

Technical review in digital image/video processing algorithms

Daniel Rodrigues Ferraz Izario
Yuzo Iano
Bruno Rodrigues Ferraz Izario
Diego Arturo Pajuelo Castro

Cite this article:

Izario, Daniel Rodrigues Ferraz, Iano, Yuzo, Izario, Bruno Rodrigues Ferraz, Castro, Diego Arturo Pajuelo Castro; 2018. Technical review on digital image/video processing algorithms. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2018.4. Web Link: <http://dx.doi.org/10.18580/setijbe.2018.4>

Technical review on digital image/video processing algorithms

¹Daniel Rodrigues Ferraz Izario, ²Yuzo Iano, ³Bruno Rodrigues Ferraz Izario and ⁴Diego Arturo Pajuelo Castro

Abstract—This article presents three techniques for applying digital image processing. Unlike conventional techniques, it will be possible to gain in terms of processing time, pixel manipulation, and image quality for database storage. Both techniques have been tested and proven through real images, and can also be applied in video. The algorithms created use JavaScript, Java and Lisp programming languages, combined with mathematical models in the time domain and frequency domain. As an expected result, improvement in digital image processing and performance is sought for future work.

Index Terms—Digital Image/Video Processing; Discrete Cosine Transform; Filters; High-Dynamic Range; Histogram; Java; JavaScript; Lisp; Partial Differential Equation.

I. INTRODUCTION

THE digital image/video processing (PDI/V) is understood as a process of manipulating an image (in video, called a frame) by computer so that the input and output of the process are images [1]; hence, generating an image is something trivial for computers, but high-resolution digital image/frame processing is very complex. There is several types of visual artifacts and distortions; some of them are inherent to the acquisition process, transmission and image visualization, which impacts negatively the image quality [2].

This paper pretend to analyze the quality and time processing from three different digital image/video processing algorithms. For this purpose, two applications (web and local) were developed, using the languages JavaScript [3], Java [4] and Lisp [5]. First, the image/frame is captured by a camera/sensor or chosen from the system itself, then, the stage of processing is performed using as many threads as available. This process initializes with a neighborhood analysis; that is composed by the scan process of the images, and the generation of the pixel sheet saved into a MxN matrix.

In this case, all pixels were filled in the generated worksheet and standardized to avoid discrepancies between current pixel and the others; increasing the quality of images by 2% (value found through matrix-level tests of pixel quality), a relatively small value but really relevant in the digital image/video processing area.

This neighborhood standardization is based on the MSAA (MultiSample Anti-Aliasing) technique, which creates an intelligent process of rendering multiple copies of each pixel and interpolates them. For instance, if there are four identical pixels inside a polygon, they are processed as if were just one,

attributed to one color. Hence, time processing can be reduced. To safe this data, a text document is created that is not influenced by the information of the pixel when is stored.

The last step is to generate a graph of all the pixels and to draw a linear of evaluation for study of that image/frame, that is, a recognition is made if those pixels are linearly plotted on the graph or if they still need to be reworked in the standardization, increasing the rate from 2% to 2.3% of image/frame quality.

Once this is done, the image/frame is prepared to go through the histogram analysis stage, generating a new assessment. This technique is applied into all images for manipulation purposes and is ready to be evaluated by three different algorithms: partial differential equation (PDE), high-dynamic range (HDR) e discrete cosine transform (DCT).

II. METHODOLOGY

A. Partial Differential Equation

The use of partial differential equations [6] in digital image processing improves the restoring process of the image by doing a mathematical modeling. This is an alternative against to conventional filtering techniques. The results obtained improving the quality of the filtered images with minimum loss of resolution [7][8].

