

**DICOM compression and decompression method using double cone**

**Método de compressão e descompressão de imagens DICOM utilizando duplo cone**

**Método de compresión y descompresión de imágenes DICOM utilizando doble cono**

Received: 08/31/2020 | Reviewed: 08/31/2020 | Accept: 09/09/2020 | Published: 09/11/2020

**Aratã Andrade Saraiva**

ORCID: <https://orcid.org/0000-0002-3960-697X>

Institute of Mathematics and Computer Sciences, University of São Paulo, Brazil

E-mail: [aratasaraiva@gmail.com](mailto:aratasaraiva@gmail.com)

**Marcos Soares de Oliveira**

ORCID: <https://orcid.org/0000-0003-2389-3334>

Department of Computing and Mathematics, University of São Paulo, Brazil

E-mail: [marcossoares10@usp.br](mailto:marcossoares10@usp.br)

**Joao Batista Neto**

ORCID: <https://orcid.org/0000-0003-1745-1177>

Institute of Mathematics and Computer Sciences, University of São Paulo, Brazil

E-mail: [jbatista@icmc.usp.br](mailto:jbatista@icmc.usp.br)

**Abstract**

The increase of information in the medical environment caused by digital imaging settings is notable. The search and use of these technological tools aimed at medicine require a greater availability of storage, generating increasing costs. In medicine, together with information technology, there is a format of images used in exams, diagnostics, tomography, among others. This format, entitled DICOM, was created in order to standardize uses in medical devices for exam answers. An open question is the compression of DICOM data, in order to maintain quality, maintaining high rates of compression. This presents a new method for compressing and decompressing DICOM data using a dual cone bijector function and a video codec, called DC (Double Cone). This work offers 3 changes to the DC method (DC1, DC2 and DC3). The results obtained with a new technique show that the compression, although with loss, has a similarity index very close to the original image (SSIM = 0.99), and an accuracy ratio equal to 69.51, in the better case. The better performing version was the DC2.

**Keywords:** Double cone; Compression; Decompression; Codecs; DICOM.

## Resumo

O aumento de informações no ambiente médico provocado pelas modalidades de imageamento digital é notável. A procura e uso dessas ferramentas tecnológicas voltadas à medicina demandaram uma maior disponibilidade de armazenamento, gerando crescentes custos. Na medicina, em conjunto com a informática, existe um formato de imagens utilizado em exames, diagnósticos, tomografia, entre outros. Esse formato, intitulado DICOM, foi criado com o intuito de padronizar usos em aparelhos médicos para visualização de exames. Uma questão ainda em aberto é a compressão de dados DICOM, de forma a manter a qualidade, mantendo altas taxas de compactação. Esta tese apresenta um novo método para a compressão e descompressão de dados DICOM por meio de uma função bijetora de duplo cone e um codec de vídeo, intitulado DC (Duplo Cone). Este trabalho propõe 3 variações do método DC (DC1, DC2 e DC3). Os resultados obtidos com a nova técnica mostram que a compressão, embora com perda, tem uma taxa de similaridade bem próxima da imagem original (SSIM = 0.99), e razão de compressão igual a 69.51, no melhor caso. A versão de melhor desempenho foi a DC2.

**Palavras-chave:** Duplo Cone; Compressão; Descompressão; *Codecs*; DICOM.

## Resumen

El aumento de información en el entorno médico provocado por las modalidades de imagen digital es notable. La búsqueda y uso de estas herramientas tecnológicas dirigidas a la medicina exigió una mayor disponibilidad de almacenamiento, generando costos crecientes. En medicina, junto con la tecnología de la información, existe un formato de imagen que se utiliza en exámenes, diagnósticos, tomografías, entre otros. Este formato, denominado DICOM, se creó con el fin de estandarizar los usos en dispositivos médicos para visualizar exámenes. Una cuestión abierta es la compresión de datos DICOM, con el fin de mantener la calidad mientras se mantienen altas tasas de compresión. Esta tesis presenta un nuevo método para la compresión y descompresión de datos DICOM mediante una función de biyector de doble cono y un códec de vídeo, denominado DC (Double Cone). Este trabajo propone 3 variaciones del método DC (DC1, DC2 y DC3). Los resultados obtenidos con la nueva técnica muestran que la compresión, aunque con pérdida, tiene una tasa de similitud muy cercana a la imagen original (SSIM = 0.99), y una relación de compresión igual a 69.51, en el mejor de los casos. La versión con mejor rendimiento fue DC2.

**Palabras clave:** Doble Cono; Compression; Descompresión; *Codecs*, DICOM.

## 1. Introduction

The 1980s were marked by an expansion of digital medical imaging techniques, and the consequent need to standardize access to them. In this task, they joined National Electrical Manufacturers Association (NEMA) and American College of Radiology (ACR), forming a committee whose objective was to develop standards for the interconnection of devices generating digital images (Bhagat & Atique, 2012; F. Liu, Hernandez-Cabronero, Sanchez, Marcellin, & Bilgin, 2017).

The effort to establish a standard led, in 1985, the emergence of a first version, strengthened by the advance of telecommunications and the resulting increase in the flow of data and medical images. Eight years later, the committee launched the third and most significant version, called Digital Imaging and Communications in Medicine (DICOM). From then on, the committee itself was also called DICOM (Huang, 2019).

Since then, DICOM has become accepted worldwide, characterized by being a protocol not only for exchanging data in general, but also for images over the network or physical medium. It continues to evolve to ensure the compatibility of new devices and medical modalities (Ndong et al., 2015).

The growing volume of data generated by medical imaging modalities, such as computed tomography (CT) scanners, Magnetic Resonance Imaging (MRI), Computed Radiography (CR), Digital Radiography (DR), Ultrasound (US), Nuclear Medicine (NM), Mammography and Digital Angiography have fostered discussions, especially regarding the need for storage (Huang, 2019; Rahmat et al., 2019).

In the medical field, data storage is essential. Only the prolonged and continuous storage of medical images can allow robust clinical studies, capable of identifying changes over a period and proposing future treatment or diagnostic actions. In addition, new techniques and ideas, yet to be created, may in the future reveal information and generate knowledge about the data stored today (Dash, Shakyawar, Sharma, & Kaushik, 2019).

Although the cost gradually decreases, storage expenses tend to increase, since the production of medical data grows at a rate much greater than the reduction in cost (Langer, 2011). In addition, the cost of operating a high-performance digital environment has increased, aggravated by factors such as the migration of mandatory data and long retention periods or indefinite periods, to digital images, due to the laws in force in each country (De Macedo, Von Wangenheim, & Dantas, 2015).

Another factor to be considered is the expectation of remote transmission and accessibility to data by the medical community that requires access, often in real time, to images with prolonged storage (Aldemir, Tohumoglu, & Selver, 2019; El Jaouhari et al., 2019; Parikh, Ruiz, Kalva, Fernandez-Escribano, & Adzic, 2018). Such a procedure will have a direct impact on the access of such data on mobile devices, which today are ubiquitous and very heterogeneous (B. J. Erickson, 2002).

Such expectations and challenges can be met through the application of image compression and decompression techniques, which are capable of preserving all clinically relevant information, even if this implies some loss of information degradation (Aldemir et al., 2019; B. J. Erickson, 2002; Ridley, 1997; Schlupkothen, 2012; Wong, Huang, Zaremba, & Gooden, 1995).

With the advancement of the means of medical examination by image and the constant flow of diagnoses performed, the demand for storage in hospital centers for such purposes is essential. Because the average size occupied by a DICOM image is large, due to the information contained and high details in the image, it is extremely important to use compression and decompression to improve space management (Bui, Chang, Li, Hsu, & Chen, 2016; Shaiboun & Shaheen, 2016).

Image compression used the Huffman coding technique, which is already widespread in the literature (Dubey & Singh, 2012; Saravanan & Ponalagusamy, 2009; Singh, Khehra, & Kohli, 2019). In (Rahmat et al., 2019), a method of compression and decompression of DICOM images is proposed, without loss, through the Huffman coding method. Experiments are performed using different types of DICOM images, and performance analysis is provided in terms of compression rate and compression/decompression time. The results indicate a space saving of 72.98 %. The results also show a limitation of Huffman encoding, in some cases the compression rate is less than 1 (ratio  $<1$ ), making the size of the compressed file larger than the original.

The literature shows the popularity and potential of the wavelet transform in medical image compression (Balasamy & Ramakrishnan, 2019; Kadam & Rathod, 2019; Tackie Ammah & Owusu, 2019). In the study by (Zheng et al., 2019), a new hybrid image compression algorithm based on shearlet and wavelet is presented. In this method, the region of interest of the image was compacted without loss by shear transform, and the background region of the image was compacted with losses by the wavelet transform.

