International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Lossless Meteorological Images Compression

Leila Sadouki *[1], Abderraouf Senhadji [2], Boualem Haddad [3]

Submitted: 15/03/2024 **Revised:** 25/04/2024 **Accepted:** 07/05/2024

Abstract: Nowadays, with the spread of imaging and examination devices, digital images have become ubiquitous. Satellite and radar images are of particular importance due to their diverse applications. In meteorology, where image resolution and pixel accuracy are critical for accurate rainfall measurements, the use of lossy compression can degrade image quality and distort pixel value, leading to inaccurate results. Hence, maintaining image resolution through lossless compression is essential to maintain the reliability of weather data and ensure the accuracy of forecasts and analyses. Satellite images are often very large, posing significant challenges for their storage and transmission. Image compression addresses this problem using lossless techniques that allow for perfect reconstruction of the original image. Therefore, our study uses a Huffman coding algorithm and two types of predictive coding which are error coding and facsimile coding. For the satellite images, predictive coders achieve a higher compression ratio than the Huffman coder and the compressed bit rate can even drop below the entropy limit. Moreover, and due to the homogenous zones of pixels with the same intensity in the radar image, the facsimile predictive coder generated the lower bit rate than the other coders in relatively shorter time.

Keywords: *Compression, Lossless, Radar image, Satellite image, Meteorology.*

1. Introduction

Vision is one of the most developed human senses, where images play an extremely important role in the human perception. Today, due to the fast development in imaging devices, such as digital cameras and different types of scanner, as well as the increasing number of screens from televisions to computers and mobile phone, digital images have become ubiquitous in people's daily lives. In fact, advanced imaging machines such as the ones present on satellites are not limited to the visible band of the electromagnetic spectrum and can cover almost the entire spectrum using an important number of sensors. The images that are obtained by these satellites are called multispectral images and have a wide range of applications depending on the band of the electromagnetic spectrum that is captured. For instance, the LANDSAT satellite, whose main purpose is to take pictures of the earth from space to monitor environmental conditions, uses multispectral images for plant vigor measurements, soil thermal mapping, shorelines definition, etc. This type of images is also used to obtain night time images of the World which may be used for the detection of human settlement areas or to discover large wild fires. In addition to that, multispectral satellite images are also extensively used for weather observation and prediction [1]. It can clearly be understood that such large amounts of information give rise to big data storage space requirements [2]. Moreover, satellites can only communicate with ground stations when the former visit the area of the earth where the later are located. Therefore, the communication systems used must be able to handle such large data sets in limited time intervals. In order to meet such requirements, satellite image compression becomes a mandatory step [3]. Satellite image compression is the technique used to reduce the size of the data without loss of important information. This process is possible by exploiting the redundancy in the data and/or by discarding irrelevant information. Many compression algorithms such as Arithmetic, Lempel-Zilv-Weslh or Goulomb coding are well defined and established in the literature [1]. Moreover, in 1987, a new technique called multi-resolution technique was developed by Mallat which allowed the representation of images at multiple resolutions. Using this technique the Discrete Wavelet Transform represents an image as a linear combination of a finite function with zero average value called the mother wavelet, and scaled and shifted versions of it called baby wavelets [4]. This image transform is the basis for the JPEG 2000 standard and the Consultative Committee for Space Data Systems (C.C.S.D.S.) recommendation for image data compression which achieve good results for satellite images [5]. Vector Quantization is another compression method; it represents blocks of the image using a single codeword selected from a codebook that is generated using training vectors [6].

There are two types of image compression; lossless compression in which the restored image is perfectly identical to the original, and lossy compression in which the decompressed image is not exactly the same as the original. In general, lossy compression can achieve a greater compression rate than lossless compression and depending on the application either one of the methods is used. For example, some images with enormous data volume necessarily require lossy compression [7] while precision applications need lossless compression. In the field of meteorological data processing, the pixel's resolution or intensity

1 University M'Hamed Bougara of Boumerdes, Institute of Electrical and Electronic Engineering, Independence Avenue-35000, Boumerdes, Algeria.
ORCID ID : 0000-0003-0333-9811
2 University M'Hamed Bougara of Boumerdes, Institute of Electrical and Electronic Engineering, Independence Avenue-35000, Boumerdes, Algeria.
ORCID ID : 0009-0006-6587-7980
3 University of Sciences and Technology Houari Boumediene, Faculty of Electrical Engineering, Algiers, Algeria.
ORCID ID : 0000-0001-5230-1559
** Corresponding Author Email: l.sadouki@univ-boumerdes.dz*

in radar or satellite image is very significant for correct rainfall estimations this is why lossless compression is the preferred choice.

