doi: 10.5028/jatm.v8i3.540

Quality Evaluation of Chromatic Interpolation Algorithms for Image Acquisition System

Diana Carolina Morón Hernández[1], Freddy Alexander Díaz González[1], Juan Sebastian Triana Correa[1], Pablo Roberto Pinzón Cabrera[1]

ABSTRACT: The main goal of the Libertad 2 mission is to take images of the Earth's surface in the visible spectrum with a multispectral sensor and send them as a Bayer array. To carry out the reconstruction in land of these images and to view them in a RGB color model, it is necessary to use a color interpolation algorithm to determine the values of the different channels of color in each pixel. This document presents an analysis of five chromatic interpolation algorithms (Nearest Neighbor, Adaptive Color Plane, Bilinear, Smooth Hue Transition and Median Based). The analysis also includes the identification of image evaluation algorithms without reference and the artifacts (False Color Measure, Blur Metric and Chromatic and Achromatic Zipper). Finally, the analysis results of the chromatic interpolation algorithms and the selection of the most suitable algorithm for the Libertad 2 satellite mission data are presented.

KEYWORDS: Bayer array, RGB images, Chromatic interpolation quality metrics.

INTRODUCTION

CubeSat satellites have gained recognition as possible platforms for future scientific missions (CubeSat Organization 2014), and previous studies determined the capabilities of these satellites and their functionality in the field of Earth observation. Daniel Selva, from the Massachusetts Institute of Technology (MIT), conducted a study about the feasibility of "Remote Sensing" technologies on CubeSat systems (Woellert *et al.* 2011). In this study, it was concluded that the type of supported technologies can be implemented in satellite systems under the CubeSat standard restrictions. Additionally, Daniel Selva proposes technologies to perform spectral estimations of the state of vegetation and its biomass, particularly to determine the Normalized Differential Vegetation Index (NDVI) (Greenland 2010).

Taking into account studies such as Selva and Krejci (2012) or Sandau (2008), the Universidad Sergio Arboleda is carrying out its second satellite mission called Libertad 2, which consists of the development and implementation of a 3U nanosatellite designed under the CubeSat standard. This mission's main goal is to validate the development of a Remote Sensing System (RSS) prototype, which will open the field for academic researches in Colombia based on the obtained images; these researches may focus on the analysis of land use for the agricultural sector, water resource analysis, planning of urban growth, among others.

The RSS of the Libertad 2 satellite mission will be based on a multispectral Complementary Metal-Oxide Semiconductor (CMOS) sensor, which will deliver the RAW data on a Bayer array. In order to one observe RAW data as a high-quality Red-Green-Blue (RGB) color model image, it is required the implementation

1.Universidad Sergio Arboleda – Escuela de Ciencias Exactas e Ingeniería – Bogotá – Colombia.
Author for correspondence: Freddy Alexander Díaz González | Universidad Sergio Arboleda – Escuela de Ciencias Exactas e Ingeniería | Calle 74 #14-14 | Bogotá – Colombia | Email: freddy.diaz@correo.usa.edu.co
Received: 08/24/2015 | **Accepted:** 04/23/2016

of a chromatic interpolation algorithm that allows to rebuild the original image from the Color Filter Array (CFA) RAW data, calculating each pixel depending on its intensity within a bounded region of the arrangement (Maschal Jr *et al.* 2010). This bounded region of neighboring pixels can vary both the size and the gain of each channel; for this reason, a study should be made to determine the most appropriate algorithm to process the data captured by the Libertad 2 mission.

To perform a chromatic interpolation algorithms analysis, it is required to establish some metrics that allow to evaluate each algorithm's efficiency, since the image reconstruction procedure is usually carried out in the ground station of the satellite mission, where the computing cost is not relevant. The analysis cannot be based on a reference pattern because the input data of interpolation methods are only in Bayer array format and, in any case, in RGB model (Malvar *et al.* 2004). For this reason, the algorithms analysis should focus on the quality of the images without an original image reference to compare pixel by pixel. The interpolated image quality will be measured by the number of defects. These defects are often called "artifacts" and correspond to pixels with inconsistent values in an interpolated image. Some of these artifacts are Blur, Zipper and False Color and decrease the image quality. Thus, only the algorithms with the least amount of defects will be considered for the implementation of the image reconstruction system in the ground station of the mission.

This document describes the evaluation process of five chromatic interpolation algorithms; the first section describes the structure of an image sensor CMOS (which allows the capture of color images) and a RBG color model. Then, the chromatic interpolation algorithms and quality metrics that correspond to the selection criteria of the most appropriate algorithm, identifying the evaluation methods of images without reference, are explained in detail. Finally, the results are presented with some analysis, determining the optimal algorithm for the implementation of the Libertad 2 mission in the Earth observation system.

ACQUISITION OF SATELLITE IMAGES

A satellite image can be defined as the visual representation of the information captured by an installed sensor on-board an artificial satellite. A satellite image is composed of a set of elements with the same size, called "pixels", which are organized in rows and columns. Pixels contain a numeric value or digital number, obtained from the sensors when they capture the amount of energy reflected by the objects in the Earth's surface. When the satellite image is obtained by multispectral sensors, it is contained in a matrix of various dimensions, where each pixel's digital number is located in a row, a column and a band (Pérez *et al.* 2003). An image obtained from a four-band multispectral sensor is shown in Fig. 1.

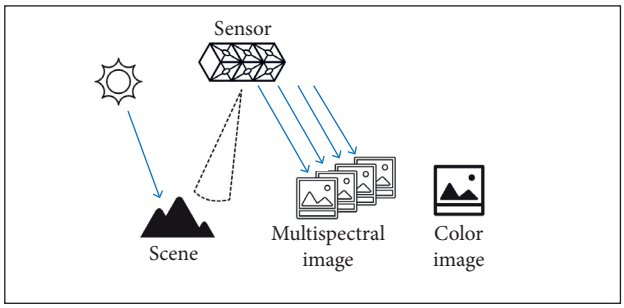

Figure 1. Image obtained from a four-band multispectral sensor.

Multispectral satellite images are those which obtain information in several bands. When the combination of bands is performed only in the visible spectrum, RGB-format images are obtained, often called "color images", because their representation is similar to the images perceived by the human eye. Sensors designed to obtain this type of images capture the energy in the corresponding bands to red, green and blue colors, as well as part of the near-infrared region, which gives more definition to the dynamic element of the captured image.

An image sensor is physically composed of three layers: the first, the outermost, consists of a microlenses array that focuses the incoming light towards each of the photosites; the second layer filters the light in the desired spectrum; the third one consists of an array of light-sensitive photodiodes which determine a value of voltage proportional to the current perceived. After the substrates, CMOS sensors incorporate a Digital Signal Processor (DSP) system with amplifiers and analog-to-digital (A/D) converters to process the image in different delivery formats and control the automatic values of gain for each channel's color and other digital functions available according to the manufacturer's ability (Lillesand *et al.* 2014).