The research developed by (Kasban & Hashima, 2019) aims to provide a proposed method for compressing radiographic images. This method consists of separating the image

into two segments: Region of Interest and background. Then, the background of the image is compressed at the maximum compression rate using image pyramid compression, followed by the vector quantization loss compression technique, based on the Generalized Lloyd Algorithm method. Next, the Region of Interest is compressed using the Huffman Code. Finally, the compressed image is obtained by combining the compressed background and the compressed Region of Interest. However, the authors used only two metrics to measure the quality of compression: normalized cross-correlation (NCC) and the structural similarity index (SSIM), leaving aside important metrics such as PSNR and SNR. It is also worth mentioning that they use the Huffman compression method and the performance of the method is offered in comparison with the Huffman coding only, which makes the real contribution of the research inconclusive.

In the work of Baraskar (N. Baraskar & R. Mankar, 2020), a comparative survey of different types of Wavelets is carried out, (Biorthogonal, Haar, Daubechies, Coiflets, Symlets, Reverso Biorthogonal, Discrete Meyer). In order to analyze the efficiency of different types of Wavelets and determine the best one. The performance of the implemented method is evaluated using some essential criteria: the compression rate obtained, the compression gain and the quality of the reconstructed image using PSNR, MSE and SNR. The generated result showed that Wavelets Biorthogonal offers better compression size, compression rate and compression gain, but image quality parameters such as PSNR and MSE are degraded.

The compression and decompression method proposed in this work is to attend hospital-based DICOM volumes. The compression combines a transformation based on the double cone, which consists of a growing bi-directional continuous function, and use of codec for high definition video. Decompression is the reverse process. The method contributes to obtaining a high compression ratio with an acceptable level of similarity.

The work defends the hypothesis that a compression method based on a dual cone bijector function and a video codec can produce compression and decompression in DICOM data with better compression rates, signal noise and similarity.

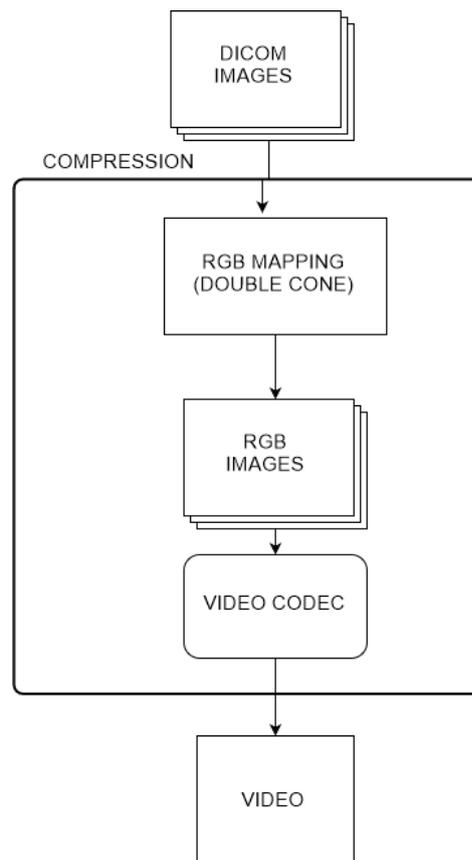
## **2. The Proposed Method**

This section presents the proposed method for compression and decompression of DICOM images entitled as Double Cone (DC). The diagrams of the Figures 1 and 2 present,

respectively, the constituent elements of compression and decompression.

Common to both stages is the process called DC. This process is responsible for performing all the necessary conversions between the color models to encode and decode the images. As it is a key element in the doctoral proposal, it will be presented in detail in the subsection 2.1.

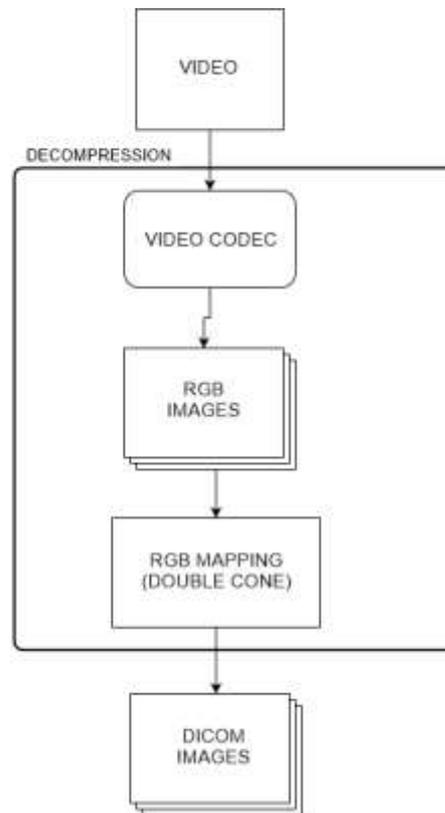
**Figure 1.** Diagram A: Process of converting the DICOM image to RGB with double cone and compression by video codec.



Source: Author.

As illustrated in Figure 1, compression is a process resulting from combining the Double Cone with a video codec. From a set of DICOM images (16-bit pixels), double cone process converts to 3-channel (RGB) images through the attribution of pseudo colors. A video codec is then used to compress the data, producing a video as the output, which is the combination of all the input images. For inverse process, decompression, is used process illustrated in Figure 2 e detailed in subsection 2.2.

**Figure 2.** Diagram B: Process of extracting images from the input video and converting to DICOM image using the double cone.



Source: Author.

The Figure 2, in turn, illustrates the decompression process. From the compressed data (RGB video of pseudo colors), a video codec is used to decompress the data, which are later converted from RGB to a single 16-bit channel, which form the images of the DICOM standard.

Codecs are elements used to encode and decode media files, that is, they compress the original format, favoring storage, and decompress at the time of reproduction, transforming it again into an image or audio. In this paper, the terms encoding, encoder and decoding, decoder will be adopted interchangeably with the terms compression and decompression to refer to the actions and processes of compression and decompression, respectively.

The subsections 2.2 and 2.3 will describe in detail the compression and decompression processes, while the subsection 2.4 presents 3 versions of the proposed method, each corresponding to the combination of Double Cone function with a specific codec image.

### **2.1. Double Cone**

The double cone is the essential part of the compression and reconstruction method

proposed in this work. It consists of a bijector function  $f:A \rightarrow B$  for converting pixels into an image. In it, each element of the  $A$  set (a 16-bit value) is exactly paired with an element of the  $B$  set (an RGB triple).

A bijection of the set  $A$  to the set  $B$  also defines an inverse function  $f'$  from  $B$  to  $A$ . In other words, the double cone allows pixels represented as 16-bit intensities to be converted to a triple RGB and vice versa, with no unpaired elements in both sets. In this work, the process of converting 16 bits to RGB is defined, as a process of attributing pseudo colors.

Formally,  $f:A \rightarrow B$  can be defined as a function  $f(I_p) = Y$ , where  $I_p$  is a pixel belonging to the image  $I$ , represented by an integer of 16 bits, and  $Y$  is a triple of the HSL color model. The *Hue* is calculated by:

$$Hue = \begin{cases} \text{if } L \leq 1 \\ \quad ((\sqrt{L * 2})/2) * \log_2(\Delta) \\ \text{else} \\ \quad ((2 - \sqrt{2 - L * 2})/2) * \log_2(\Delta) \end{cases} \quad (1)$$

where Lightness  $L$  is computed by:

$$L = I_p / \Delta \quad (2)$$

where  $\Delta$  is given by:

$$\Delta = \max(I_n) - \min(I_n) \quad (3)$$

and  $\max(I_n)$  and  $\min(I_n)$  represent the largest and smallest pixel among all images  $I_n$ , respectively, and, finally,  $n$  is the number of images in the set.