2. Data Used

A digital image could be defined as a two-dimensional sequence of sample values I[m,n], where “m” is the vertical spatial coordinate also called the row index and “n” is the horizontal spatial coordinate also called the column index. Each image sample is called a pixel, and the pixel size of a digital image is called the resolution. The sample value I[m,n] represents the intensity or brightness of the image at location [m,n] [1], [8]. In order to obtain the sample values, the image must be sampled and then quantized using a digitizer such as a scanner or a digital camera. Thus, the digitizer takes samples (pixels) of the continuous tone image at given spatial intervals. After that, each pixel is assigned a B-bit value, generally B= 8 bits. Each pixel value is a scalar that is proportional to the real valued brightness quantity; using 8 bits, the pixel values range from 0 to 255.

Images with multiple components, such as color images are represented with separate sequences; one for each component. For instance, color images can be represented using the three components; Red, Green and Blue color (RGB). If each color component is digitized using 8 bits, each pixel will be represented using 24 bits, this is known as 24-bit color depth which can represent more than 16.7 million colors [9].

The data set used in this study consists of three meteorological images that are described as follows:

a. A radar image of a cloud saved as “Im1.tif” file that has a size of 18.1 Kbytes. Its resolution is 510 x 510 pixels .This image has an associated color map which is not coded.
b. A satellite images displaying planet Earth from the African side and surrounding clouds. They are saved as “Im2.jpeg” and “Im3.jpeg” files with the sizes of 2.08 Mbytes and 2.26 Mbytes respectively, and the resolution of 3712 x 3712 pixels (gray scale).

3. Data pre-processing

In order to be able to reduce the size of data used to represent the amount of information that exists in an image to achieve compression, some of this data needs to be redundant. Thus, it is either a duplicate, it can be predicted using the rest of data or it is simply irrelevant to the application.

3.1. Spatial redundancy

The pixels of most images are not statistically independent. In other words, they are more or less correlated; especially adjacent pixels which are highly correlated [10]. There are two main types of neighborhoods in two-dimensional arrays, such as the ones used to represent digital images, 4-neighborhoods and 8-neighboorhoods, as shown in Fig.1.

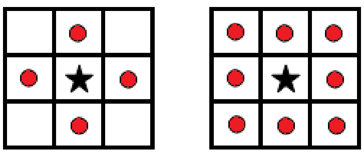

Fig 1. 4-neighborhood (on the left) and 8-neighborhood (on the right)

Spatial redundancy implies that the value of a given pixel can be reasonably predicted from its neighbors [1]. In that sense, the actual pixel value does not hold much information since it can be guessed using its neighbors. This mainly possible due to the fact that, except for the edge regions, the pixel values change gradually along a row or a column. Moreover, images; especially computer generated ones, may contain regions with uniform pixel values in which the inter-pixel redundancy is very high.

3.2. Coding redundancy

A code is a system of symbols used to represent a body of information. Each part of information is assigned a sequence of code symbols, called a code word. The number of symbols in each code word is called the code length [1]. In this study bits are the symbols used to represent the information which is the pixel intensity values. More than often, the 8-bit codes used to represent pixel intensity values in digital images are longer than needed. Indeed, the number of bits required to code the pixel values depends on the statistical properties of the image [8]. Thus, in the worst case, if all the 256 levels are equally like; i.e. have the same probability, there is no redundancy and all the 8-bit codes are needed. However, if all 256 intensity levels are not present in the image, it might be possible to code the image with fewer bits per code word. Moreover, if the probability distribution of the different intensity levels is highly non-uniform and a few values are more likely than the others, a variable length code can achieve high compression rates.

3.3. Irrelevance

It is obvious that removing unnecessary information from a dataset contributes to data size reduction. Two main types of irrelevant information can be distinguished in the context of digital image compression: information that is not useful for the intended purpose and information that is ignored by the Human Visual System (HVS) [1]. The first type of irrelevance is application-specific. For example, only one region of the image is of interest in some military and medical applications, which makes the rest of the image irrelevant, and can therefore be omitted [8].

Likewise, some applications do not require the use of all image components, such as specific applications of multispectral satellite image. The data related to these components are therefore removed.

On the other hand, the second type of irrelevance is concerned with the characteristics of the HVS. In fact, the human eye’s perception of images is complicated and is different from that of camera sensors [10]. The HVS is more sensitive to some visual information rather than the other. Therefore, using fewer data to represent less visually important information, also referred to as psycho-visually redundant data, does not affect the visual quality of the image. Consequently, image compression is achieved. For instance, the HVS is more sensitive to the luminance of color image rather than their hue or saturation. Thus, for compression purposes the original RGB image is mapped to a luminance-chrominance space using a linear transformation. After that, the chrominance components are sub-sampled in both the horizontal and vertical directions, reducing the number of chrominance components by 4.