The mathematical model is then presented: The image is represented by a function $\vartheta(x_i, y_j)$, where, 'i' and 'j' are the rows and columns of the matrix. To obtain the differences relatives to the partial derivatives, the finite difference approximation method is used to make an approximation in the expressions ϑ_x (1), ϑ_y (2), ϑ_{xx} (3), ϑ_{yy} (4) e ϑ_{xy} (5).

$$\vartheta_x = \frac{\vartheta_{i+j} - \vartheta_{i-j}}{2} \quad (1)$$

$$\vartheta_y = \frac{\vartheta_{ij+1} - \vartheta_{ij-1}}{2} \quad (2)$$

$$\vartheta_{xx} = \frac{\vartheta_{i+2j} - 2\vartheta_{ij} + \vartheta_{i-2j}}{4} \quad (3)$$

$$\vartheta_{yy} = \frac{\vartheta_{ij+2} - 2\vartheta_{ij} + \vartheta_{ij-2}}{4} \quad (4)$$

$$\vartheta_{xy} = \frac{\vartheta_{i+j+1} - \vartheta_{i+j-1} - \vartheta_{i-j+1} + \vartheta_{i-j-1}}{4} \quad (5)$$

The solution of the partial differential equation is obtained using the time scale, $t_n = n\Delta t$, where 'n' represents the course of time in the application. The new function is (6):

All Author are with the Laboratory of Visual Communications from University of Campinas/SP, Brazil. For ¹Daniel Rodrigues Ferraz Izario the e-mail is daniel_izario@hotmail.com; ²Yuzo Iano is

yuzo@decom.fee.unicamp.br; ³Bruno Rodrigues Ferraz Izario is bruno_izario@hotmail.com; and ⁴Diego Arturo Pajuelo Castro is diego.pajuelo.castro@gmail.com.

$$\vartheta(x_i, y_j, t_n) \quad (6)$$

Finally, to obtain the resulting image (I_{result}), it is necessary to apply (7), thus generating in the application an image filtered by the method of partial differential equations.

$$I_{\text{result}} = \delta |\nabla \vartheta| \operatorname{div} \left(\frac{\nabla \vartheta}{|\nabla \vartheta|} \right) + \sigma (1 - \delta) (\vartheta - I_{\text{original}}) \quad (7)$$

B. High-Dynamic Range

HDR is a feature developed to electronic devices in order to generate and reproduce image with better contrast levels and brighter/darker black tones (light source/shadows) [9]. The first step of the algorithm is to obtain the images with their respective information. In (8), each image is represented by x , but all contain in their information the exposure time data (Δt_j) and radiance (E_i), represented in (9).

$$Z_{ij} = f(x) \quad (8)$$

$$Z_{ij} = f(E_i \Delta t_j) \quad (9)$$

After this analysis, a Z_{ij} for each value of intensity of the representative pixel, which is a non-linear function with respect to the exposure values x , represented in (10) and (11).

$$f^{-1}(Z_{ij}) = E_i \Delta t_j \quad (10)$$

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (11)$$

In the equations presented, the known values are Z_{ij} and Δt_j of each image. The values to be obtained are E_i , as well as a new function $g(Z_{ij})$. This new function, represented in (12), has a finite and well-defined range of values (0-255), which is the value of the RGB pixels, since the brightness domain associated with each of these pixels is finite [9].

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (12)$$

In this case, we need to use a mathematical analysis to find the points of $g(Z_{ij})$ and E_i which minimize the quadratic error of the function, represented in (13).

$$e = \sum_{I=1}^N \sum_{J=1}^P [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} [w(z)g''(z)]^2 \quad (13)$$

Where:

P = number of photos available;

N = number of pixels evaluated;

Z_{\max} = maximum pixel intensity values;

Z_{\min} = minimum pixel intensity values;

Using (13), the image is not suitable for the application,

then a triangular weight function is created, smoothing the image, represented in (14).

$$w(z) = \begin{cases} z - Z_{\min} \rightarrow z \leq \frac{1}{2}(Z_{\min} + Z_{\max}) \\ Z_{\max} - z \rightarrow z > \frac{1}{2}(Z_{\min} + Z_{\max}) \end{cases} \quad (14)$$

By inserting the smoothing in equation (13), the new function is obtained, represented by (15).

$$e = \sum_{I=1}^N \sum_{J=1}^P \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} [w(z)g''(z)]^2 \quad (15)$$

Getting the values of $g(Z_{ij})$, are converted to relative radiance values, assuming that Δt_j of each photo is known, represented in (16).