Assuming maximum saturation ( $S = 1$ ), the HSL to RGB color model is converted using the equations 4 (a-f). The resulting image is then submitted to the compression process by a codec, as described in the subsection 2.2.

$$C = (1 - |2 * L| - 1) * S \quad (4a)$$

$$X = C * (1 - |(\frac{Hue}{60^\circ})_{mod2} - 1|) \quad (4b)$$

$$m = L - \frac{C}{2} \quad (4c)$$

$$L = \frac{(C_{max} + C_{min})}{2} \quad (4d)$$

$$(R', G', B') = \begin{cases} \text{if } 0^\circ \leq Hue \leq 60^\circ \\ \quad (C, X, 0) \\ \text{else if } 60^\circ \leq Hue \leq 120^\circ \\ \quad (X, C, 0) \\ \text{else if } 120^\circ \leq Hue \leq 180^\circ \\ \quad (0, C, X) \\ \text{else if } 180^\circ \leq Hue \leq 240^\circ \\ \quad (0, X, C) \\ \text{else if } 240^\circ \leq Hue \leq 300^\circ \\ \quad (X, 0, C) \\ \text{else if } 300^\circ \leq Hue \leq 360^\circ \\ \quad (C, 0, X) \end{cases} \quad (4e)$$

$$(R, G, B) = ((R' + m) * 255, (G' + m) * 255, (B' + m) * 255) \quad (4f)$$

---

**Algorithm 1:** 16-bit conversion to RGB in pseudo color

---

**Input:** 16bitPixel,  $\Delta$ ;  
**Output:** r,g,b;  
 $L \leftarrow 16bitPixel / \Delta$ ;  
 $S \leftarrow 1$ ;  
 $Hue \leftarrow L * 2$ ;  
**if**  $Hue \leq 1$  **then**  
    $Hue \leftarrow \sqrt{Hue}$ ;  
**else**  
    $Hue \leftarrow 2 - \sqrt{2 - Hue}$ ;  
 $Hue \leftarrow Hue / 2$ ;  
 $Hue \leftarrow Hue * \log_2(\Delta) / 2$ ;  
 $r, g, b \leftarrow HSLtoRGB(Hue, S, L)$ ;  
**return**  $r, g, b$ ;

---

Source: Author.

The algorithm 1 describes the process of calculation of the Double Cone  $f$  function, which converts 16-bit values to an RGB triple.

The inverse process of the Double Cone, that is, the conversion of the RGB triple into a 16-bit pixel (or the calculation of the  $f'$  function) can be described as follows: an RGB triple

is converted to the HSL color model by means of the equations 5(a-c). The 16-bit pixel  $I_p$  can be easily computed by the equation 6.

$$\begin{aligned} R' &= \frac{R}{255} \\ G' &= \frac{G}{255} \\ B' &= \frac{B}{255} \end{aligned} \quad (5a)$$

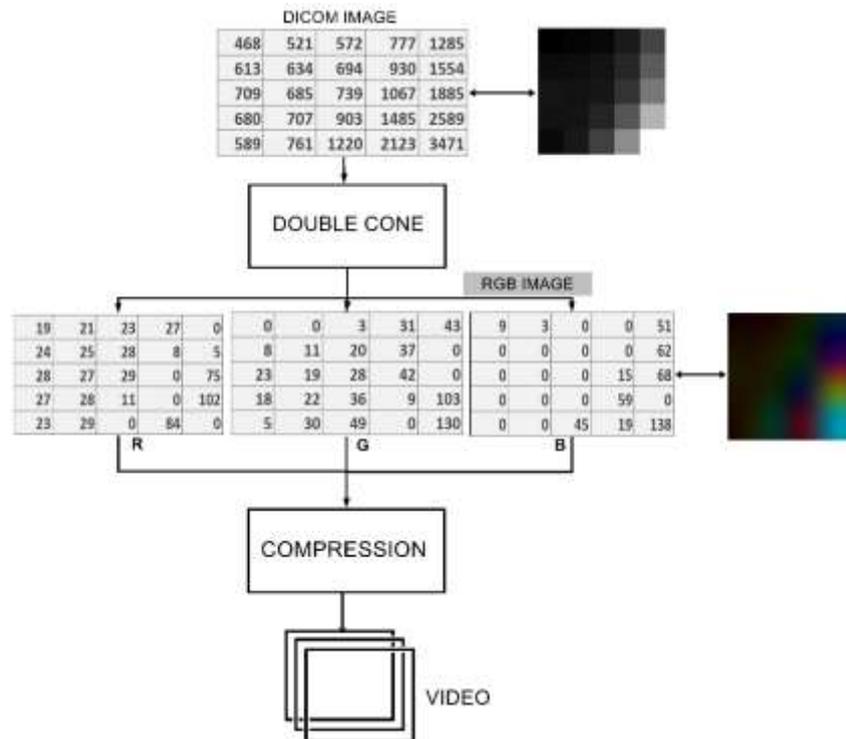
$$\begin{aligned} Cmax &= \max(R', G', B') \\ Cmin &= \min(R', G', B') \\ \Delta &= Cmax - Cmin \end{aligned} \quad (5b)$$

$$L = \frac{(Cmax + Cmin)}{2} \quad (5c)$$

$$I_p = L * \Delta \quad (6)$$

The Figures 3 and 4 illustrate a numerical representation of the Double Cone in the compression process - converting the DICOM image (16 bits) to an RGB triple - and decompression - converting an RGB triple for the DICOM image, respectively.

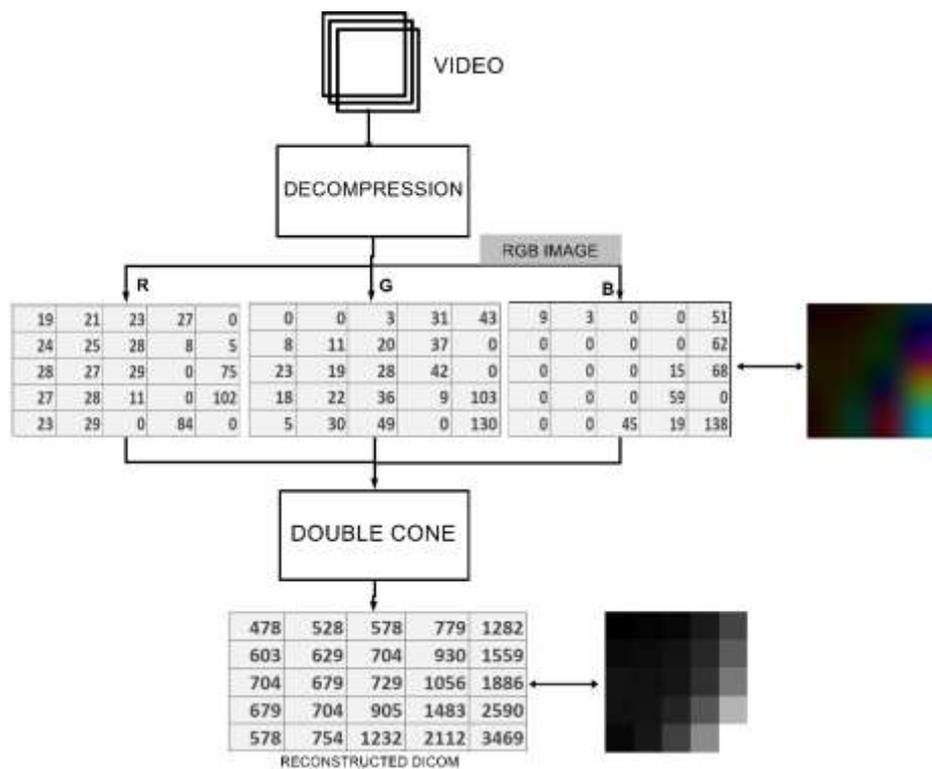
**Figure 3.** Diagram of the  $f$  bijector function, responsible for transforming 16 bits to RGB in the compression process for video.



Source: Author.

The matrix shown at the top of the Figure 3 corresponds to pixels from an arbitrary region of a 16-bit DICOM image. The Double Cone then maps the image in an RGB version, with each channel (R, G and B) consisting of 8 bits, represented in the figure by 3 matrices. The compression process with a video codec is then applied over this image, resulting in the compressed data.

**Figure 4.** Diagram of the  $f'$  bijector function, responsible for transforming RGB to 16 bits in the final DICOM reconstruction process.



Source: Author.

The Figure 4, illustrates the reverse process. From the video the slices are extracted which will result in a compressed image in the RGB format, represented in the figure by 3 matrices, each corresponding to a channel (R, G and B). Then, the reverse process of the Double Cone follows, which reconstructs the final DICOM image.

We said that the proposed Double Cone proposed is a bijector function. The purpose of both Figures 4 and 5 is to empirically show this behavior. Note that the values of the input matrices in Figure 3 and the reconstructed in 4 are very close. The values are not exactly the same for reasons of rounding between the calculations made by the Double Cone (equations 4(a-f) and 5(a-c)) and the conversion to image pixels, which are integer values.

## 2.2. Compression – Encoding

The process illustrated in Figure 1, is based on the principle of the double cone function presented in the subsection 2.1 where the conversion of each image takes place, pixel by pixel of DICOM images in 3 channels of 8 bits with the application of false colors.