4. Compression system

A typical compression system consists of two parts: a compressor and a decompressor which can be modeled as mapping operations M and M* respectively [1], [8]. The compressor may be preceded by a pre-processor that can perform various application-specific image processes such as image enhancement or noise elimination to prepare the image for compression [6]. Additionally, the decompressor might be followed by a post-processor to improve the final look of the image.

After the pre-processing of the original input image, if the application requires such a step, it is fed onto the compressor which generates a compressed representation of the input, also referred to as the bit stream. This bit stream may have a fixed length or a variable length depending of the type of encoder used by the compressor. Generally, variable length encoding is preferred since it takes advantages of the statistical redundancy present in the input. This resulting compressed representation is the one that is stored or transmitted to a remote location in a communication system. On the other hand, the decompressor receives the bit stream which it decodes creating a reconstructed output image. If the decompression mapping M* is exactly the inverse of the compression mapping M; i.e. $M^*= M^{-1}$, the system performs lossless compression. Finally, the reconstructed image is post-processed if it is necessary for the application. A systematic view of a typical compression/decompression system is given in Fig.2.

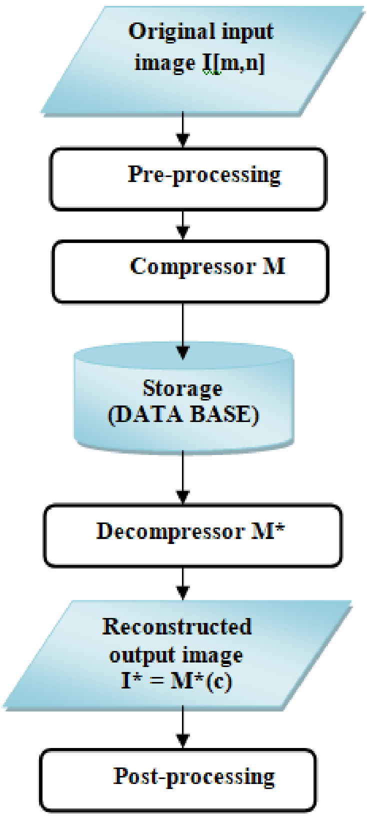

Fig 2. Compression/Decompression system.

4.1. Measures of compression

In order to quantitatively measure the amount of compression that is performed by a given compression system, two main quantities are defined, the **Compression Ratio** (C.R) and the **Compressed Bit Rate** (C.B.R).

Without compression, an input image of size "m x n" is represented by "m x n x B" bits if each pixel value is represented by B bits. The generated bit stream is denoted "c" and its length is ||c|| [8].

Thus, the Compression Ratio (C.R) achieved by the compressor is defined as follows:

$$\text{C.R}= \frac{m \times n \times B \text{ (size of the input)}}{\|c\| \text{ (size of the bit stream)}} \quad (1)$$

The Compressed Bit Rate (C.B.R) is defined as in the following equation and is expressed in bits per pixel (bpp).

$$\text{C.B.R}= \frac{\|c\| \text{ (size of the bit stream)}}{m \ \times n \text{ (total number of pixels)}} \quad (2)$$

4.2. Lossless compression

Information theory provides the mathematical framework for the study of the statistical properties of information and how it can be compressed without losses [1]. The first person to formally introduce this theory is Claude Elwood Shannon, an electrical engineer at Bell Labs [11]. Indeed, he developed a quantity to measure the amount of information conveyed by an event, which he called a message "s". This quantity is self-information i(s) which he defined as follows:

$$i(s)= \log_b \frac{1}{p(s)} =- \log_b p(s) \quad (3)$$

Where p(s) is the probability of occurrence of the message "s".

Note that the base of the logarithm, "b", determines the information unit. If b=2, the unit is bits and the self-information is then, roughly speaking, the number of bits needed to represent the message without losses [12]. This definition implies that totally deterministic messages for which p(s) =1, i(s) = 0. In other words, they hold no information since they always occur.