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j \quad (16)$$

To generate the map of radiance/histogram, the information of the images is used, performing a weighted average of these values, represented in (17).

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})} \quad (17)$$

C. Discrete Cosine Transform

Discrete Cosine Transform (DCT) [10] makes use of real numbers and is widely used in digital image processing and data compression. It expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies, where the high frequency components may be discarded with no visual impact.

Before applying the DCT transform to an image, is important to split a frame into blocks of sizes, 4x4, 8x8 or 16x16. In some case, sizes of 32x32 or 64x64 blocks are used in order to optimize the computational complexity and improve the coding efficiency. The block size has an impact on the compression ratio, and the larger the blocks, the lower the compression ratio [10].

This algorithm divides the luma component and the two-chrominance components matrices into several matrices, each with a size of 8x8 pixels. The DCT transform is applied to these matrices, whose values are close to zero. This process transforms the time domain to the frequency domain [10].

The $N \times M$ DCT, is presented in (18):

$$F(\mu, \vartheta) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Delta(i) \times \Delta(j) \times \cos \left[\frac{\pi \times \mu}{2 \times N} (2i + 1) \right] \cos \left[\frac{\pi \times \vartheta}{2 \times M} (2j + 1) \right] \times f(i, j) \quad (18)$$

Where:

$F(\mu, \vartheta)$ = coefficient in the field of processing;

μ = horizontal axis in the transform domain;

ϑ = vertical axis in the transform domain;

$$\Delta(x) = \frac{1}{\sqrt{2}} \text{ for } x = 0;$$

$$\Delta(x) = 1 \text{ for } x \neq 0;$$

$$x = \mu | \vartheta;$$

$f(i, j)$ = amplitude in the time domain;

i = horizontal axis in the time domain;

j = vertical axis in the time domain;

The inverse (IDCT), is calculated as (19):

$$F^{-1}(u, v) \quad (19)$$

Because of the DCT coefficients are real numbers instead of integer numbers, a quantization process is applied to reduce the number of bits required to store a value by reducing the precision of an integer. This process removes the spatial redundancy and is known as lossy compression.

Applying the (19), it is seen that when the pixels of the image were recovered, the values were very close to the originals, proving that the losses are negligible, only the high frequencies were discarded.

III. RESULTS

For this comparative analysis, three NVIDIA GeForce GTX 1070 Ti video cards and two Intel Core i9-8950HK processors were used.

A. Partial Differential Equation

The first test of the Partial Differential Equation was to measure the image/frame processing time in the noise filtering stage. The Signal to Noise Ratio (SNR) is a useful and universal way of comparing the relative amounts of signal and noise to any electronic system in which high ratios will have little visible noise while the opposite is true for reasons. Thus, an intermediate-level SNR was used, that is, random noise, which is characterized by fluctuations in intensity and tone of color in relation to the original image/frame, leaving less clear. As a result, "Algorithm 1" (represented as the standard filtering technique using mask and pixel analysis by neighborhood [6]) processed the image in 32.1 seconds, while in "Algorithm 2" (represented as the search technique) was 25.13 seconds, thus generating a time savings of 6.97 seconds.

The second test generated, is to analyze the profile in terms of the quality of the pixels in the image/frame, for this, the image/frame was matricially compared at the pixel level. Processing time of this technique took 7 hours, 13 minutes and 07 seconds to complete. As a result, "Algorithm 2" achieved on average 92% of quality and precision in the generated pixels, while "Algorithm 1" reached 84%. To calculate the mean value of the PSNR (Peak Signal to Noise Ratio), which is used to define the relation between the maximum energy of a signal and the noise that affects its reliable representation, the MSE (Mean Squared Error) is formulated and then applies the signal-to-noise ratio.