Along with the colorization, all the information present in the tags from the DICOM files is extracted and a file is created containing their respective tags and content without the set of pixels in the image, so that it can later be joined to the resulting video file. After colorization, the images created in a video are added and compressed using a video codec.

In the codec encoding process, the libraries contained in the FFMPEG were used. Its use made it possible to create the video where parameters such as the Frame Rate per Second (FPS), the codec to be used and the bitrate are established. The step by step of this process is described in the algorithm 2.

---

**Algorithm 2:** Creating images and compressing.

---

```
Input: dicomPath, imagesPath;  
Output: NewPath;  
images ← duplocone(dicomPath);  
tags ← extractTags(dicomPath);  
saveToImage(imagePath, images);  
saveToTags(imagePath, tags);  
ffmpeg( framerate, imagePath, vcodec, bitrate,  
namefinalvideo.mkv);  
return NewPath;
```

---

Source: Author.

The process described by the algorithm 2 illustrates the sequence of the proposed method. The double cone is initially applied, then the tags are extracted and the images are finally compressed into video using video codecs.

To validate the proposed method and assess the quality of the codec in compressing medical data (Bui et al., 2016), methods based on popularity and the support mechanism offered to DICOM were used: H.264 (Pole & Shriram, 2018), H.265 (Bross, Han, Ohm, Sullivan, & Wiegand, 2012; Parikh et al., 2018) and FFV1 (Venkat & Vaidyanathan, 2019).

### 2.3. Decompression – Decoding

---

**Algorithm 3:** Reconstruction of DICOM

---

```
Input: videoPath, deltaValue  
Output: dicomPath  
ffmpeg(-i video.mkv -qscale : v2  
imagesExtracted/);  
Tags ← getTags(videoPath);  
dicom ← duplocone(imagesExtracted, deltaValue);  
dicom ← insertTags(Tags);  
saveDicom(dicom);  
return dicomPath;
```

---

Source: Author.

The algorithm 3, illustrates the process of decompression represented by the diagram in Figure 2. It starts when a video made up of DICOM images, described in the subsection 2.2, is fed to the frame extraction process using the FFMPEG tool. The double cone processes the extracted frames, reconstructing the original DICOM either with the original pixel values or a value very close to it (this is due to loss of approximation). For reconstruction, it is still necessary to convert the RGB images, extracted from the video, to the HSL color model, as described in the section 2.1. Only the third element, Lightness, is used in the reconstruction process, thus enabling the redistribution of the RGB triples back to the 16-bit image. The final DICOM data is created by inserting the required tags.

### 2.4. The Methods: DC1, DC2 e DC3

The proposed method is divided into three variations of itself. The first, called DC1, is formed by the combination of the double cone and the H.264 codec. The second variation, called DC2, is formed by a double cone and the H.265 codec, while DC3 is formed by the combination of the double cone and the FFV1 codec.

The volumetric image was reconstructed with a 1 mm slice thickness and  $512 \times 512$  image matrix size. The number of frames in the data set comprises 211 slices, covering the patient's skull. Evaluation has also been conducted on images with 3 and 5 mm thickness.

### 3. Metrics for Assessing Compression Quality

Compression aims to reduce the number of bits required to represent an image, removing spatial and spectral redundancies as much as possible. The peak signal-to-noise ratio used to be a measure of image quality (Mrak, Grgic, & Grgic, 2003). The ideal is to obtain the resulting image close to the original. This can be measured quantitatively using the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics.

Data redundancy is the central issue in digital image compression. If  $n_1$  and  $n_2$  represent, respectively, the number of units of information carried in the original image and encoded image, then the compression ratio (CR) can be defined as:

$$CR = \frac{n_1}{n_2} \quad (7)$$

and the data redundancy of the original image can be defined by:

$$DR = 1 - \frac{1}{CR} \quad (8)$$

#### 3.1. Compression rate

The compression ratio is the average number of Bits Per Pixel (BPP) before compression, divided by the number of bits per pixel after compression. For example, if an 8 *bits* image is compressed and then each pixel is represented by a BPP, the 8/1 compression ratio results in 8 or equivalent for a 24 bits image, if the compression rate is 18, the compressed image will be 24/18 resulting in 1.33 BPP (CAR. Canadian association of radiologists., 2011).

#### 3.2. Correlation Coefficient

Correlation Coefficient (CC) is a numerical measure of some type of correlation, meaning a statistical relationship between two variables. Which can be used to indicate the correlation between two signals, as can be seen in the equation 9, where A and B are compared from that a numerical representation between 0-1 is obtained where the worst case of correlation and 1 the best case  $\bar{A}$  and  $\bar{B}$  are the averages of those entries (Lee Rodgers & Alan Nice Wander, 1988).

$$CC = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (9)$$

### 3.3. DICOM compression analysis

The quality of the images stored by health professionals is not always satisfactory. Despite technological advances in imaging equipment and reconstruction algorithms, the presence of noise in images is inevitable and originates in the capture of images even in the process of transmitting them (W. Liu & Lin, 2013).

Noises are random variations in intensity in the image signal, which impair the visualization of details, especially when the object is small and has low contrast. Among the various types of noise are Gaussian noise (white) (Russo, 2003; Toh & Isa, 2010) and impulsive noise (salt and pepper) (Gonzalez & Woods, 2007; Toh & Isa, 2010). In addition, in order to have an image quality evaluation, it is necessary to use metric evaluation methods, such as PSNR and SSIM (Horé & Ziou, 2010), mathematically defined by equations 10 and 11, respectively.

$$PSNR = 10 \log \left( \frac{MAX^2}{MSE} \right) = 20 \log \frac{MAX}{MSE^{\frac{1}{2}}} \quad (10)$$

$$MSE = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} I(x, y) - K(x, y)^2$$

In PSNR, *MAX* represents the maximum possible pixel value in the image, *I* and *K* respectively the first and second images, *x* is the vector that denotes values of *n* number of predictions *y* is a vector representing a number *n* of true values. O Mean Squared Error (MSE) is a measure of quality of an estimator where *I* represent the original image and *K* the distorted image.

From a statistical point of view, the MSE can present problems when used to compare similarity. The main one is that large differences between pixel intensities do not necessarily mean that the content of the images is dramatically different. It is important to note that an MSE value equivalent to 0 indicates perfect similarity. A value greater than 1 implies less similarity and will continue to increase as the average difference between pixel intensities also increases (Saraiva et al., 2019).

When comparing compression codecs, PSNR is an approximation to human perception of the quality of reconstruction. Typical values for PSNR in compression with loss

of image and video are between 30 and 50 dB, as long as the bit depth is 8 bits, the higher the better. For 16-bit data, typical values for PSNR are between 60 and 80 dB (Mauro, 2006; Saraiva et al., 2019). In order to remedy some of the problems associated with MSE for image comparison, we have SSIM described by equation 11:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_1)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (11)$$

where  $\mu_x$  is the average of  $x$ ,  $\mu_y$  is the average of  $y$ ,  $\sigma_x^2$  is the variance of  $x$ ,  $\sigma_y^2$  is the variance of  $y$ ,  $\sigma_{xy}$  is covariance,  $c_1 = (K_1L)^2$  and  $c_2 = (K_2L)^2$  are Variables to stabilize the division with a weak denominator. In which,  $L$  is the dynamic range of pixel values and  $K_1 = 0.01$  and  $K_2 = 0.03$  by default.

The essence of SSIM is to model the perceived change in the structural information of the image, while the MSE is actually estimating the perceived errors. There is a subtle difference between the two, but the results can be great. Unlike the MSE, the SSIM value can vary between -1 and 1, where 1 indicates perfect similarity.

In addition, SSIM is used to analyze small sub-samples instead of the entire image as in MSE. In this way, a more robust approach is obtained, capable of explaining the changes in the image structure, instead of just the perceived change.

It is worth mentioning that the sensitivity of the PSNR is similar to that of the SSIM. However, in images that used the JPEG compression standard, the sensitivity of the SSIM provides better precision, bearing in mind that both measures have a sensitivity slightly similar to JPEG 2000 (Gupta, Srivastava, Bhardwaj, & Bhateja, 2011; Horé & Ziou, 2010).

#### 4. Results

In this section, the results regarding the developed compression and decompression method will be presented. Subsections 4.1, 4.2 and 4.3 present experiments with images acquired in 1, 3 and 5 mm, respectively.