In order to get color images, a filter has to be overlapped in each one of the cavities of an image sensor — CMOS or Charge-Coupled Device (CCD) —, which only allows the passage of required light intensity of primary colors (red, green, and blue) or complementary colors (cyan, magenta, and yellow), depending on how the sensor is built. The data delivered by the sensor form a matrix known as CFA (Li *et al.* 2008). There are different types of CFAs, such as Yamanaka CFA, Diagonal Stripe,

Vertical Stripe, RGBE, CYYM, Bayer (Fig. 2), among others, that vary depending on the manufacturer. CFA data are known as RAW data. To form a color image of the visible spectrum, the intensities of each array of the RGB format are calculated from the RAW data provided by the sensor (Wheeler 2009).

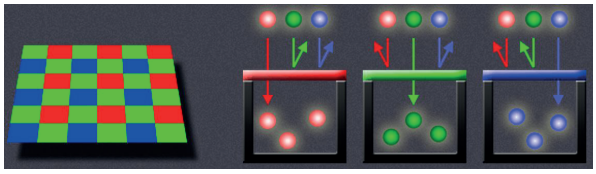

Figure 2. Bayer array (Cambridge in Colour 2013).

Colors can be represented by mathematical methods, known as "color models", for example, RGB or Lab (Hurkman 2010). The RGB color model obtains the different existing colors by combining the three primary ones of the light — red, green, and blue —, which are known as additive colors, because, by adding light intensity, new colors are created (Adobe Creative Team 2007). This color model is located at the RGB space represented in the trichromatic cube, as shown in Fig. 3. Each coordinate in the cube represents a color (Martínez and Díaz 2005).

The data from the RGB model are three stacked arrays which contain the image information in blue, red and green components, as shown in Fig. 4. Through the combination of the information of these components, a color image is obtained.

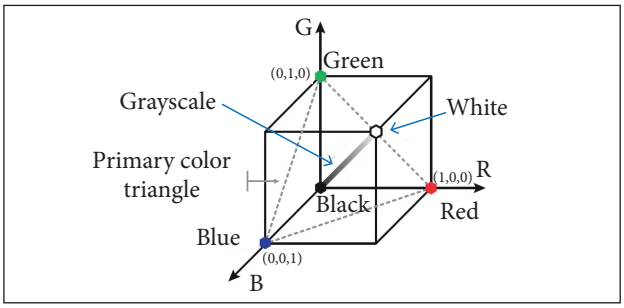

Figure 3. Trichromatic cube (Muñoz 2012).

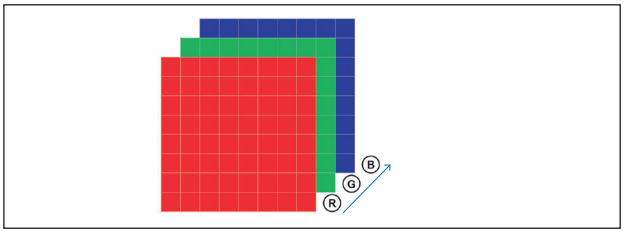

Figure 4. RGB matrix.

COLOR INTERPOLATION ALGORITHMS

Color interpolation is a process used on images to convert a Bayer array into a RGB color model (Losson *et al.* 2009). There are several mathematical methods to calculate RGB values from the data delivered by multispectral sensor CMOS. The most commonly used are: Nearest Neighbor, Bilinear, Adaptive Color Plane, Smooth Hue Transition and Median Based. The main purposes pursued by the color interpolation methods are to:

- Get an RGB image from a Bayer array.
- Rebuild a color image avoiding possible artifacts.
- Lose the fewest possible details in the color image reconstruction.
- Get low computational complexity to accelerate the processing when it is implemented on an embedded system or a mobile device with reduced computational resources.

Most color interpolation algorithms base the calculation of the RGB color model on the intensity of each element, inside a bounded region of the Bayer array, known as "neighboring elements". Generally, the algorithms vary their interpolation method depending on the quantity of elements in the region specified as neighboring elements (Elizondo and Maestre 2005). Color interpolation algorithms are divided into adaptive and non-adaptive algorithms.

Non-adaptive color interpolation algorithms calculate the pixels values of the missing colors taking into account the neighboring pixels. They use few computing resources and are easy to implement. Among these types of algorithms are the Nearest Neighbor, Bilinear and Smooth Hue Transition, which are explained next.

Nearest Neighbor algorithm calculates RGB color model of the image taking the nearest neighbors values and dividing the Bayer array into 2 × 2 subarrays, as in the following example (Ramanath *et al.* 2002).

If the pixel in analysis is the Bayer (1,1), where i is the row and j is the column,

$$\begin{aligned} RGB(i,j,1) &= Bayer(i+1, j) \\ RGB(i,j,2) &= Bayer(i,j) \\ RGB(i,j,3) &= Bayer(i, j+1) \end{aligned} \quad (1)$$

Then RGB(i,j,1) is the component in red of the pixel in analysis, RGB(i,j,2) is the component in green and RGB(i,j,3) is

the component in blue. If the pixel in analysis is the Bayer (1,2),

$$\begin{aligned} RGB(i,j,1) &= Bayer(i+1, j-1) \\ RGB(i,j,2) &= Bayer(i, j-1) \\ RGB(i,j,3) &= Bayer(i, j) \end{aligned} \quad (2)$$

If the pixel in analysis is the Bayer (2,1),

$$\begin{aligned} RGB(i,j,1) &= Bayer(i, j) \\ RGB(i,j,2) &= Bayer(i, j+1) \\ RGB(i,j,3) &= Bayer(i-1, j+1) \end{aligned} \quad (3)$$

If the pixel in analysis is the Bayer (2,2),

$$\begin{aligned} RGB(i,j,1) &= Bayer(i, j-1) \\ RGB(i,j,2) &= Bayer(i, j) \\ RGB(i,j,3) &= Bayer(i-1, j) \end{aligned} \quad (4)$$

Bilinear chromatic interpolation algorithm calculates the value of each pixel in the RGB image computing the average of the pixels of each missing colors in a 3 × 3 submatrix (Ramanath *et al.* 2002).