For "Algorithm 1" the value of the PSNR reached was 32.70dB, but in "Algorithm 2" the value was 44.09dB, with that, it generated a gain of 11.39dB. And the last test was at the histogram level, comparing the image/frame, classifying "Algorithm 2" as best at the level of histogram quality, presenting in each RGB layer a larger number of pixels

located in the zone of medium tones. The "Algorithm 1", in turn, had a higher concentration of pixels in the zones of shadows and high lights. All tests are shown in Fig. 1.

B. High-Dynamic Range

The first test of High Dynamic Range was to measure the processing time in the resulting image generation. As a result, "Algorithm 1" (represented as Debevec and Malik [9]) processed the image in 17.8 seconds while "Algorithm 2" (represented as the search technique) in 11.7 seconds, obtaining a gain of 6.1 seconds from the previous one.

The second test generated, is to analyze the profile in terms of pixel quality in the resulting HDR image, for this, a new resulting image was generated matricially and compared at the pixel level. Processing time of this technique took 6 hours, 11 minutes and 25 seconds to complete. As a result, "Algorithm 2" achieved 93% of quality and accuracy in the generated pixels, while "Algorithm 1" reached 81%.

And the last test was at the histogram level, since "Algorithm 2" standardizes the pixels of the resulting HDR image, that is, it performs a matrix and graphical analysis in advance, obtaining the best combinations of the neighborhood pixels, the image is compared, classifying "Algorithm 2" as best at the level of histogram quality, presenting in each RGB layer a larger number of pixels located in the zone of medium tones. The "Algorithm 1", in turn, had a higher concentration of pixels in the zones of shadows and high lights. All tests are shown in Fig. 2.

C. Discrete Cosine Transform

The first test of the Discrete Cosine Transform was to measure the processing time of two images/frames. As a result, "Algorithm 1" (represented as the standard technique of the DCT/IDCT application [10]) processed the image/frame in 32.3 seconds while "Algorithm 2" (represented as the search technique) in 27.7 seconds, with that, generated a saving in time of 4.6 seconds.

The second test generated is to analyze the profile in terms of pixel quality in the image/frame, for this, the two images/frames were matricially compared at the pixel level. Processing time of this technique took 5 hours, 12 minutes and 17 seconds to complete. As a result, "Algorithm 2" achieved 97% of quality and accuracy in the generated pixels, while "Algorithm 1" reached 90%.

And the last test was at the histogram level, comparing the image/frame, classifying "Algorithm 2" as best at the level of histogram quality, presenting in each RGB layer a larger number of pixels located in the zone of medium tones. The "Algorithm 1", in turn, had a higher concentration of pixels in the zones of shadows and high lights. All tests are shown in Fig. 3.

IV. CONCLUSION

This paper assess the impact on quality and processing time of three different state-of-the art algorithms. In summary, the three algorithms developed in the research were better, since the technique used allows the search of the best pixels in a matrix form, allowing the neighborhood pixels to be standardized, besides the use of the histogram to evaluate the average area of their position. All content and algorithms will be used for future work in the digital image/video processing area.



Fig. 1. (A) Original Image; (B) Test 1; (C) Test 2; and (D) Test 3.