For 8-bit intensity levels the alphabet is $A_X = \{\alpha_0, \alpha_1, \alpha_2,\ldots\}=\{0,1,2,\ldots,255\}$. Thus, the self-information conveyed by the occurrence of a given pixel intensity is computed as follows.

$$i(\alpha_i)= \log_b \frac{1}{p(\alpha_i)} =- \log_b p(\alpha_i) \quad (4)$$

Where, $p(\alpha_i)$ is the probability of the occurrence of a pixel with intensity value αi computed with considering that all pixel values are statistically independent.

$$p(\alpha_i)= \frac{\text{Number of pixels with intensity value } \alpha_i}{\text{Total number of pixel } (m \times n)} \quad (5)$$

In addition to the concept of self-information, Shannon has also defined entropy which he has borrowed from physics where it measures the disorder and randomness of a system [12]. For a random variable X, the entropy is defined as follows and is measured in the unit of bits/pixel [1].

$$H(X) = -\sum_{\alpha_i \in A_X} p(\alpha_i) \log_2 p(\alpha_i) \quad (6)$$

4.3. Huffman coding

Huffman codes are optimal prefix codes generated by the Huffman Coding Algorithm (HCA) developed by David Albert Huffman in the 1950's [13]. In general in computer science, **fixed-length codes** are used to associate each message with a codeword of a predetermined length such as the ASCII code that maps each printable character to a 7-bit code-word [12]. However, such codes do not take advantage of the statistical redundancy that may exist in a data set. For this reason **variable-length codes** which assign to each element of the alphabet or message $\boldsymbol{\alpha_i}$ a distinct codeword $\boldsymbol{c_{\alpha_i}}$ of length $\boldsymbol{L_{\alpha_i}}$ are used [8]. The sequence of messages is thus represented by concatenating the individual codeword of each message in the sequence. In the context of digital image compression, the messages are pixel intensity values and the sequence of pixels is the image. Therefore, the compressed bit stream is generated by concatenating the codeword of each pixel intensity values as it appears on the original image. In fact, assigning shorter code-words for messages with higher probabilities and vice versa will contribute to the reduction of the data size.

In order to avoid ambiguity and allow the decoder to clearly identify the code-words from the concatenated bit stream, it is necessary to design a code that is **uniquely decodable**. One interesting type of uniquely decodable is **prefix codes** in which no code is the prefix of any other. Prefix codes can be represented as **binary tree** where each message is a leaf of the tree, and the codeword for each message is obtained by following the path from the root of the tree until the leaf and assigning either 1 or 0 when going left or right down the tree.

For the purpose of compression, the **average code rate** of the prefix code, which is defined as follows, must be minimized (close to the entropy).

$$R = \sum_{\alpha_i \in A} L_{\alpha_i} p(\alpha_i) \geq H \quad (7)$$

A particular type of Huffman codes is taken into consideration for this algorithm; it is **canonical Huffman codes** [14]. Canonical Huffman codes are characterized by two main properties. The first is that when shorter codes are filled with zeros to the right, their numerical value is superior to that of longer codes. Secondly, the numerical values of canonical Huffman code-words increase alphabetically within fixed lengths. In the context of digital image compression the numerical values of canonical Huffman code-words increase with the increase of the coded pixel intensity value within fixed lengths. (Example: if both pixel intensity values "7" and "19" are coded using codewords of length "1"1, the numerical value of the codeword for "19" must be greater than that of the codeword for "7").

4.4. Predictive coding

4.4.1. Facsimile predictive encoding

It is called facsimile predictive encoding because it is one of the techniques used for FAX telecommunication [9]. Unlike the previous approach where the error between the actual and predicted pixel intensity values is encoded,

This method uses a tag for spatially correlated pixels. In fact, the algorithm checks the similarity between the actual pixel intensity value and the predicted pixel intensity value, if they match, the actual pixel intensity value is replaced by the marker, 256; otherwise, the actual value is encoded. At the end of the process, homogeneous areas of the images are represented as blocks of the marker 256, thus converting spatial redundancy into statistical redundancy. These new values are then injected to the entropy coder.

4.4.2. Predictive error encoding

The predictive error e[i, j], is computed as the difference between the actual pixel intensity value I[m, n] and the predicted pixel intensity value Ip[i, j] which is defined as the intensity value of the previous pixel in the row if its row index is higher than 1 and, as the intensity value of the above pixel if the row index is 1. Finally, the predictive error values are fed to the entropy encoder [6]. In the decoder, the predictive errors are reconstructed and the decoded pixel intensity values are obtained by adding the predictive error values to the previous pixel intensity values according to how the error is calculated in the compressor. Fig.3 illustrates the diagram for a predictive error encoder system.