All tables in subsections 4.1, 4.2 and 4.3 have a similar format: the 3 versions of the proposed compression method DC1, DC2 and DC3 are compared to traditional methods of compression using codecs H.264, H.265 and FFV1. Evaluation metrics PSNR, SSIM - decibel values, in addition, CC were used. Such methods are known as full reference, as they consider the original image as a reference. The tables also provide quantitative analysis for the CR metrics, which measure the actual compression in bytes of the data and the DR.

In the vast majority of experiments, the metrics SSIM, PSNR and CR, in this sequence, were chosen to select the best performing methods. The MSE, as mentioned earlier in section 3, can present problems when used to compare similarities and its use is not viable. The CC presents very close values. the CR is more sensitive to changes than the DR. SSIM as a compression metric is more relevant than PSNR, as reported in the section 3.

#### **4.1. Proposed method for volume compression on 1mm thick data**

In this section, the proposed methods DC1, DC2 and DC3 are compared to codecs H.264, H.265 and FFV1 for rates of FPS = 60, 120 and 240. The experiments were replicated for JPEG and PNG images. The experiments were carried out with the public dataset of 106 MB, composed of 211 DICOM images (C., 2018; Clark et al., 2013). Each pixel has 12 bits of information and is stored inside 16 bits containers. Therefore, there is a waste of 25 %, 4 bit storage. The slices are 1 mm thick.

##### **4.1.1. Experiment 1**

In this first experiment, a fixed rate of FPS = 60 was used. The results are presented in the Table 1.

**Table 1.** Results of the reconstruction of DICOM images in JPEG and PNG formats to 1mm thickness and FPS = 60.

JPEG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	60	33.330	0.976	0.999	7.68	0.86
H.264	150000	60	33.330	0.976	0.999	7.68	0.86
H.264	200000	60	33.330	0.976	0.999	7.68	0.86
H.265	N/A	60	33.314	0.975	0.992	64.24	0.98
FFV1	N/A	60	33.330	0.976	0.999	7.73	0.87
DC1	100000	60	46.267	0.972	0.963	4.52	0.77
DC1	150000	60	46.267	0.972	0.963	4.52	0.77
DC1	200000	60	46.267	0.972	0.963	4.52	0.77
DC2	N/A	60	46.472	0.978	0.957	39.25	0.97
DC3	N/A	60	46.234	0.972	0.963	5.02	0.80
PNG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	60	33.332	0.976	1.000	5.43	0.81
H.264	150000	60	33.332	0.976	1.000	5.43	0.81
H.264	200000	60	33.332	0.976	1.000	5.43	0.81
H.265	N/A	60	33.315	0.975	0.992	65.03	0.98
FFV1	N/A	60	33.332	0.976	1.000	6.50	0.84
DC1	100000	60	66.719	0.999	0.997	2.36	0.57
DC1	150000	60	67.693	0.999	0.997	2.20	0.54
DC1	200000	60	67.693	0.999	0.997	2.20	0.54
DC2	N/A	<b>60</b>	<b>66.777</b>	<b>0.998</b>	<b>0.994</b>	<b>37.72</b>	<b>0.97</b>
DC3	N/A	60	79.377	1.000	0.999	1.69	0.41

Source: Author.

For JPEG data in Table 1, it is noted that the PSNR values for the codecs are good (typical PSNR values in lossy compression for videos and images are in the range of 30-50 dB) and regardless of the bitrate variation, the values of H.264 (PSNR = 33.330) and DC1 (PSNR = 46.267) remain constant. The proposed method, in its three versions DC1, DC2 and DC3, presents the best of PSNR with images in JPEG and PNG format, however when observing the SSIM metric with JPEG images, the proposed methods present a lower performance than traditional codecs, on the other hand for the metric CR the methods H.265 and DC2 present the best results in both cases.

It is observed that the DC2 method presents the best performance. This statement is corroborated by some factors: a) high SSIM value produced by the method, for both JPEG and PNG images and b) high compression ratio (high values for DR metric).

#### 4.1.2. Experiment 2

In this experiment, the number of frames was increased to a fixed rate of FPS = 120. The results given in JPEG and PNG are shown in the Table 2.

**Table 2.** Results of the reconstruction of DICOM images in JPEG and PNG formats for 1mm thickness and FPS = 120.

JPEG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	120	33.330	0.976	0.999	7.85	0.87
H.264	150000	120	33.330	0.976	0.999	7.73	0.87
H.264	200000	120	33.330	0.976	0.999	7.68	0.86
H.265	N/A	120	33.311	0.975	0.990	68.38	0.98
FFV1	N/A	120	33.330	0.976	0.999	7.73	0.87
DC1	100000	120	46.267	0.973	0.963	5.30	0.81
DC1	150000	120	46.267	0.972	0.963	4.58	0.78
DC1	200000	120	46.267	0.972	0.963	4.54	0.78
DC2	N/A	120	46.073	0.978	0.952	46.90	0.97
DC3	N/A	120	46.234	0.972	0.963	5.02	0.80
PNG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	120	33.332	0.976	1.000	5.63	0.82
H.264	150000	120	33.332	0.976	1.000	5.46	0.81
H.264	200000	120	33.332	0.976	1.000	5.43	0.81
H.265	N/A	120	33.311	0.975	0.990	68.83	0.98
FFV1	N/A	120	33.332	0.976	1.000	6.50	0.84
DC1	100000	120	59.990	0.999	0.994	4.47	0.77
DC1	150000	120	63.691	0.999	0.996	3.01	0.66
DC1	200000	120	66.623	0.999	0.997	2.38	0.58
DC2	N/A	<b>120</b>	<b>66.600</b>	<b>0.998</b>	<b>0.990</b>	<b>43.08</b>	<b>0.97</b>
DC3	N/A	120	79.377	1.000	0.999	1.69	0.41

Source: Author.

Table 2 compared to the previous experiment (FPS = 60), has a very similar behavior is observed for all the proposed codecs and methods. The proposed methods are better, with respect to the PSNR and SSIM values, especially for PNG images. In terms of data compression, the best results are those shown by the DC2 and H.265 method, for both JPEG and PNG images. Also like the FPS rate = 60, taking into account the SSIM values and the compression ratio, the best method is DC2, when compared to traditional codecs.

### 4.1.3. Experiment 3

In this experiment, the number of frames was increased to a fixed rate of FPS = 240. The results are shown in the Table 3.

**Table 3.** Results of the reconstruction of DICOM images in JPEG and PNG formats to 1mm thickness and FPS = 240.

JPEG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	33.330	0.976	0.999	10	0.90
H.264	150000	240	33.330	0.976	0.999	8.03	0.87
H.264	200000	240	33.330	0.976	0.999	7.85	0.87
H.265	N/A	240	33.311	0.975	0.990	68.38	0.98
FFV1	N/A	240	33.330	0.976	0.999	7.73	0.87
DC1	100000	240	46.266	0.973	0.963	8.76	0.88
DC1	150000	240	46.268	0.973	0.963	6.54	0.84
DC1	200000	240	46.267	0.973	0.963	5.32	0.81
DC2	N/A	240	46.073	0.978	0.952	46.90	0.97
DC3	N/A	240	46.234	0.972	0.963	5.02	0.80
PNG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	33.332	0.976	0.999	9.05	0.88
H.264	150000	240	33.332	0.976	0.999	6.62	0.84
H.264	200000	240	33.332	0.976	1.000	5.66	0.82
H.265	N/A	240	33.311	0.975	0.990	68.83	0.98
FFV1	N/A	240	33.332	0.976	1.000	6.50	0.84
DC1	100000	240	54.641	0.996	0.989	8.21	0.87
DC1	150000	240	57.734	0.998	0.992	5.76	0.82
DC1	200000	240	60.142	0.999	0.994	4.39	0.77
DC2	N/A	<b>240</b>	<b>59.976</b>	<b>0.998</b>	<b>0.990</b>	<b>43.08</b>	<b>0.97</b>
DC3	N/A	240	79.377	1.000	0.999	1.69	0.41

Source: Author.

Table 3 shows a similar behavior in which is observed for all codecs and methods proposed. The DC1, DC2 and DC3 methods continue to have better PSNR and SSIM rates, especially for PNG images. However, there is a greater variability in this rate for the proposed method DC1, as the bitrate rate increases. The DC1 method is significantly better for bitrate = 200000 (PSNR = 60.142) when compared to bitrate = 100000 (PSNR = 54.641). In terms of data compression, the best results continue with the DC2 and H.265 methods, for both JPEG and PNG images.

Similar to FPS = 60 and FPS = 120, while taking into account the values of SSIM and

CR, the best method is DC2.