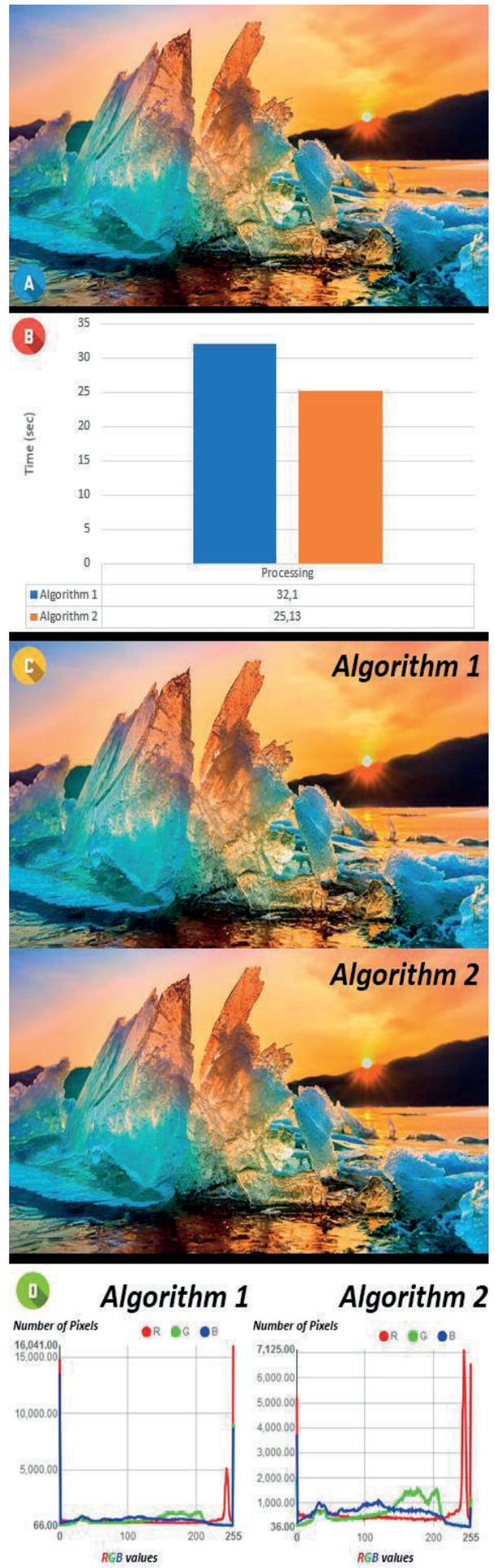


Fig. 2. (A) Original Image; (B) Test 1; (C) Test 2; and (D) Test 3.