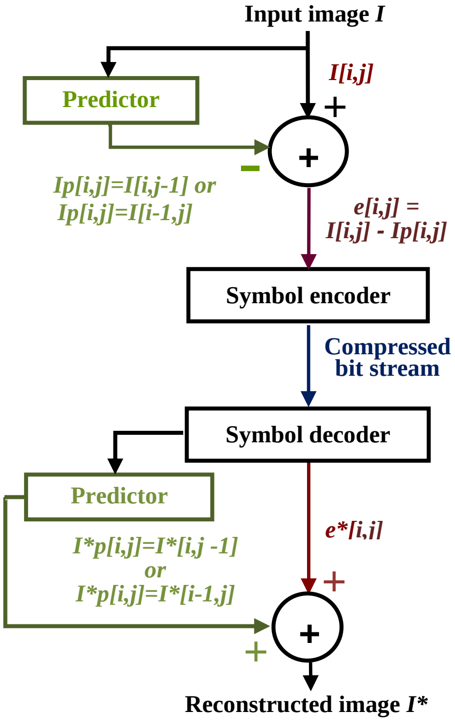

Fig 3. The structure of a predictive error compression system

4.5. Lossy compression

A typical lossy compressor consists of three successive main components [1]. The first part is a **transformer** that maps the digital image matrix representation "I" into another representation "T(I)", usually non visual, that is more adequate for coding [6]. After that, the transform values are fed into a **quantizer** which diminishes the number of possible values for the mapper's output to discard irrelevant information. Quantization is the step responsible for introducing distortion since it causes information loss [8]. Finally, the quantization values are encoded using a **symbol encoder** which assigns fixed or variable length symbols (or codewords) to the quantizer's output to generate a compressed bit-stream which will be stored or transmitted.

On the other hand, the decoder is composed of a **symbol decoder** and an **inverse transformer**. Since the quantization step is irreversible, an inverse quantizer cannot be used in the decompression procedure.

The Joint Photographic Experts Group (JPEG) baseline mode is the most adequate for the majority of image compression applications and it only supports lossy compression [6]. The compression occurs in four successive steps:

a. 8x8 pixels image blocking. If the image resolution is not a multiple of 8, the border blocks are padded either using zeros or some symmetry of the original pixels.

b. Discrete Cosine Transform (DCT) coefficients computation for each block to generate a set of 64 DCT coefficients [9]. Next, these coefficients are normalized using a normalization array. The standard provides a default normalization array which can be scaled by a quality factor that ranges up to 100% to reduce the compressed file size even further.

c. Quantization: is achieved by rounding the normalized DCT coefficients to the nearest integer.

d. Variable-length encoding.

5. Results and discussions

MATLAB is the language chosen to implement the compression algorithms developed for the sake of this study. The three images have been compressed and decompressed using the two predictive coding schemes, and the canonical Huffman coding algorithm. The use of the faster Huffman compression scheme for the satellite images is due to their big size because the execution time of original Huffman algorithm is about 167 hours which is highly inefficient. For each compression type, the entropy, the compression ratio, the compressed bit rate and the execution time are recorded in the tables,1, 3 and 4.

Table 1. The experimental results for Radar image Im1.tif.

Compression type	***Huffman***	***Predictive error***	***Facsimile prediction***
Entropy (bpp)		0.476	
Compression ratio	6.951	7.343	7.451
Compressed bit rate (bpp)	1.151	1.089	1.073
Coder execution time (s)	34.323	2.160	5.320
Decoder execution time (s)	6.940	7.131	1.538

In addition to the lossless techniques, and in order to compare our results, the radar image "Im1.tif" was compressed using the lossy pseudo JPEG algorithm. The entropy, the compression ratio (C.R), the compressed bit rate (C.B.R) and the coding and decoding execution times for this image with the different quality factors (1, 0.5 and 0.25) are recorded in Table 2

Table 2. The experimental results for Radar image Im1.tif using the lossy compression..

Quality factor	***1***	***0.5***	***0.25***
Entropy (bpp)		0.476	
Compression ratio	66.108	86.740	106.370
Compressed bit rate (bpp)	0.121	0.092	0.075
Coder execution time (s)	1.182	1.660	1.501
Decoder execution time (s)	1.800	1.442	1.055

The original radar image and the reconstructed compressed image for both lossy compression, with the quality factors (1 and 0.5), and lossless compression are shown in Fig.4.1, 4.2, Fig.4.3 and Fig.4.4 respectively.

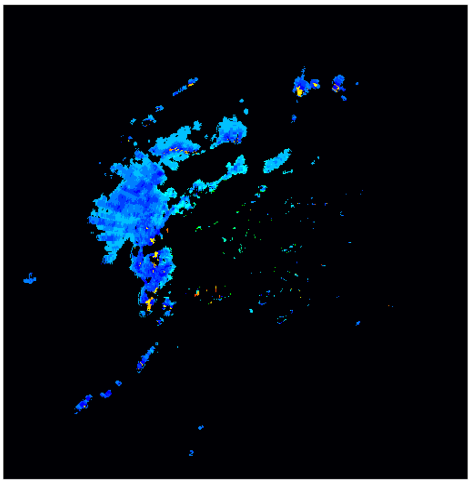

Fig 4.1. The original Radar image "im1.tif".