If the pixel in analysis is the Bayer (2,2),

$$\begin{aligned} RGB(i,j,1) &= \frac{Bayer(i, j-1) + Bayer(i, j+1)}{2} \\ RGB(i,j,2) &= Bayer(i, j) \\ RGB(i,j,3) &= \frac{Bayer(i-1, j) + Bayer(i+1, j)}{2} \end{aligned} \quad (5)$$

If the pixel in analysis is the Bayer (2,3),

$$\begin{aligned} RGB(i,j,1) &= Bayer(i, j) \\ RGB(i,j,2) &= [Bayer(i-1, j) + Bayer(i, j-1) + \\ &\quad + Bayer(i+1, j) + Bayer(i, j+1)]/4 \\ RGB(i,j,3) &= [Bayer(i-1, j-1) + Bayer(i-1, j+1) + \\ &\quad + Bayer(i+1, j-1) + Bayer(i+1, j+1)]/4 \end{aligned} \quad (6)$$

If the pixel in analysis is the Bayer (3,2),

$$\begin{aligned} RGB(i,j,1) &= [Bayer(i-1, j-1) + Bayer(i-1, j+1) + \\ &\quad + Bayer(i+1, j-1) + Bayer(i+1, j+1)]/4 \\ RGB(i,j,2) &= [Bayer(i-1, j) + Bayer(i, j-1) + \\ &\quad + Bayer(i+1, j) + Bayer(i, j+1)]/4 \\ RGB(i,j,3) &= Bayer(i, j) \end{aligned} \quad (7)$$

If the pixel in analysis is the Bayer (3,3),

$$\begin{aligned} RGB(i,j,1) &= \frac{Bayer(i-1, j) + Bayer(i+1, j)}{2} \\ RGB(i,j,2) &= Bayer(i, j) \\ RGB(i,j,3) &= \frac{Bayer(i, j-1) + Bayer(i, j+1)}{2} \end{aligned} \quad (8)$$

Median Based chromatic interpolation algorithm is based on the bilinear interpolation, adding to the model calculation a factor derived from the calculation of the median of the neighbors in a 3 × 3 submatrix (Ramanath *et al.* 2002). This demosaicing algorithm preserves edges well. The **Red (*i, j*)** matrix is the result of applying the bilinear interpolation algorithm to the RAW data in order to obtain the red component. The **Blue (*i, j*)** represents the blue component and **Green (*i, j*)**, the green component.

The values of the missing pixels are calculated as follows.

$$\begin{aligned} RGB(i,j,1) &= [(Red(i,j) - Green(i,j))] + Bayer(i,j) \\ RGB(i,j,2) &= Bayer(i, j) \\ RGB(i,j,3) &= [(Blue(i,j) - Green(i,j))] + Bayer(i,j) \end{aligned} \quad (9)$$

If the pixel in analysis is the Bayer (2,2),

$$\begin{aligned} RGB(i,j,1) &= Bayer(i,j) \\ RGB(i,j,2) &= [(Green(i,j) - Red(i,j))] + Bayer(i,j) \\ RGB(i,j,3) &= [(Blue(i,j) - Red(i,j))] + Bayer(i,j) \end{aligned} \quad (10)$$

If the pixel in analysis is the Bayer (2,3),

$$\begin{aligned} RGB(i,j,1) &= [(Red(i,j) - Blue(i,j))] + Bayer(i,j) \\ RGB(i,j,2) &= [(Green(i,j) - Blue(i,j))] + Bayer(i,j) \\ RGB(i,j,3) &= Bayer(i,j) \end{aligned} \quad (11)$$

If the pixel in analysis is the Bayer (3,2),

Smooth Hue Transition interpolation algorithm is similar to the Bilinear method; however, for the calculation of the red and blue channels, it contemplates the values of luminance channel (green) (Dargahi and Daneshpande 2007). To calculate the pixel value, first, it must determine the green value as the interpolation Green Bilinear (i, j).

If the pixel in analysis is the Bayer (2,2),

$$RGB(i,j,1) = \left(\frac{Green(i,j)}{2}\right) \times \times \left(\frac{Bayer(i,j-1)}{Green(i,j-1)} + \frac{Bayer(i,j+1)}{Green(i,j+1)}\right)$$

$$RGB(i,j,2) = Bayer(i,j) \quad (12)$$

$$RGB(i,j,3) = \left(\frac{Green(i,j)}{2}\right) \times \times \left(\frac{Bayer(i-1,j)}{Green(i-1,j)} + \frac{Bayer(i+1,j)}{Green(i+1,j)}\right)$$

If the pixel in analysis is the Bayer (2,3),

$$RGB(i,j,1) = Bayer(i,j)$$

$$RGB(i,j,2) = \left(\frac{Green(i,j)}{4}\right) \times \left(\frac{Bayer(i-1,j)}{Green(i-1,j)} + \frac{Bayer(i,j-1)}{Green(i,j-1)} + \frac{Bayer(i+1,j)}{Green(i+1,j)} + \frac{Bayer(i,j+1)}{Green(i,j+1)}\right) \quad (13)$$

$$RGB(i,j,3) = \left(\frac{Green(i,j)}{4}\right) \times \left(\frac{Bayer(i-1,j)}{Green(i-1,j)} + \frac{Bayer(i-1,j+1)}{Green(i-1,j+1)} + \frac{Bayer(i+1,j-1)}{Green(i+1,j-1)} + \frac{Bayer(i+1,j+1)}{Green(i+1,j+1)}\right)$$

If the pixel in analysis is the Bayer (3,2),

$$RGB(i,j,1) = \left(\frac{Green(i,j)}{4}\right) \times \left(\frac{Bayer(i-1,j-1)}{Green(i-1,j-1)} + \frac{Bayer(i-1,j+1)}{Green(i-1,j+1)} + \frac{Bayer(i+1,j-1)}{Green(i+1,j-1)} + \frac{Bayer(i+1,j+1)}{Green(i+1,j+1)}\right)$$

$$RGB(i,j,2) = \left(\frac{Green(i,j)}{4}\right) \times \left(\frac{Bayer(i-1,j)}{Green(i-1,j)} + \frac{Bayer(i,j-1)}{Green(i,j-1)} + \frac{Bayer(i+1,j)}{Green(i+1,j)} + \frac{Bayer(i,j+1)}{Green(i,j+1)}\right) \quad (14)$$

$$RGB(i,j,3) = Bayer(i,j)$$

If the pixel in analysis is the Bayer (3,3),

$$RGB(i,j,1) = \left(\frac{Green(i,j)}{2}\right) \times \left(\frac{Bayer(i-1,j)}{Green(i-1,j)} + \frac{Bayer(i+1,j)}{Green(i+1,j)}\right)$$

$$RGB(i,j,2) = Bayer(i,j)$$

$$RGB(i,j,3) = \left(\frac{Green(i,j)}{2}\right) \times \left(\frac{Bayer(i,j-1)}{Green(i,j-1)} + \frac{Bayer(i,j+1)}{Green(i,j+1)}\right) \quad (15)$$

Adaptive chromatic interpolation algorithm allows making decisions and simultaneously calculate the missing colors. Some adaptive algorithms are Adaptive Color Plane and Edge-Direct, which are explained next.