#### 4.1.4. Remarks

From the data shown in the Tables 1,2 and 3 it can be seen that the H.264 codec is insensitive to variations in both the bitrate rate and in the FPS rate. In addition, it always shows inferior results for both the PSNR and SSIM metrics in PNG format. The same is observed for the compression ratios (CR). The FFV1 codec behaves similarly to H.264.

The H.265 codec, on the other hand, has high compression ratios, although it displays PSNR and SSIM values below those obtained by the proposed methods DC1, DC2 and DC3, especially when applied to PNG images.

From the variations of the proposed methods, DC2 is the one that best combines compression rate (better CR values) and compression quality (high SSIM values). It is important to note that SSIM values are relevant to the nearest hundredth. Hence, DC2 (SSIM = 0.998) and DC1 (SSIM = 0.999) methods have virtually the same compression quality.

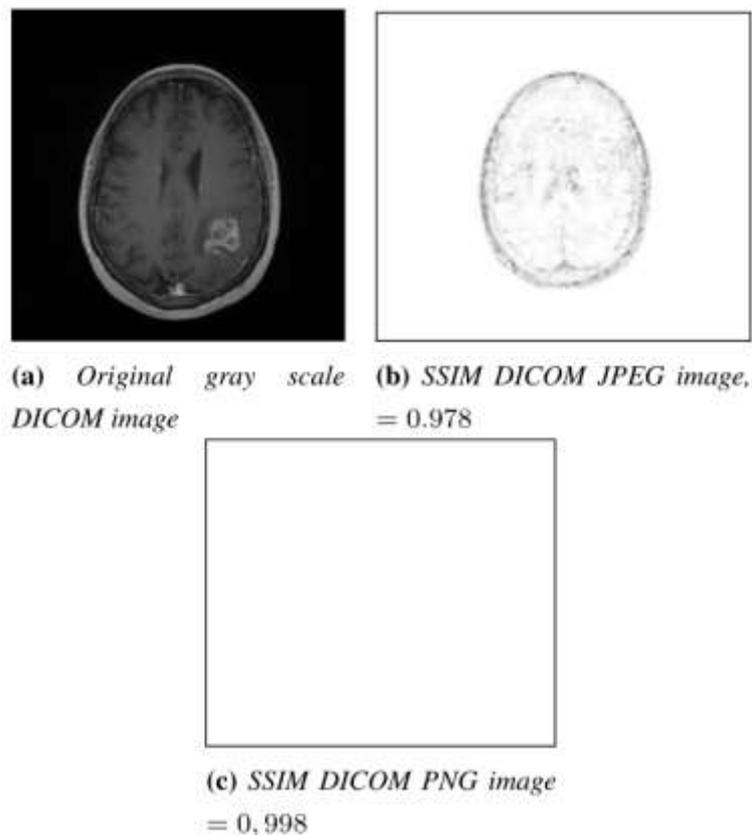
Also note that, for FPS = 120 and FPS = 240, and increasing bitrate values, the DC1 method produces increasing values for the SSIM metric. Therefore, it can also be said that the DC1 method with the best performance is the one applied to bitrate = 200000.

Some important conclusions can be drawn from the graphs and tables above. Are they:

- The H.265 encoder, despite generating the best compression ratio among all analyzed, has a notable loss of information.
- DC3 shows the best result when applied to images in PNG format and has shown no loss according to the SSIM metric. However, the size of the compressed file is not competitive with competing methods.
- DC1 and DC2 have significant results aiming at the ratio between DR and similarity metrics.

A qualitative analysis of the SSIM metric, for images in JPEG and PNG format, is given below and is illustrated in the Figures 5b and 5c. The images represent the perception of degradation in the image after compression and decompression, with changes in structural information and it was made under the 200000 and FPS = 240 bitrate conditions.

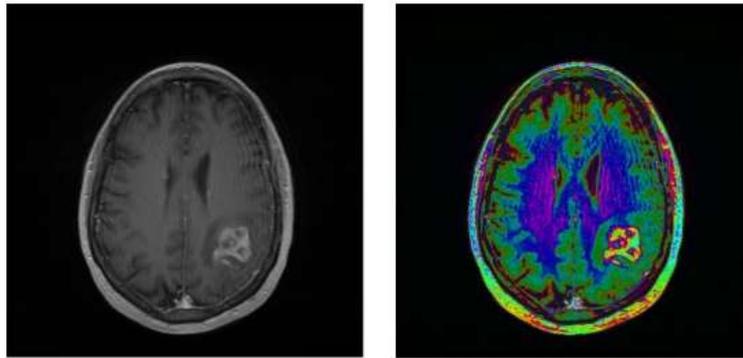
**Figure 5.** Graphical representation of the original DICOM image with the results of the SSIM metric in compression, with image in JPEG format (Figure 5b) and with image in PNG format (Figure 5c).



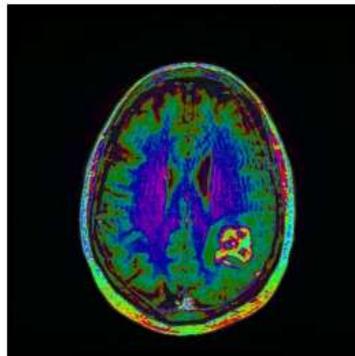
Source: Author.

From the SSIM representation of the Figures 5b and 5c together with the Table 3, it can be seen that when using the JPEG format in compression it is possible to make the file relatively smaller. However, there was a notable loss in the similarity of the metrics, whether the methods with or without double cone. On the other hand, in compression with the PNG format it is noticed that the fidelity with the original image increases and the compression ratio tends to decrease. It was noticed that the images in PNG format with double cone obtained a higher similarity rate, being observed in the Figure 5c where it is not possible to visualize the noise.

**Figure 6.** Original DICOM image (Figure 6a) and image with double cone function applied, in JPEG format (Figure 6b) and in PNG format (Figure 6c).



(a) *Original gray scale DICOM image*      (b) *Pseudo color image in JPEG format*

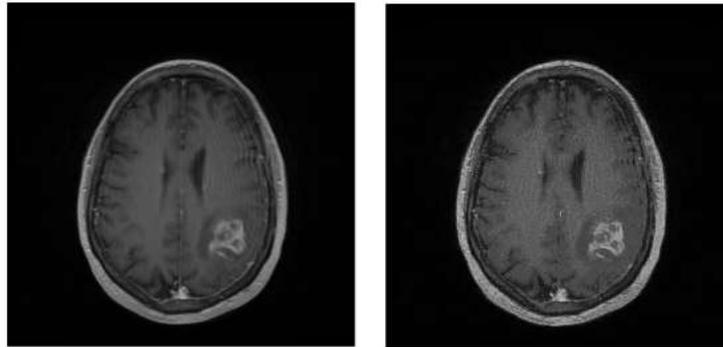


(c) *Pseudo color image in PNG format*

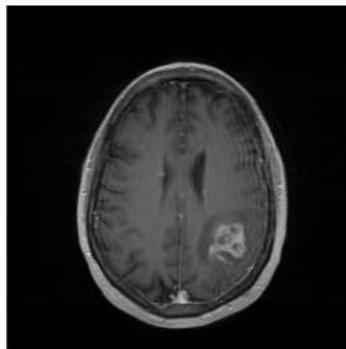
Source: Author.

The Figure 6a represent the original image, while the Figures 6b and 6c represent the JPEG and PNG image when applied color with double cone, note that there is not a big noticeable change between the two formats, this only becomes clearer when the image is reconstructed, illustrated by Figure 7.

**Figure 7.** Image reconstructed from JPEG and PNG format.



(a) *Original gray scale DICOM image*      (b) *Reconstructed DICOM image from JPEG format.*  
*SSIM = 0.978*



(c) *Reconstructed DICOM image in PNG format.* *SSIM = 0.998*

Source: Author.

The Figures 7b and 7c represent the reconstructed grayscale image after decompression to the DICOM standard. In Figure 7b is noticeable a noise after this process while in Figure 7c reconstruction shows the best result between them.

#### **4.2. Proposed method for volume compression on 3mm thick data**

In this experiment, the performance of the methods will be evaluated for volumes of 3mm and FPS = 240 only. The dataset is also public (Schmainda & Prah, 2018) and contains 60 DICOM images of approximately 515 KB each, totaling 30.1 MB. The evaluation for FPS = 60 and FPS = 120 were ignored due to the uniform behavior presented by the methods for all of them. The results are presented in the Table 4.