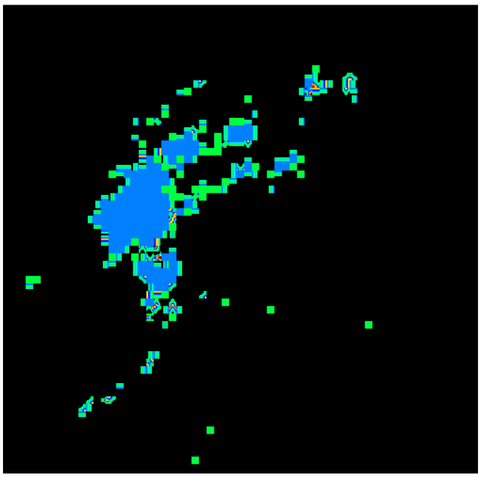

Fig 4.2. The reconstructed Radar image 'im1.tif' using lossly compression with quality factors =0.5.

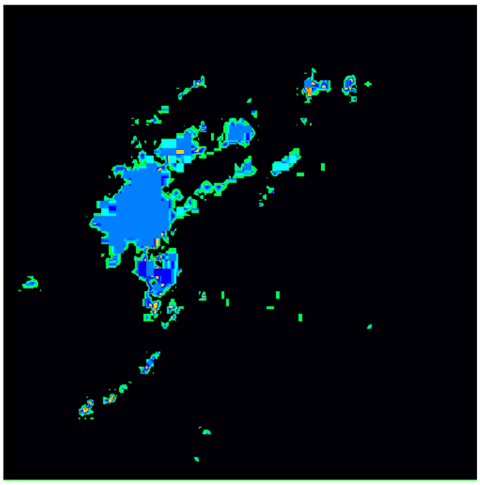

Fig 4.3. The reconstructed Radar image 'im1.tif' using lossly compression with quality factors =1.

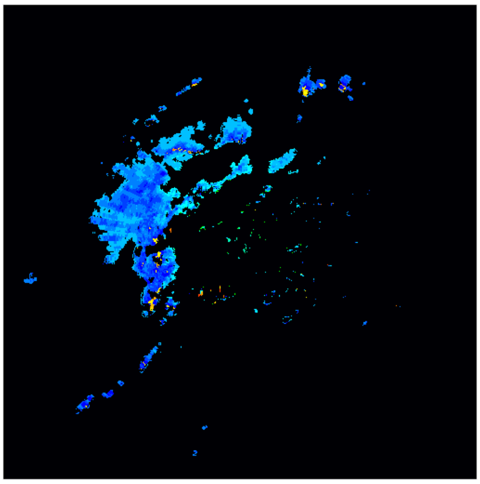

Fig 4.4. The reconstructed Radar image 'im1.tif' using lossless compression.

The Huffman coder achieved a relatively good compression ratio for the Radar image "im1.tif" in a relatively moderate time, but it couldn't approach the entropy because of the 1bpp limit imposed by the nature of Huffman coding. Concerning the predictive coders, they both achieved a compression ratio that is higher than the one of the simple Huffman coder.

The results of the pseudo JPEG reconstruction of the radar image were theoretically good; the reconstructed image itself does not visually resemble the original. This can be explained by the nature of this image which consists of a large homogenous zone of pixels with intensity value of "1" interspersed with a small number of pixels with different values in a small scattered region which cause significant high frequency components in the DCT spectrum of the image. Since, these high frequency components are highly quantized by the pseudo JPEG compression, the reconstructed image is poor in visual quality despite the good theoretical objective quality assessment measurements values.

Table 3. The experimental results for satellite image Im2.jpeg.

Compression type	*Faster Huffman*	*Predictive error*	*Facsimile prediction*
Entropy (bpp)		6.640	
Compression ratio	1.202	1.707	1.313
Compressed bit rate (bpp)	6.656	4.687	6.092
Coder execution time (s)	501.566	139.723	471.468
Decoder execution time (s)	2019.886	610.719	1865.544

Table 4. The experimental results for satellite image Im3.jpeg.

Compression type	**Faster** *Huffman*	*Predictive error*	*Facsimile prediction*
Entropy (bpp)		6.568	
Compression ratio	1.213	1.751	1.350
Compressed bit rate (bpp)	6.595	4.569	5.924
Coder execution time (s)	458.635	180.035	426.332
Decoder execution time (s)	1604.596	489.694	1505.247

The original and reconstructed compressed satellite images using the lossless compression are shown in Fig.5.1, Fig.5.2, Fig.6.1 and Fig.6.2.