Adaptive Color Plane chromatic interpolation algorithm uses classifications (α and β) to estimate the address to which the pixels belong in order to maintain uniformity of the edges (Ramanath *et al.* 2002). The calculation of the components in the green channel must be performed in the following way, with α as the vertical direction classifier and β as the horizontal one.

$$\alpha = \left|-Bayer(i,j-2) + \left(2 \times Bayer(i,j) - Bayer(i,j+2)\right)\right| + \left|Bayer(i,j-1) - Bayer(i,j+1)\right|$$

$$\beta = \left|-Bayer(i-2,j) + \left(2 \times Bayer(i,j) - Bayer(i+2,j)\right)\right| + \left|Bayer(i-1,j) - Bayer(i+1,j)\right|$$

if $\alpha < \beta$

$$RGB(i,j,2) = \left(\frac{Bayer(i,j-1) + Bayer(i,j+1)}{2}\right) + \left(\frac{-Bayer(i,j-2) + \left(2 \times Bayer(i,j) - Bayer(i,j+2)\right)}{4}\right) \quad (16)$$

if $\alpha = \beta$

$$RGB(i,j,2) = \begin{cases} \{[\, Bayer(i,j-1) + Bayer(i,j+1) + \\ \quad + Bayer(i-1,j) + Bayer(i+1,j)\,]/4\} + \\ + \{[\, -Bayer(i,j-2) - Bayer(i,j+2) + \\ \quad + \left(4Bayer(i,j) - Bayer(i-2,j)\right) - \\ \quad - Bayer(i+2,j))\,]/8\} \end{cases}$$

If the pixel in analysis is the Bayer (2,2),

$$RGB(i,j,1) = \left(\frac{Bayer(i,j-1)+Bayer(i,j+1)}{2}\right) + \\ + \left(\frac{-Bayer(i,j-1)+(2\times Bayer(i,j)-Bayer(i,j+1))}{2}\right)$$

$$RGB(i,j,3) = \left(\frac{Bayer(i-1,j)+Bayer(i+1,j)}{2}\right) + \\ + \left(\frac{-Bayer(i-1,j)+(2\times Bayer(i,j)-Bayer(i+1,j))}{2}\right) \quad (17)$$

If the pixel in analysis is the Bayer (3,2),

$$\alpha = |-Green(i-1,j+1)+(2\times Green(i,j)- \\ -Green(i+1,j-1))|+|Bayer(i-1,j+1)- \\ -Bayer(i+1,j-1)|$$

$$\beta = |-Green(i-1,j-1)+(2\times Green(i,j)- \\ -Green(i+1,j+1))|+|Bayer(i-1,j-1)- \\ -Bayer(i+1,j+1)| \quad (18)$$

if $\alpha < \beta$

$$RGB(i,j,1) = [Bayer(i-1,j+1)+Bayer(i+1,j-1)]/2 + \\ +[-Green(i-1,j-1)+(2\times Green(i,j)- \\ -Green(i+1,j+1))]/2$$

if $\alpha = \beta$

$$RGB(i,j,1) = \begin{cases} [Bayer(i-1,j-1)+Bayer(i-1,j+1)+ \\ +Bayer(i+1,j-1)+Bayer(i+1,j+1)]/4 + \\ +[-Green(i-1,j-1)-Green(i-1,j+1)+ \\ +(4Bayer(i,j)-Green(i+1,j-1)- \\ -Bayer(i+1,j+1))]/4 \end{cases}$$

If the pixel in analysis is the Bayer (2,3),

$$\alpha = |-Green(i-1,j+1)+(2\times Green(i,j)- \\ -Green(i+1,j-1))|+|Bayer(i-1,j+1)- \\ -Bayer(i+1,j-1)|$$

$$\beta = |-Green(i-1,j-1)+(2\times Green(i,j)- \\ -Green(i+1,j+1))|+|Bayer(i-1,j-1)- \\ -Bayer(i+1,j+1)| \quad (19)$$

if $\alpha < \beta$

$$RGB(i,j,3) = \begin{cases} [Bayer(i-1,j+1)+ \\ +Bayer(i+1,j-1) \quad + \\ +[-Green(i-1,j+1)+ \\ +(2\times Green(i,j)- \\ -Green(i+1,j-1))]/2 \end{cases}$$

if $\alpha > \beta$

$$RGB(i,j,3) = \begin{cases} [Bayer(i-1,j-1)+ \\ +Bayer(i+1,j+1)]/2 + \\ +[-Green(i-1,j-1)+ \\ +(2\times Green(i,j)- \\ -Green(i+1,j+1))]/2 \end{cases} \quad (19)$$

if $\alpha = \beta$

$$RGB(i,j,3) = \begin{cases} [Bayer(i-1,j-1)+ \\ +Bayer(i-1,j+1)+ \\ +Bayer(i+1,j-1)+ \\ +Bayer(i+1,j+1) \\ \\ [-Green(i-1,j-1)- \\ -Green(i-1,j+1)+ \\ +(4\times Bayer(i,j)- \\ -Green(i+1,j-1)- \\ -Bayer(i+1,j+1))]/4 \end{cases}$$

In the Edge-Direct chromatic interpolation, the green channel is computed first, in the direction of the pixel analyzed by some classifiers (α, β), where α represents the vertical direction and β, the horizontal one; then it continues with the red and blue channels. An example of the calculation is shown next.

If the pixel in analysis is the Bayer (4,3) or Bayer (3,4),

$$\alpha = \left|\frac{(Bayer(i,j-2)+Bayer(i,j+2))}{2} - Bayer(i,j)\right|$$

$$\beta = \left|\left(\frac{Bayer(i-2,j)+Bayer(i+2,j)}{2} - Bayer(i,j)\right)\right| \quad (20)$$

if $\alpha < \beta$

$$RGB(i,j,2) = \frac{Bayer(i,j-1)+Bayer(i,j+1)}{2}$$

if $\alpha > \beta$

$$RGB(i,j,2) = \frac{Bayer(i-1,j) + Bayer(i+1,j)}{2}$$

if $\alpha = \beta$ (20)

$$RGB(i,j,2) = [Bayer(i-1,j) + Bayer(i+1,j) + Bayer(i,j-1) + Bayer(i,j+1)]/4$$

After calculating the green channel, the red and blue channels are computed as follows.

If the pixel in analysis is the Bayer (2,2),

$$RGB(i,j,1) = \frac{Bayer(i,j-1) + Bayer(i,j+1)}{2} - \left(\frac{(Green(i,j-1) + Green(i,j+1) - (2 \times Green(i,j)))}{2}\right)$$

$$RGB(i,j,3) = \frac{Bayer(i-1,j) + Bayer(i+1,j)}{2} - \left(\frac{(Green(i-1,j) + Green(i+1,j) - (2 \times Green(i,j)))}{2}\right) \quad (21)$$

If the pixel in analysis is the Bayer (2,3),

$$RGB(i,j,1) = Bayer(i,j)$$

(22)

$$RGB(i,j,3) = \begin{cases} Bayer(i-1,j-1) + Bayer(i+1,j+1) + \\ + Bayer(i-1,j+1) + Bayer(i+1,j-1)]/4 \\ [Green(i-1,j-1) + Green(i+1,j+1) + \\ + Green(i-1,j+1)]/4 \\ + \left(\frac{Green(i+1,j-1) - (4 \times Green(i,j))}{4}\right) \end{cases}$$

If the pixel in analysis is the Bayer (3,2),

$$RGB(i,j,1) = \begin{cases} [Bayer(i-1,j-1) + Bayer(i+1,j+1) + \\ + Bayer(i-1,j+1) + Bayer(i+1,j-1) \\ [Green(i-1,j-1) + Green(i+1,j+1) + \\ + Green(i-1,j+1)]/4 + \\ [Green(i+1,j-1) - (4 \times Green(i,j))]/4 \end{cases}$$

(23)

$$RGB(i,j,3) = Bayer(i,j)$$

IMAGE QUALITY EVALUATION

In the evaluation of an algorithm, two kinds of metrics can be set: the first is designed to measure the effectiveness of the algorithm, and the quality of the result is evaluated; the second metric evaluates the efficiency of the algorithm for the computational case the complexity is established based on the amount of mathematical operations required for its implementation (Tuya *et al.* 2007). The research development focuses on the identification of analysis methods of the quality of images without reference.

