathworks Inc (2022) Wavelet filters. <https://www.mathworks.com/help/wavelet/ref/wfilters.html>
 21. Mahmoud Afifi (4 Jan 2019) Structure similarity (SSIM) and PSNR, MATLAB central file exchange (Online). <https://www.mathworks.com/matlabcentral/fileexchange/64151-structure-similarity-ssim-and-psnr>
 22. Kemper G, Iano Y (2011) An audio compression method based on wavelets subband coding. *IEEE Lat Am Trans* 9(5):610–621. <https://doi.org/10.1109/TLA.2011.6030967>
 23. Ranjan R (2021) Canonical Huffman coding based image compression using wavelet. *Wirel Pers Commun* 117(3):2193–2206. <https://doi.org/10.1007/s11277-020-07967-y>
 24. Semtech (2022) SX1272/73—860 MHz to 1020 MHz low power long range transceiver. SX1272/73 datasheet, Jan 2019 (Revised Feb 2022)
 25. Kok W, Tam WS (2019) Digital image interpolation in MATLAB, 1st edn. Wiley, New York
 26. Horé A, Ziou D (2010) Image quality metrics: PSNR vs. SSIM. In: Proceedings of international conference on pattern recognition, pp 2366–2369. <https://doi.org/10.1109/ICPR.2010.579>
 27. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612. <https://doi.org/10.1109/TIP.2003.819861>
 28. Lenna.org (2022) The Lenna story (Online). <http://www.lenna.org/>
 29. Eastman Kodak Company, True color kodak images, R0k.us (Online). <http://r0k.us/graphics/kodak/>

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.