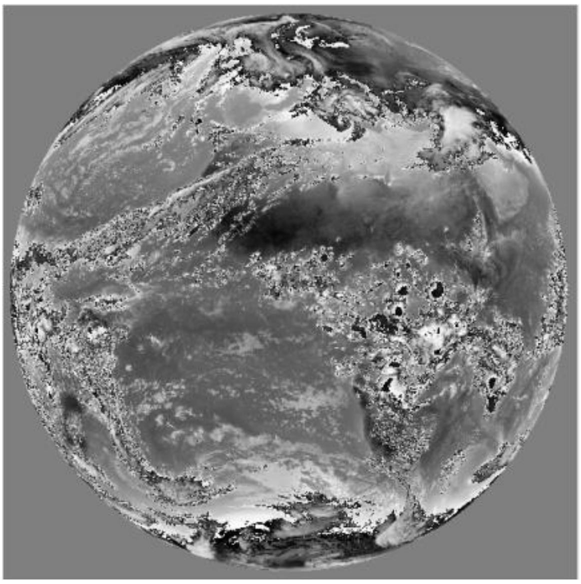

Fig 5.1. The original Satellite image "im2. Jpeg".

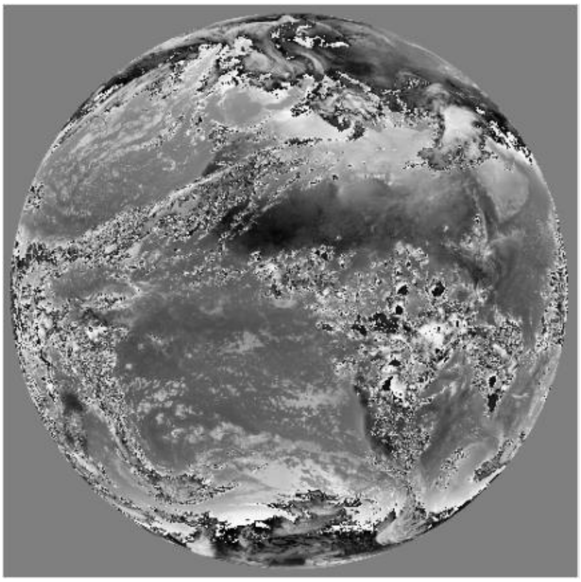

Fig 5.2. The reconstructed Satellite image "im2. Jpeg".

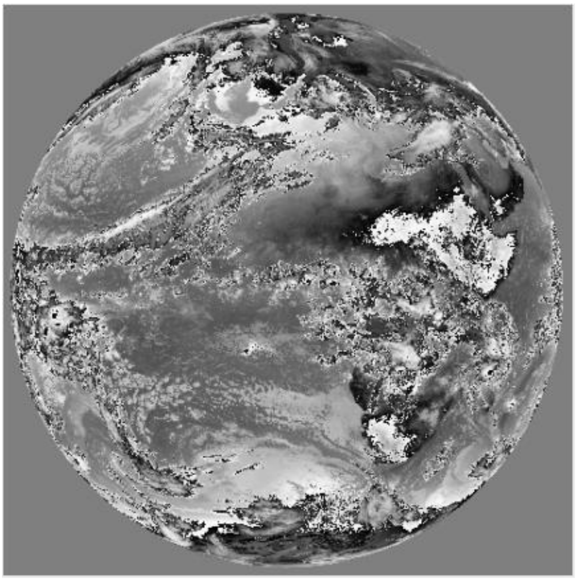

Fig 6.1. The original Satellite image "im3. Jpeg".

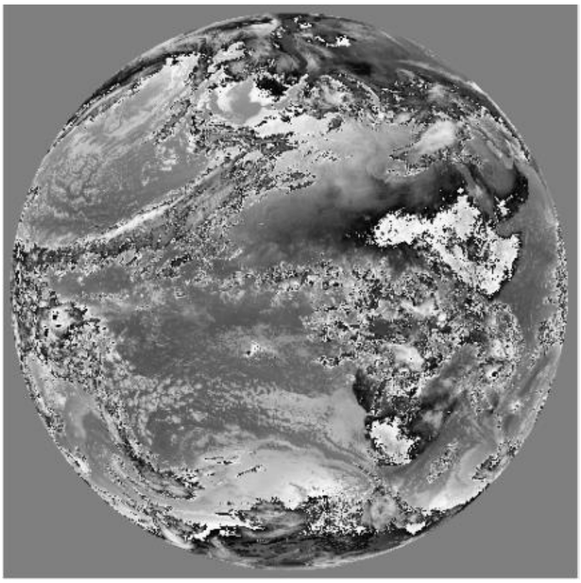

Fig 6.2. The reconstructed Satellite image "im3. Jpeg".