There are several evaluation methods for the chromatic interpolation algorithms used to measure the quality of the interpolated image (RGB). The most common evaluation methods are those which have a reference image, for example, Color Mean Squared Error (CMSE), Color Peak Signal-to-Noise Ratio (CPSNR), CIELAB ΔE, among others. However, there are methods to measure the quality of the image when there is no reference, for example, Blur Metric, False Color Measure and Achromatic Zipper, which evaluate the quantity of artifacts or defects contained in the calculated RGB image.

The false color artifact usually appears on the edges of the images. It occurs when the result of the interpolation calculates color pixels that do not truly belong to the image (Wenmiao and Tan 2003). Figure 5 shows a false color image.

Figure 5. False color.

The blur artifact is caused by a loss of information of high frequencies in the calculated RGB image (Crete *et al.* 2007; INTECO 2009), making it look blurry and without very well defined edges, as shown in Fig. 6.

Zipper effect occurs commonly in the image edges making them look zipper-shaped. Figure 7 shows an image with zipper effect.

The blur effect affects the edges definition in an image. Since the defined edges in an image are the elements that add

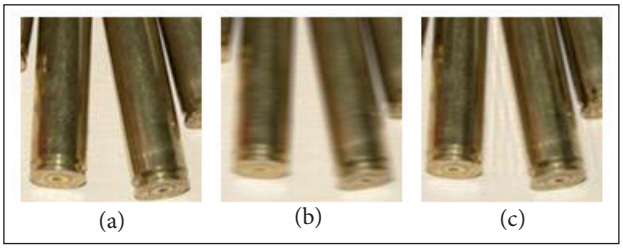

Figure 6. Blur artifact: (a) Original image; (b) Image with high Blur artifact; (c) Image with low Blur artifact.

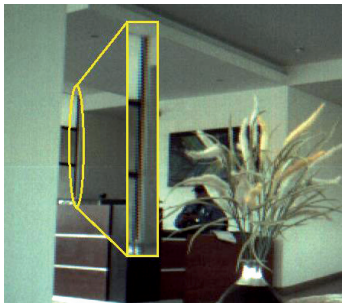

Figure 7. Zipper effect — image with bilinear interpolation.

high frequency components, the blur artifact identification is based on the comparison of the image against itself, but after filtering the high frequencies; the similarity degree of the images is directly proportional to the presence of the blur artifact. The blur index is calculated in the luminance component of the image, that is, in the green channel (Crete *et al.* 2007). The detailed procedure to calculate the blur index is presented in Fig. 8, which shows that the green channel is taken from the RGB image and passes through a Low-Pass Filter — the result is a "Blur G" image. The vertical and horizontal components refer to the rows and columns of the RGB image matrix with a size of $m \times n$. The applied filter is:

$$F = \frac{[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]}{9}$$
$$\mathrm{F'} = \mathrm{Transpose(F)} = (F^t)_{m,n} \tag{24}$$

where F', for the horizontal components, is the transpose of the filtered vertical components of matrix F.

Then the variations between neighboring pixels are calculated for the green channel of the original image and the "Blur G". This is achieved calculating the absolute difference between rows ($D_{F_{ver}}$ and $D_{B_{ver}}$) and columns ($D_{F_{hor}}$ and $D_{B_{hor}}$) for the original and "Blur G" images.

$$D_{F_{ver}(i,j)} = |F(i,j) - F(i-1,j)|\ ;\ for\ i = 1\ to\ m-1, j = 0\ to\ n-1$$
$$D_{B_{ver}(i,j)} = |B(i,j) - B(i-1,j)|\ ;\ for\ i = 1\ to\ m-1, j = 0\ to\ n-1$$
$$D_{F_{hor}(i,j)} = |F(i,j) - F(i,j-1)|\ ;\ for\ j = 1\ to\ n-1, i = 0\ to\ m-1$$
$$D_{B_{hor}(i,j)} = |B(i,j) - B(i,j-1)|\ ;\ for\ j = 1\ to\ n-1, i = 0\ to\ m-1 \tag{25}$$

where: m is the width and n the height of the image.

This calculation considers the two directions, vertical and horizontal, in order to get the VP_RGB and VP_BlurRGB matrixes, as shown in Fig. 8; the magnitude of the variation between these two matrixes is the blur metric (Crete *et al.* 2007).

The False Color Measure evaluates the amount of pixels with a different color from the neighboring pixels; this algorithm is applied on multiple versions of the same image, calculated from the same data of the Bayer array using different chromatic interpolation algorithms.

For the calculation of False Color Measure, first one must find the edges that are common in the different versions of the image, then this particular version to be evaluated is divided into 5 × 5 submatrixes which are moved one position, as shown in Fig. 9. The median of the corresponding submatrix is calculated in the analysis of each pixel. Finally, the matrix with

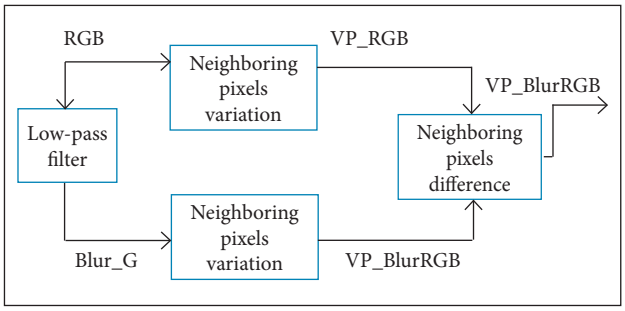

Figure 8. Blur Metric process. (Adapted from Crete *et al.* 2007).