**Table 4.** Comparison of results of the reconstruction of the JPEG 3D image compared to the original DICOM (3mm).

JPEG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	46.667	0.998	0.998	13.25	0.92
H.264	150000	240	46.667	0.998	0.998	10.23	0.90
H.264	200000	240	46.485	0.998	0.989	8.62	0.88
H.265	N/A	240	46.614 5	0.997	0.986	92.61	0.98
FFV1	N/A	240	46.667	0.998	0.998	8.36	0.88
DC1	100000	240	58.464	0.998	0.955	11.31	0.91
DC1	150000	240	58.457	0.998	0.955	8.67	0.88
DC1	200000	240	58.243	0.998	0.953	7.04	0.85
DC2	N/A	240	58.023	0.998	0.939	73.23	0.98
DC3	N/A	240	58.223	0.998	0.952	7.27	0.86
PNG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	46.682	0.998	0.999	9.90	0.89
H.264	150000	240	46.683	0.998	0.999	7.03	0.85
H.264	200000	240	46.483	0.998	0.990	5.79	0.82
H.265	N/A	240	46.614	0.997	0.986	93.18	0.98
FFV1	N/A	240	46.683	0.998	1.000	6.54	0.84
DC1	100000	240	67.834	0.999	0.988	8.31	0.87
DC1	150000	240	70.386	0.999	0.991	5.57	0.82
DC1	200000	240	71.806	0.999	0.992	4.53	0.77
DC2	N/A	<b>240</b>	<b>67.625</b>	<b>0.998</b>	<b>0.980</b>	<b>69.51</b>	<b>0.98</b>
DC3	N/A	240	86.603	1.000	0.998	2.11	0.52

Source: Author.

In the Table 4 as compared to the experiment with a volume of 1 mm, excellent values are observed for the SSIM metric for images in JPEG and PNG format (SSIM = 0.99). The PSNR values are quite close (PSNR = 46.6) for traditional codecs. The DC1, DC2 and DC3 methods continue to have better PSNR rates, especially for PNG images. The proposed DC1 method again exhibits an increase in the PSNR value as the bitrate rate increases reaching the value of (PSNR = 71.806) to bitrate = 200000. In this experiment, the PSNR of DC1 is slightly more significant than DC2 (PSNR  $\cong$  67.625). When observing data compression, the best performance methods are H.265, followed by DC2, for both JPEG and PNG images.

However, if we consider the combined values of SSIM and CR, the most efficient method is DC2.

#### 4.3. Proposed method for volume compression on 5mm thick data

This last experiment evaluates the performance of the methods for volumes of 5mm and a fixed rate of FPS = 240. As in the previous experiments, a public dataset (B. Erickson,

Akkus, Sedlar, & Kofiatis, 2017) containing 24 images of 166 KB each is used, totaling a volume of size 3.88 MB. The results are presented in the table 5, for JPEG and PNG input data.

**Table 5.** Comparison of results of the reconstruction of the JPEG 3D image compared to the original DICOM (5mm).

JPEG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	43.333	0.759	0.904	27.71	0.96
H.264	150000	240	43.333	0.759	0.904	27.32	0.96
H.264	200000	240	43.308	0.757	0.904	27.51	0.96
H.265	N/A	240	43.327	0.759	0.487	44.09	0.97
FFV1	N/A	240	43.333	0.759	0.904	27.51	0.96
DC1	100000	240	56.049	0.996	0.940	3.18	0.68
DC1	150000	240	56.047	0.996	0.940	2.55	0.60
DC1	200000	240	55.885	0.996	0.938	2.32	0.56
DC2	N/A	240	56.506	0.995	0.943	20.00	0.95
DC3	N/A	240	55.870	0.996	0.938	2.50	0.60
PNG							
<i>Encoder</i>	<i>Bitrate</i>	<i>FPS</i>	<i>PSNR</i>	<i>SSIM</i>	<i>CC</i>	<i>CR</i>	<i>DR</i>
H.264	100000	240	43.337	0.759	0.962	11.65	0.91
H.264	150000	240	43.337	0.759	0.963	11.34	0.91
H.264	200000	240	43.308	0.757	0.963	6.65	0.84
H.265	N/A	240	43.329	0.759	0.573	44.09	0.97
FFV1	N/A	240	43.334	0.759	0.965	2.30	0.56
DC1	100000	240	72.692	0.999	0.998	1.94	0.48
DC1	150000	240	76.406	1.000	0.999	1.51	0.34
DC1	200000	240	78.516	1.000	0.999	1.33	0.25
DC2	N/A	<b>240</b>	<b>70.601</b>	<b>0.998</b>	<b>0.994</b>	<b>18.83</b>	<b>0.94</b>
DC3	N/A	240	93.524	1.000	1.000	0.98	-0.01

Source: Author.

In Table 5, low values are observed for the SSIM metric for traditional codecs ( $\cong 0.75$ ) in relation to the proposed methods ( $\cong 1$ ). The same can be observed for the PSNR metric of double cone methods, especially for images in PNG format. The DC3 method has a high PSNR value (93.524) for PNG images. Still with respect to PSNR, the DC1 method presents considerable values, which grow as the bitrate increases (values between 78.516 and 72.692). The DC2 method has the lowest PSNR values for PNG images (70.601), still much higher than those presented by traditional codecs.

On the other hand, there is a very high value of the CR metric (18.83) for the DC2 method, in PNG images, when compared to the other double cone methods. The compression ratio for DC2 is about 9 times higher than the best of the other techniques proposed with Double Cone, DC1 with bitrate = 100000 and CR = 1.94. It is also possible to observe a small

improvement in the values of the CC metric for the DC1, DC2 and DC3 methods ( $CC \cong 0.94$ ), and like the other experiments, it is observed that when considering the combined values for the SSIM and CR metrics, the most efficient method remains DC2.

## 5. Conclusion

After analyzing the results in DICOM images with a thickness of 1, 3 and 5 mm, it was observed that the best lossy compression method is DC2. Among all the proposed alternatives and traditional codecs, it is the one that best presents similarity between the original and compressed data (SSIM metric) and compression rates (CR metric).

The DC2 method combines the proposal of a bijector function called Double Cone for conversion and reconstitution of DICOM data, together with H.265 video codec.

The performance analysis of the proposed methods and comparison with traditional methods, adopted the following metrics, in this order of importance: SSIM, PSNR and CR. As explained earlier, this sequence takes into account the similarity (SSIM) between the original data and the compressed data, as the main factor. This is due to the fact that in medical applications, it is necessary to have information with the least possible error.

The need to have a value of  $SSIM = 1$ , or close to it, is relevant. In it, it is possible to observe that for values  $SSIM = 0.99$  there are, qualitatively, no errors visible to the human eye while the values of  $SSIM = 0.97$  already have visual errors. This is of fundamental importance for the medical community, particularly for specialties that perform diagnoses based on images. It appears that in all experiments with the DC2 method, when using data in PNG format, the results for the SSIM metric are in the order of  $\geq 0.99$ . Clinically, this is an acceptable level of similarity for exam interpretation.

When analyzing the CR compression ratio for the DC2 method, it is noted that there is a high value coming from the H.265 codec. Such a property makes it a much more attractive method than DC1 and DC3.

In general, better compression results are also observed when using the PNG format. In addition, it was observed that by increasing the thickness, the SSIM metric values for the DC1, DC2 and DC3 methods became even more expressive when compared to compression using traditional codecs.

Therefore, it appears that the DC2 method based on a dual cone bijector function and a video codec (H.265) produced results in DICOM data with better compression rates, signal

noise and similarity, considering that the combination of such metrics it is indispensable for application in a clinical environment.

## 6. Future perspectives

As future work it is proposed to integrate an application to the method, to perform compression and decompression in an automated and simplified way. Additionally, the use of other codecs, such as VP9 and AV1, can be investigated and compared with the proposed method, in order to improve the similarity and increase the compression rate. It is believed that the test on other organs, with and without pathologies, is of relevance, as well as the test in thicknesses less than 1mm.

Measure the processing time of the compacting and unpacking processes. The main concern of this work was to quantitatively evaluate the methods in relation to their capacity to provide compression rate and low loss of information.

Check the tolerance for errors, due to distortions in the compressed data flows, according to specific future legislation in the hospital area.

For a more significant validation of the image compression, it is suggested to perform specificities of the receiver (health specialist in images). As an image analyzer, the human visual system is considered.

For greater security guarantee, developing a compression together with digital watermark should be applied to the image. Possible adulterations can be verified.

For even greater loss reduction, it is suggested to add error reduction algorithms such as Reed-Solomon and Viterbi (Li, Chen, & Sun, 2020; Wang, Yan, Pedrycz, Zhou, & Li, 2020) to the method.