We can see through the results of the lossless compression of the two satellite images, “im2.Jpeg” and “im3.Jpeg”, that the Huffman coder was able to closely approach the entropy. However, the execution time for these two images; especially the decoder time, was considerably high despite the use of the faster algorithm. Concerning the predictive coders, they both achieved compression ratios that are higher than the one of the Huffman coder, and were even able to go below the entropy limit for the two images.

6. Conclusion

Satellite images serve nowadays a wide spectrum of applications ranging from mapping and cartography to meteorology and weather forecast. With an increasing file size especially with the improvement of the sensors used to capture the images and the rise of multi- and even hyper-spectral satellite images, the storage space requirements and the transmission times necessary to communicate the images to the Earth station increase dramatically. In order to overcome both these issues image compression is the most adequate solution.

In this paper, we were able to successfully implement a various algorithms, lossless compression, such as Huffman and predictive error coders and lossy compression given by the pseudo JPEG reconstruction for only radar image in order to compare the two techniques of compression.

For the satellite images, the predictive error coder generated bit streams with a lower bit rate than those of the facsimile predictive coder and the faster Huffman coder in a considerably shorter time.

In the case of radar images, the facsimile predictive coder generated bit streams with a lower bit rate than those of the predictive error coder and the Huffman coder in a considerably shorter coder and decoder execution time. Hence, Huffman coding performs the highest compressed bit rate in relatively long execution time.

However, lossy compression achieved a higher compression ratio and subsequently a lower compressed bit rate and execution time compared to the lossless compressor, mainly because lossy schemes remove a given part of the information present in the image which gives a considerable loss in the useful data.

Depending on the properties of the image especially the inter-spatial pixel correlation, the optimal lossless compression can be achieved using the predictive error scheme when no particular homogenous zones of pixels with the same intensity value appear in the image.

Acknowledgements

The authors would like to thank the National Meteorology Office of Algeria for the radar and satellite data set used.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] R.C. Gonzalez, and R.E. Woods, “Digital Image Processing,” 4th ed.. New York: Pearson Education, 2018.

[2] S. Sehgal, L. Ahuja, and M. H. Bindu, “High Resolution Satellite Image Compression using DCT and EZW,” *Int. Conf. on Sustainable Computing in Science, Technology and Management, Amity Univ. Rajasthan, Jaipur – India,* pp. 112-115, 2019.

[3] K.S. Gunasheela, H.S. Prasantha, “ Satellite Image Compression-Detailed Survey of the Algorithms,” In *Proc. of Int. Conf. on Cognition and Recognition*, Singapore : Springer , vol. 14, pp. 187–198, 2018.

[4] S. Dhawan, “A Review of Image Compression and Comparison of its Algorithms,” *Int. Journal of Electronics & Communication Technology (IJECT),* vol. 2, no. 1 , pp. 22-26, Mar. 2011.

[5] A. Indradjad, A. Nasution, H. Gunawan, and A. Widipaminto, “A comparison of Satellite Image Compression methods in the Wavelet Domain,” *IOP Conf. Series: Earth and Environmental Science*, 2019.

[6] A. Hussain, A. Al-Fayadh, and N. Radi, “Image Compression Techniques: A Survey in Lossless and Lossy algorithms,” *Neurocomputing*, vol. 300, no. 1, pp. 44-69, 2018.

[7] K. Sahnoun, and N. Benabadji, “On-board Satellite Image Compression Using the Fourier Transform and Huffman Coding,” *The Int. Journal of Computational Science, Information Technology and Control Engineering*, vol. 1, no. 1, pp. 17-23, 2014.

[8] D.S. Taubman, and M.W. Marcellin, “JPEG2000: image compression fundamentals, standards, and practice,” New York: Springer, 2002.

[9] G. Held, and T. Marshall, “Data and Image Compression: tools and techniques,” New Jersey: John Wiley & Sons, 1996.

[10] Y.Q. Shi, and H. Sun, “Image and Video Compression for Multimedia Engineering: fundamentals, algorithms and standards,” Boca Raton: CRC Press LLC, 2000.

[11] K. Sayood, “Introduction to Data Compression,” 3rd ed., Burlington: Morgan Kaufmann Publishers, 2006.

[12] G.E. Blelloch, “Introduction to Data Compression,” Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, Oct.2001.

[13] T. Acharya and P. S. Tsai, “JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures,” *John Wiley & Sons, New York*, Oct. 2004.

[14] M. Schindler, (1998), “Practical Huffman coding,” Compression consulting Schindler, http://www.compressconsult.com/huffman/.