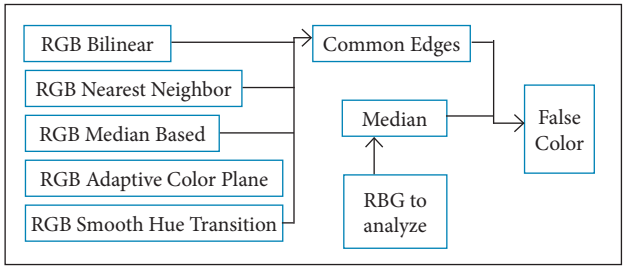

Figure 9. Calculation process for False Color.

the calculated median is analyzed from the information of the identified edges by applying Eq. 24 (Maschal Jr. *et al.* 2010).

$$FC = \frac{\sum_{i,j \in CE} \left[(G_{i,j} - C_{i,j}) - M_{i,j} \right]^2}{nCE} \quad (26)$$

where: *FC* is the false color; *CE* are the common edges; *G* is the green channel; *C* is the analyzed channel (green, red or blue); *M* is the channel median to be analyzed; *nCE* is the number of common edges.

For the calculation of Chromatic and Achromatic Zipper metric, the RGB input image is converted to a grayscale image using Eq. 27:

$$\begin{aligned} Gray(i,j) = {} & (Red(i,j) \times 0.299) + \\ & + (Green(i,j) \times 0.587) + \\ & + (Blue(i,j) \times 0.114) \end{aligned} \quad (27)$$

Then the edges are identified using the following gradient filter:

$$\begin{cases} GV = \begin{bmatrix} -1 & 1 \end{bmatrix} \\ GH = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \end{cases} \quad (28)$$

where: *GV* is the vertical gradient; *GH* is the horizontal gradient.

The horizontal and vertical masks have to be calculated in the following way:

$$SMapx(x,y) = \begin{cases} 2 & Gx(x,y) < 0 \\ 1 & Gx(x,y) > 9 \\ 0 & Other \end{cases} \quad (29)$$

where: *Gx* is the vertical or horizontal gradient.

Thus, the image information that presents artifacts zipper (*zipRGBx*) is removed using Eq. 30. *SMapy* is calculated in the same way.

$$zipRGBx = \left(SMapx \times RGB \right) \quad (30)$$

ZipRGBx and *zipRGBy* are converted to CIELAB color space; finally, the Chromatic Zipper (DC) and Achromatic Zipper (DL) indexes are calculated by applying Eq. 31:

$$\begin{aligned} DL(x,y) &= \left(\Delta L(x,y)^2 \right)^{1/2} \\ DC(x,y) &= \left(\left(\frac{\Delta C(x,y)}{Sc} \right)^2 + \left(\frac{\Delta H(x,y)}{Sh} \right)^2 \right)^{1/2} \end{aligned}$$

where:

$$\begin{aligned} \Delta L(x,y) &= L(x,y) - L(x,y-1) \\ \Delta C(x,y) &= \left(a(x,y)^2 + b(x,y)^2 \right)^{1/2} - \\ &\quad - \left(a(x,y-1)^2 + b(x,y)^2 \right)^{1/2} \\ \Delta H(x,y) &= \left(\Delta E_{76}(x,y)^2 \right) - \\ &\quad - \left(\Delta L(x,y)^2 - \Delta C(x,y)^2 \right)^{1/2} \end{aligned} \quad (31)$$

where: *Sc* and *Sh* are the constants of CIE-94 color difference formula — $Sc = 1 + 0.045\ C^*$; $Sh = 1 + 0.015\ C^*$; C^* is the geometrical mean chroma; *L*, *a* and *b* are CIELAB values; ΔE_{76} is the Euclidean distance between *L*, *a* and *b* values.

CHROMATIC INTERPOLATION ALGORITHMS

For the tests execution, first, the selection of the images database was made, then the tool software was executed using the captured Bayer array and, as a result, the images in the RGB color model were acquired. The RGB images were evaluated by the following methods: Blur Metric, Chromatic and Achromatic Zipper and False Color Measure. Finally, the obtained results were analyzed.

The criteria to select the Bayer array database were:

- The captured images should present similar characteristics in their frequency components.
- Data should be captured with the multispectral imaging sensor OV2640, which is similar to the payload sensor of the Libertad 2 satellite mission.
- The database should contemplate different types of lighting.
- The database should contemplate different color types.

For the tests, 39 images — three groups with 13 images each — were chosen, and the tests scenarios are shown in Table 1. The first group of images was taken with natural lighting; examples of such images are shown in Fig. 10 a,b; the second was captured in a controlled environment with artificial lighting; examples of such images are shown in Fig. 10 c,d; and, finally, the third group was captured in the same controlled environment but with low-frequency images and continuous color regions; examples of such images are shown in Fig. 10 e,f.

Table 1. Scenarios for group of images.

	Group 1	Group 2	Group 3
Edges	Bigger quantity in high frequencies	Bigger quantity in high frequencies	Bigger quantity in low frequencies
Lighting	Natural	Artificial	Artificial
Color	Different shades	Different shades	Continuous color regions

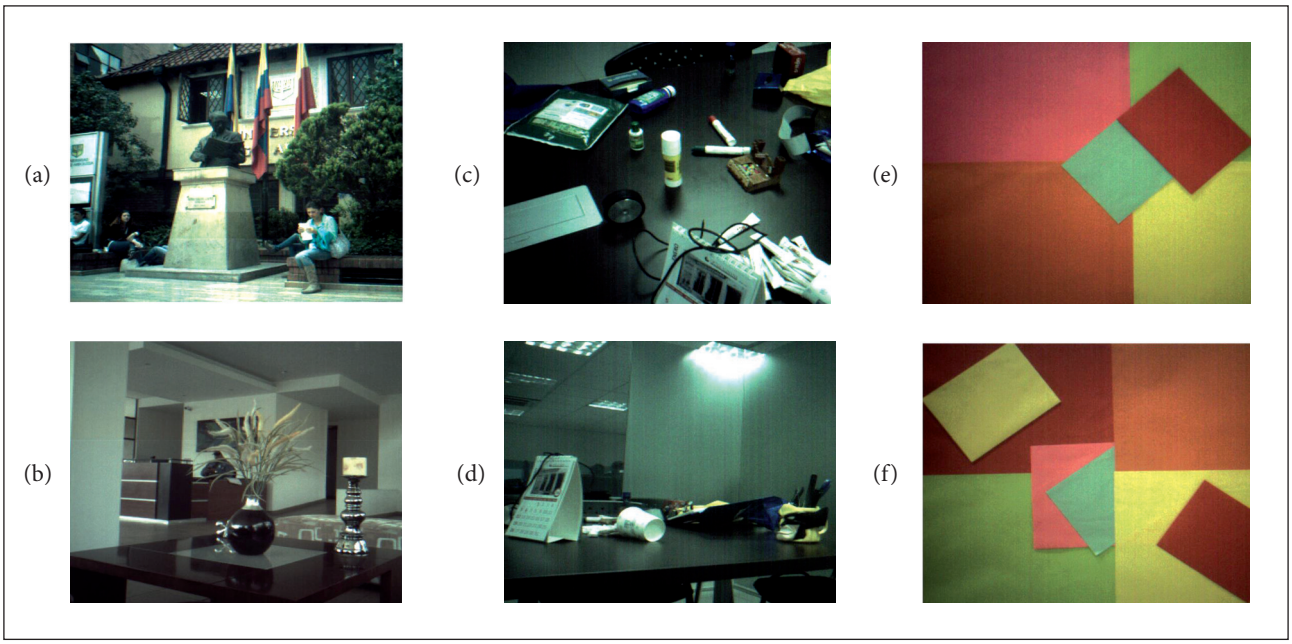

Figure 10. Located image. (a) and (b) at first group; (c) and (d) at second group; (e) and (f) at third group.