## References

- Aldemir, E., Tohumoglu, G., & Selver, M. A. (2019). Performance Evaluation of Lossless Compression Algorithms for Medical Images. *2019 27th Signal Processing and Communications Applications Conference (SIU)* 1–4. IEEE.
- Balasamy, K., & Ramakrishnan, S. (2019). An intelligent reversible watermarking system for authenticating medical images using Wavelet and PSO. *Cluster Computing*.

Bhagat, A. P., & Atique, M. (2012). Medical images: Formats, compression techniques and DICOM image retrieval a survey. *2012 International Conference on Devices, Circuits and Systems, ICDCS 2012*.

Bross, B., Han, W. J., Ohm, J. R., Sullivan, G. J., & Wiegand, T. (2012). High efficiency video coding (HEVC) text specification, draft 8, no. *JCTVC-J1003\_d7, ISO/IEC Draft Int'l. Standard*.

Bui, V., Chang, L. C., Li, D., Hsu, L. Y., & Chen, M. Y. (2016). Comparison of lossless video and image compression codecs for medical computed tomography datasets. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*.

C., N. C. I. C. P. T. A. (2018). Radiology data from the clinical proteomic tumor analysis consortium glioblastoma multiforme [cptac-gbm] collection [data set]. *the cancer imaging archive*. CAR. Canadian association of radiologists. (2011). Canada health infoway. Retrieved from <https://car.ca/wp-content/uploads/Compression-in-Digital-Imaging-2011.pdf>

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., et al. (2013). The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6), 1045–1057. Springer.

Dash, S., Shakyawar, S. K., Sharma, M., & Kaushik, S. (2019). Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*.

Dubey, V. G., & Singh, J. (2012). 3D medical image compression using Huffman encoding technique. *International Journal of scientific and research publications*, 2(9). Citeseer.

Erickson, B., Akkus, Z., Sedlar, J., & Kofiatidis, P. (2017). Data from LGG-1p19qDeletion. *The Cancer Imaging Archive*.

Erickson, B. J. (2002). Irreversible compression of medical images. *Journal of Digital Imaging*.

Gonzalez, R. C., & Woods, R. E. (2007). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©2006.

Gupta, P., Srivastava, P., Bhardwaj, S., & Bhateja, V. (2011). A modified PSNR metric based on HVS for quality assessment of color images. *Proceedings of the 2011 International Conference on Communication and Industrial Application, ICCIA 2011*.

Horé, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. *Proceedings - International Conference on Pattern Recognition*.

Huang, H. K. (2019). *PACS-Based Multimedia Imaging Informatics: Basic Principles and Applications*. John Wiley & Sons.

El Jaouhari, S., Gibaud, B., Lemonnier, P., Pasquier, G., Poiseau, E., Guiffard, E., Hardy, P., et al. (2019). Introduction to DICOM-RTV: a new standard for real-time video communication in hospitals. *2019 IEEE International Conference on E-Health Networking, Application and Services, HealthCom 2019*.

Kadam, S., & Rathod, V. R. (2019). Medical image compression using wavelet-based fractal quad tree combined with huffman coding. *Advances in Intelligent Systems and Computing*.

Kasban, H., & Hashima, S. (2019). Adaptive Radiographic Image Compression Technique using Hierarchical Vector Quantization and Huffman Encoding. *Journal of Ambient Intelligence and Humanized Computing*.

Langer, S. G. (2011). Challenges for data storage in medical imaging research. *Journal of Digital Imaging*.

Lee Rodgers, J., & Alan Nice Wander, W. (1988). Thirteen ways to look at the correlation coefficient. *American Statistician*.

Li, M., Chen, Z., & Sun, Z. (2020). A novel approach for blind recognition of shortened RS codes.

Liu, F., Hernandez-Cabronero, M., Sanchez, V., Marcellin, M. W., & Bilgin, A. (2017). The current role of image compression standards in medical imaging. *Information (Switzerland)*.

Liu, W., & Lin, W. (2013). Additive white gaussian noise level estimation in SVD domain for images. *IEEE Transactions on Image Processing*.

De Macedo, D. D. J., Von Wangenheim, A., & Dantas, M. A. R. (2015). A Data Storage Approach for Large-Scale Distributed Medical Systems. *Proceedings - 2015 9th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2015*.

Mauro, B. (2006). *Document and image compression*. CRC press.

Mrak, M., Grgic, S., & Grgic, M. (2003). Picture quality measures in image compression systems. *The IEEE Region 8 EUROCON 2003. Computer as a Tool*. (Vol. 1, pp. 233–236). IEEE.

N. Baraskar, T., & R. Mankar, V. (2020). The DICOM Image Compression and Patient Data Integration using Run Length and Huffman Encoder. *Coding Theory*.

Ndong, B., Diop, O., Bathily, E. H. A. L., Mbodj, M., Gassama, S. S., Mboup, M. L., Tall, K., et al. (2015). JPEG2000 compression for scintigraphic images of metastasis of the prostatic cancer. *2015 2nd World Symposium on Web Applications and Networking, WSWAN 2015*.

Parikh, S. S., Ruiz, D., Kalva, H., Fernandez-Escribano, G., & Adzic, V. (2018). High Bit-Depth Medical Image Compression with HEVC. *IEEE Journal of Biomedical and Health Informatics*.

Pole, A., & Shriram, R. (2018). 3-D Medical Image Compression by Using HEVC. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–5. IEEE.

Rahmat, R. F., Andreas, T. S. M., Fahmi, F., Pasha, M. F., Alzahrani, M. Y., & Budiarto, R. (2019). Analysis of dicom image compression alternative using huffman coding. *Journal of Healthcare Engineering*.

Ridley, E. (1997). Universities collaborate on de facto standard for lossy compression. *Telemedicine and Telehealth Networks*, 3(6).

Russo, F. (2003). A method for estimation and filtering of Gaussian noise in images. *IEEE Transactions on Instrumentation and Measurement*, 52(4), 1148–1154. IEEE.

Saraiva, A. A., Castro, F. M. J., Costa, N. J. C., Sousa, J. V. M., Ferreira, N. M. F., Valente, A., & Soares, S. (2019). Comparative Study of Compression Techniques Applied in Different Biomedical Signals. *BIOSIGNALS*, 132–138.

Saravanan, C., & Ponalagusamy, R. (2009). Lossless grey-scale image compression using source symbols reduction and Huffman coding. *International Journal of Image Processing (IJIP)*, 3(5), 246.

Schlupkothen, F. R. N. (2012). Interoperability between medical image archives and consumer devices through web services. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*.

Schmainda, K. M., & Prah, M. (2018). Data from brain-tumor-progression. *The Cancer Imaging Archive*.

Shaiboun, M. M., & Shaheen, M. (2016). Streaming medical images using video compression. *2016 IEEE Symposium on Computers and Communication (ISCC)*, 125–128. IEEE.

Singh, A., Khehra, B. S., & Kohli, G. K. (2019). Differential Huffman Coding Approach for Lossless Compression of Medical Images. *International Conference on Intelligent Computing and Communication*, 579–589. Springer.

Tackie Ammah, P. N., & Owusu, E. (2019). Robust medical image compression based on wavelet transform and vector quantization. *Informatics in Medicine Unlocked*.

Toh, K. K. V., & Isa, N. A. M. (2010). Noise adaptive fuzzy switching median filter for salt-

and-pepper noise reduction. *IEEE Signal Processing Letters*.

Venkat, R. A., & Vaidyanathan, C. (2019). Lossless Video Compression Using Bayesian Networks and Entropy Coding. *2019 IEEE Region 10 Symposium (TENSymp)*, 254–259. IEEE.

Wang, C., Yan, Z., Pedrycz, W., Zhou, M., & Li, Z. (2020). A weighted fidelity and regularization-based method for mixed or unknown noise removal from images on graphs. *IEEE Transactions on Image Processing*, 29, 5229–5243. IEEE.

Wong, S., Huang, H. K., Zaremba, L., & Gooden, D. (1995). Radiologic Image Compression—A Review. *Proceedings of the IEEE*.

Zheng, T.-L., Zhu, G.-Y., Wang, Y., Zhao, X.-Y., Li, Y.-Y., & Zhao, L. (2019). A Hybrid Method for Medical Image Compression in Mobile Picture Archiving and Communication System. *Journal of Medical Imaging and Health Informatics*, 9(7), 1401–1406. American Scientific Publishers.

**Percentage of contribution of each author in the manuscript**

Aratã Andrade Saraiva – 34%

Marcos Soares de Oliveira – 33%

João Batista Neto – 33%