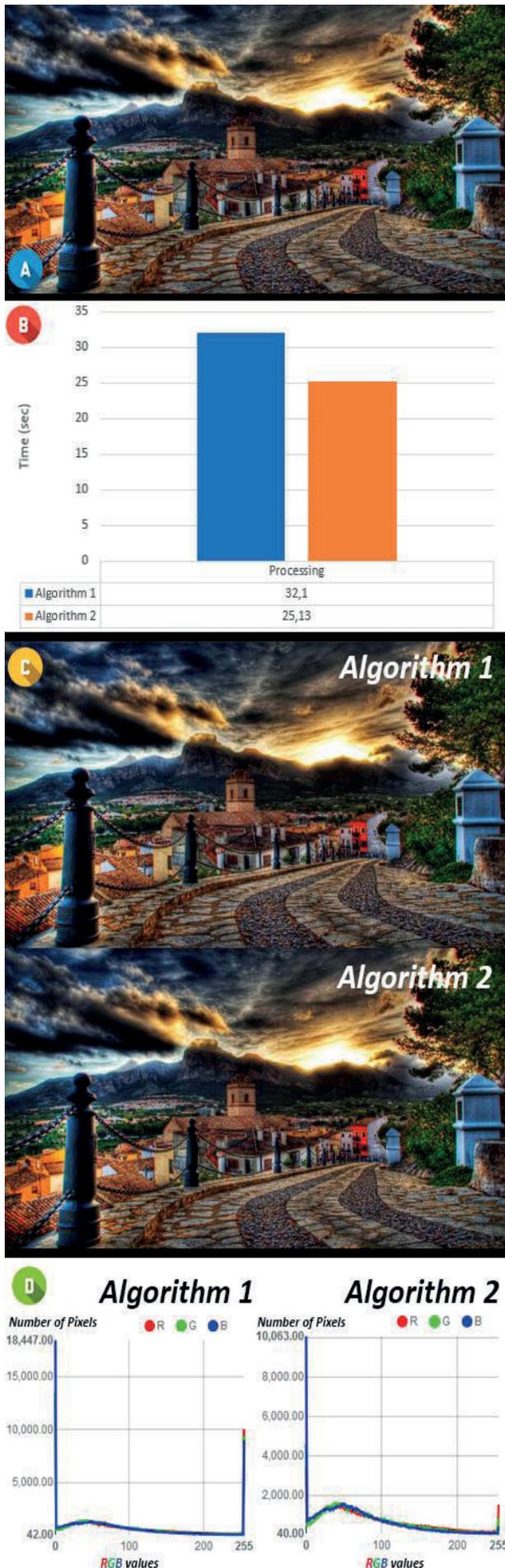


Fig. 3. (A) Original Image; (B) Test 1; (C) Test 2; and (D) Test 3.

REFERENCES

- [1] G. Câmara, R. C. M. Souza et al., "Spring: Integrating remote sensing and gis by object-oriented data modelling," Elsevier - Computers & Graphics, vol. 20, pp. 395–403, Atlanta, Georgia / USA: Elsevier Ltd, May-June 1996.
- [2] R. Gonzalez and P. Woods, Digital Image Processing, 3rd ed. New York City, New York / USA: Pearson, June 2007.
- [3] D. Flanagan, JavaScript: The Definitive Guide, 1st ed. Sebastopol, California / USA: O'Reilly Media, June 2011.
- [4] S. Oaks, Java Performance: The Definitive Guide: Getting the Most Out of Your Code, 1st ed. Sebastopol, California / USA: O'Reilly Media, May 2014.
- [5] P. Graham, On Lisp, 1st ed. Upper Saddle River, New Jersey / USA: Prentice Hall, June 1993.
- [6] T. Acharya and A. K. Ray, Image Processing: Principles and Applications, 1st ed. Hoboken, New Jersey / USA: JohnWiley & Sons, September 2005.
- [7] G. V. Pedrosa; C. A. Z. Barcelos, "Noise eliminations in images using partial differential equations", VIII Encontro Interno / XII Seminário de Iniciação Científica da Universidade Federal de Uberlândia, June 2008.
- [8] C. A. Z. Barcelos; M. Boaventura; E. C. S. Júnior. "A well-balanced flow equation for noise removal and edge detection". IEEE Transaction on Image Processing, May 2003.
- [9] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," Special Interest Group on Computer Graphics and Interactive Techniques, vol. 1, New York City, New York / USA: SIGGRAPH ACM, August 2008.
- [10] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages and Applications, 1st ed. Cambridge, Massachusetts / USA: Academic Press, June 1990.



Izario, D. R. F. Bachelor's at National Institute of Telecommunications/MG, Brazil - Inatel in Computer Engineering (2017), master's degree student at Computer Engineering from State University of Campinas. He is currently a freelancer in the development and planning of websites for stores and businesses and personal websites.



Iano, Y. is the head and founder of the Laboratory of Visual Communications since 1972. He obtained his BSc (1972), MSc (1974) and PhD (1986) in Electrical Engineering at University of Campinas, SP-Brazil. Research interests: digital signal processing (images/audio/video), digital TV, 4G (LTE) and 5G cellular networks.



Izario, B. R. F. Bachelor's at National Institute of Telecommunications/MG, Brazil - Inatel in Electrical Engineering (2011), master's at Electrical Engineering from Mackenzie Presbyterian University (2015) and is working towards his doctoral degree in Sciences and Telecommunications at the State University of Campinas (Unicamp).



Castro, D. A. P. Graduate in Electrical Engineering from the Peruvian University of Applied Sciences (UPC), Lima, Peru in 2012. He is currently working towards his doctoral degree in Sciences and Telecommunications at the State University of Campinas (Unicamp). His research interests are video and audio coding.

Received in 2018-11-30 | Approved in 2018-12-07