The first and second groups present similar characteristics in their frequency components (similar in the amount of edges). These image groups were chosen in order to evaluate the zipper artifacts and false color because they are evaluated at the edges of the image. The third group of images was chosen especially to evaluate the false color artifact.

Then the algorithms of the quality metrics (False Color - Green, False Color - Blue, False Color - Red, Blur Metric, DL and DC) were applied in each one of the images, and the obtained results are presented next.

As shown in Fig. 11 and Table 2, the Nearest Neighbor chromatic interpolation algorithm presents more false color in the red channel, with an average of 30,482; the lowest false color belongs to the Smooth Hue Transition algorithm, with an average of 23,415. The highest blur corresponds to Bilinear, Smooth Hue Transition and Median Based algorithms, with an average of 0.1582; less blur occurs in the Nearest Neighbor algorithm, with an average of 0.1127. In the blue channel, the highest false color is reached by the Nearest Neighbor algorithm, with an average of 12,555, and a lower false color occurs in the Smooth Hue Transition algorithm, with an average of 9,716. The Adaptive Color Plane has the highest DL, with an average of 40,5593. The Bilinear and Median Based algorithms present the lowest DL, with 15,6431. For DC, the highest value is 1.4029 from the Adaptive Color Plane, while the Bilinear and Median Based reach the lowest value, 0.6799. The false color metric for the green channel gets the highest value with the Bilinear, Smooth Hue Transition and Median Based algorithms, with an average of 22,269, while the Nearest Neighbor presents the lowest value, 22,018.

The algorithm that presents higher false color in the red channel is the Nearest Neighbor, with an index of 30,485. This algorithm presents false color 17.32% higher than the Bilinear, 23.18% higher than the Smooth Hue Transition, 18.60% higher than the Adaptive Color Plane and 17.32% higher than the Median Based. The algorithms that present higher blur are Bilinear, Smooth Hue Transition and Median Based, with an index of 0.1582; these algorithms present blur 11.12% higher than the Adaptive Color Plane and 28.76% higher than the Nearest Neighbor. The algorithm with more achromatic zipper artifact in the images was the Adaptive Color Plane, with a

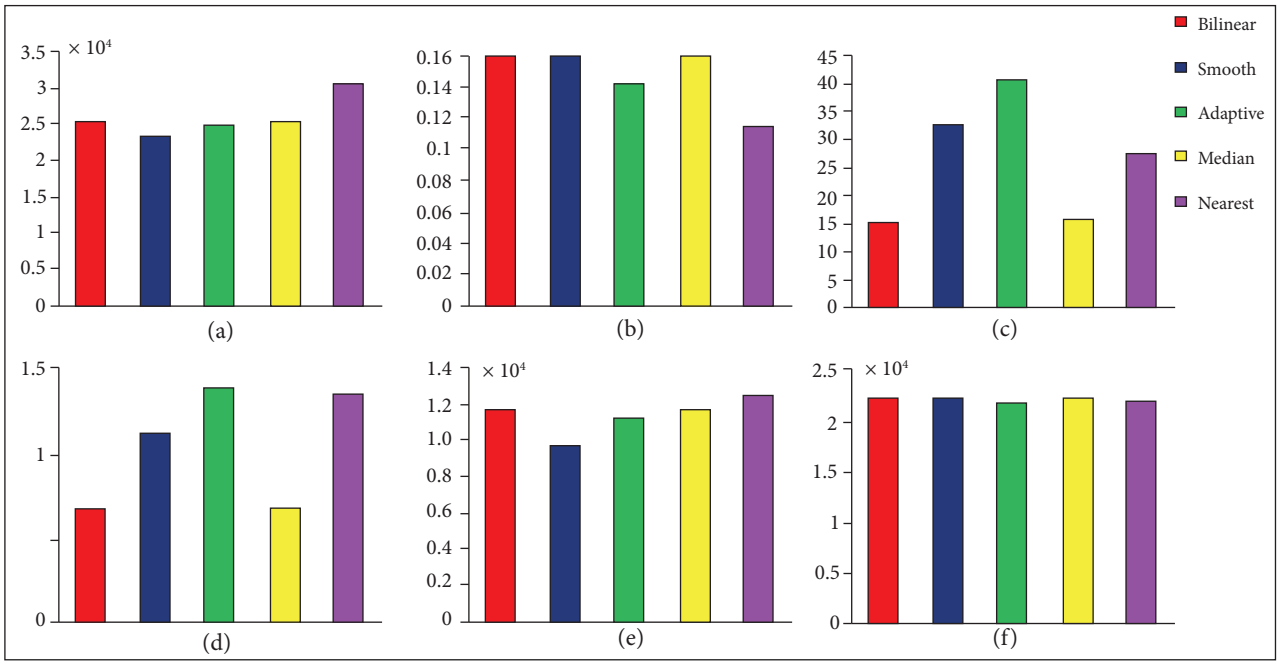

Figure 11. Metric average of 39 images. (a) False Color - Red channel; (b) Blur; (c) Zipper DL; (d) Zipper DC; (e) False Color - Blue channel and (f) False Color - Green channel

Table 2. Metric average of 39 images.

Average	Bilinear	Smooth Hue Transition	Adaptive Color Plane	Median Based	Nearest Neighbor
False Color - Red	25,201	23,415	24,812	25,201	30,482
False Color - Blue	11,735	9,716	11,282	11,735	12,555
False Color - Green	22,269	22,269	21,911	22,269	22,018
Blur Metric	0.1582	0.1582	0.1406	0.1582	0.1127
DL	15.6431	32.6303	40.5593	15.6431	27.3644
DC	0.6799	1.1227	1.4029	0.6799	1.3593

40.5593 index; this algorithm has an achromatic zipper 61.43% higher than the Bilinear and the Median Based, 19.54% higher than the Smooth Hue Transition, 61.43% higher than the Median Based and 32.52% higher than the Nearest Neighbor.

In the test, the algorithm with more chromatic zipper artifact was the Adaptive Color Plane, with a rate of 1.4029; this algorithm has a chromatic zipper 51.53% higher than Bilinear and Median Based, 19.97% higher than the Smooth Hue Transition and 3.10% higher than the Nearest Neighbor. In the test, the algorithm with more false color artifact in the blue channel of the image was the Nearest Neighbor, with an index of 12,555; this algorithm has false color 6.53% higher than Bilinear and Median Based, 22.61% higher than the Smooth Hue Transition and 10.13% higher than the Adaptive Color Plane. Algorithms that have higher false color in the green channel are Bilinear, Smooth Hue Transition and Median Based, with an index of 22,269; these algorithms have false color 1.60% higher than the Adaptive Color Plane and 1.12% higher than the Nearest Neighbor.

RESULT ANALYSIS

The results of the comparison between chromatic interpolation algorithms performed with the 39 images are presented, applying the Blur Metric, False Color Measure, Chromatic and Achromatic Zipper evaluation methods. The sum of the artifacts is presented in Table 2.

Adaptive Color Plane chromatic interpolation algorithm presents the best performance in the quality analysis for the

three evaluation metrics, with a total value of 200,281.3084, as shown in Table 3. The Smooth Hue Transition algorithm has the highest index artifacts values, according to the evaluation metrics, with a total value of 281,347.7491, as shown in Table 3. The Bilinear and Median Based algorithms present equal index artifacts values according to the evaluation metrics, with a total value of 204,970.444, as shown in Table 3. The Smooth Hue Transition chromatic interpolation algorithm presents the highest artifacts values index, 25.21%. The Adaptive Color Plane presents the lowest artifacts values index with respect to the totality of the presented artifacts in the 39 images, 17.94%.

Table 3. Total artifacts in the chromatic interpolation algorithms.

Chromatic interpolation algorithms	Total artifacts	Percentage (%)
Bilinear	204,970.444	18.36
Smooth Hue Transition	281,347.7491	25.21
Adaptive Color Plane	200,281.3084	17.94
Median Based	204,970.444	18.36
Nearest Neighbor	224,325.5096	20.10

CONCLUSIONS

Given the acquisition and download process of the images, the efficiency analysis of the algorithms is irrelevant because the worst case, 8.2593 s of processing, is insignificant for the acquisition and download time of the image. Therefore, the assessment of the algorithms must focus on the quality instead the efficiency, since the artifacts can disturb the researches based on satellite images with badly interpolation procedure.

The chromatic interpolation algorithm to be used must be chosen according to the type of research that will be carried out; for example, research based on region segmentation from satellite images could be affected if the interpolation algorithm does not present low blur artifacts index. On the other hand, in a research based on low-resolution images, it is mandatory to choose an interpolation algorithm with low zipper artifact, due to the low resolution; each pixel means more information, and the artifact can disturb the image interpretation. However, for researches in which the color is essential, as crop recognition, the algorithm with low false color artifact should precede over the algorithm with low zipper artifact.

REFERENCES

Adobe Creative Team (2007) Adobe Illustrator CS3. Berkeley: Peachpit Press.

Cambridge in Colour (2013) Digital camera sensors. A learning community for photographers; [accessed 2014 Nov 23]. http://www.cambridgeincolour.com/tutorials/camera-sensors.htm

Crete F, Dolmiere T, Ladret P, Nicolas M (2007) The blur effect: perception and estimation with a new no-reference perceptual blur metric. Proceedings of SPIE - The International Society for Optical Engineering 12. doi: 10.1117/12.702790

CubeSat Organization (2014) CubeSat; [accessed 2014 Feb 13]. http://www.cubesat.org/

Dargahi N, Daneshpande V (2007) New methods in Bayer demosaicing algorithms, Proceedings Psychology, 221, Applied Vision & Image Systems Engineering, Stanforg University, UK.

Elizondo JJE, Maestre LEP (2005) Fundamentos de procesamiento de imágenes. Mexicali: Universidad Autónoma de Baja California.

Greenland S, Clark C (2010) Cubesat platforms as an on-orbit technology validation and verification vehicle. In Proceedings of the Pestana Conference Centre; Funchal, Portugal.

Hurkman AV (2010) Color correction handbook: professional techniques for video and cinema. Berkeley: Peachpit Press.

Maschal Jr. RA, Young SS, Reynolds J, Krapels K, Fanning J, Corbin T (2010) Review of bayer pattern color filter array (CFA) demosaicing with new quality assessment algorithms (No. ARL-TR-5061). Army Research Lab Adelphi Md Sensors and Electron Devices Directorate.

Instituto Nacional de Tecnologías de la Comunicación (2009) Ingeniería del software: metodologías y ciclos de vida; [accessed 2014 Oct 11]. http://www.inteco.es/calidad_TIC/

Li X, Gunturk B, Zhang L (2008) Image demosaicing: a systematic survey. In: International Society for Optics and Photonics. Electronic Imaging 2008 (p. 68221J-68221J).. Boulder: National Institute of Standards and Technology.

Lillesand T, Kiefer RW, Chipman J (2014) Remote sensing and image interpretation. Hoboken: John Wiley & Sons.

Losson O, Macaire L, Yang Y (2009) Comparison of color demosaicing methods. Adv Imag Electron Phys 162:173-265. doi: 10.1016/S1076-5670(10)62005-8

Malvar HS, He L, Cutler R (2004) High-quality linear interpolation for demosaicing of Bayer-patterned color images. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing; Montreal, Canada.

Martínez J, Díaz A (2005) Percepción remota "Fundamentos de teledetección espacial". Comisión Nacional del Agua Conagua; [accessed 2014 Nov 23]. http://siga.conagua.gob.mx/

Muñoz AV (2012) Principios de color y holopintura. Alicante: Club Universitario.

Pérez C, Aguilera DG, Muñoz AL (2003) Estudio de viabilidad del uso de imágenes comprimidas e procesos de clasificación. Teledetección y Desarrollo Regional. Proceedings of the X Congreso de Teledetección; Cáceres, Spain.

Ramanath R, Snyder WE, Bilbro GL, Sander III WA (2002) Demosaicking methods for Bayer color arrays. J Electron Imaging 11(3):306-315.

Sandau R (2008) Status and trends of small satellite missions for Earth observation. Acta Astronaut 66(1-2):1-12. doi: 10.1016/j.actaastro.2009.06.008.

Selva D, Krejci D (2012) A survey and assessment of the capabilities of Cubesats for Earth observation. Acta Astronaut 74:50-68. doi: 10.1016/j.actaastro.2011.12.014

Tuya J, Román IR, Cosín JD, editors (2007) Técnicas cuantitativas para la gestión en la ingeniería del software. Oleiros: Netbiblo.

Wenmiao L, Tan YP (2003) Color filter array demosaicking: new method and performance measures. IEEE Trans Image Process 12(10):1194-1210. doi: 10.1109/TIP.2003.816004

Wheeler P (2009) High definition cinematography. New York: Taylor & Francis.

Woellert K, Ehrenfreund P, Ricco AJ, Hertzfeld H (2011) Cubesats: Cost-effective science and technology platforms for emerging and developing nations. Adv Space Res 47(4):663-684. doi: 10.1016/j.asr.2010.10.009