RETOOLING OF COLOR IMAGING IN THE QUATERNION ALGEBRA

Artyom M. Grigoryan and Sos S. Agaian

Department of Electrical and Computer Engineering,
University of Texas at San Antonio, USA

ABSTRACT

A novel quaternion color representation tool is proposed to the images and videos efficiently. In this work, we consider a full model for representation and processing color images in the quaternion algebra. Color images are presented in the threefold complex plane where each color component is described by a complex image. Our preliminary experimental results show significant performance improvements of the proposed approach over other well-known color image processing techniques. Moreover, we have shown how a particular image enhancement of the framework leads to excellent color enhancement (better than other algorithms tested). In the framework of the proposed model, many other color processing algorithms, including filtration and restoration, can be expressed.

KEYWORDS

Image Color Analysis, Discrete Fourier Transform, Quaternion Fourier Transformation

1. INTRODUCTION

Color image processing has attracted much interest in the recent years [1],[2] ,[43],[52]. The reason of these are: a) color features are robust to several image processing procedures (for example translation and rotation of the regions of interest) b) color features are efficiently used in many vision tasks, including object recognition, tracking, image segmentation and retrieval, image registration etc.; c) color is of vital importance in many real life applications such as visual communications, multimedia systems, fashion and food industries, computer vision, entertainment, consumer electronics, production printing and proofing, book publishing, digital photography, digital artwork reproduction, industrial inspection, and biomedical applications [1],[2],[5],[43],[44]. Over the years, several important contributions were made in the color image processing systems [2],[52]. Additionally, the traditional color image processing approaches are based on dealing out each color-channel (red, green, and blue) separately [1],[2],[44]. However, this methodology fails to capture the inherent correlation between the components and results in color artefacts [6],[28],[43],[44]. It is natural to ask, how to couple the information contained in the given color- channels, how to process the three color components as a whole unit without loss of the spectral relation that is present in them, or how to develop a mathematical color model that may help to process the color components simultaneously.

Recently, the theory of the quaternion algebra has been used in the application of color science and color systems which process the three color channels simultaneously [6]-[10]. Quaternions were first discovered by Hamilton in 1843" [3]. A quaternion q is an extension of complex numbers and has four components; one is a "real" scalar number, and the other three mutually orthogonal components i,j,k, i.e., q=a+bi+cj+dk, where the coefficients a,b,c, and d are real [6]-[10],[53]. Currently, quaternions have an awe-inspiring amount of influence on various areas of mathematics and physics, including group theory, topology, quantum mechanics, computer graphic, etc [4],[5],[34],[35]. More recently, quaternions have been employed in bioinformatics, navigation systems [5], and image and video processing [6]-[8],[53]. Quaternion algebra for color image was first used by Pei and it led to the description of new tools, such as quaternion Fourier transforms and correlation for image processing by represented the red, green, and blue values at each pixel in the color image as a single pure quaternion valued pixel [6]. In recent years, there have been a number of studies on quaternions in color image processing [12]-[13],[32]-[36],[53]. But all these color processing systems are using pure complex quaternions representation but not the complete quaternions components. Therefore, it is natural to ask, how to use the complete quaternions representation, or more precisely, how to use the "real" scalar number information in the color image processing applications, or what the advantage of the use of the complete representation model over the pure complex quaternions model, particularly in the color image processing applications.

In this paper, we provide a new view of expressing color images using quaternion-based representation. We consider a full model for representation and processing color images in the quaternion algebra. Color images are presented in the threefold complex plane where each color component is described by a complex image. The key contributions of this work are a) an extending model for representing and processing color images by describing each color component as a complex image, b) the practice of the complete quaternions representation models in color image processing application, c) the advantages of the presented approach by using a color image enhancement procedure. The rest of this paper is organized as follows. Section II introduces the background of quaternion algebra and color representation models. Section III presents a new view of expressing color images using quaternion-based representation. Section IV gives the experimental results for color image enhancement by using presented new view of expressing color images using quaternion. Finally, it concludes in Section V that the proposed new quaternion image model is a powerful tool in color image analysis and processing domain which may have many other applications.

2. QUATERNION NUMBERS AND COLOR IMAGES

In recent years, the quaternion algebra has been applied more and more in color image processing. In quaternions the imaginary part of the complex number is extended to three dimensions, i.e., it has three imaginary parts. The imaginary dimensions are represented as i, j, and k, which are orthogonal to each other and to real numbers. Any quaternion is represented in a hyper-complex form as $q = a + (bi + cj + dk)$= $a + bi + cj + dk$, where the coefficients a, b, c, and d are real numbers and i, j, and k are three imaginary units with the following multiplication laws:

$$ij = -ji = k, \;\; jk = -kj = i, \;\; ki = -ik = -j, \quad i^2 = j^2 = k^2 = ijk = -1.$$

The number a is referred to as the "real" part of q and $(bi + cj + dk)$ is the "imaginary" part of q. We also will use the following notation for the quaternion number: $q = q_e + iq_i + jq_j + kq_k$. The quaternion conjugate and modulus of q equal $= a - (bi + cj + dk)$ and $|q| = \sqrt{a^2 + b^2 + c^2 + d^2}$, respectively. The quaternion conjugate is $\bar{q} = q_e - iq_i - jq_j - kq_k$.

The quaternion can be represented in classic polar form as $q = |q|\exp(\mu\vartheta)$, where μ is a unit pure quaternion $\mu = i\mu_i + j\mu_j + k\mu_k$, such that $|\mu| = 1$, and ϑ is a real angle in the interval $[0, \pi]$. The exponential number is defined as $\exp(\mu\vartheta) = \cos(\vartheta) + \mu\sin(\vartheta)$. When multiplying quaternion numbers, it should be noted that commutate property does not hold in quaternion algebra, i.e., $q_1q_2 \neq q_2q_1$. In matrix form, the product of these numbers is

$$\begin{bmatrix}(q_1q_2)_e\\(q_1q_2)_i\\(q_1q_2)_j\\(q_1q_2)_k\end{bmatrix} = \begin{bmatrix}(q_1)_e - (q_1)_i - (q_1)_j - (q_1)_k\\(q_1)_i + (q_1)_e - (q_1)_k + (q_1)_j\\(q_1)_j + (q_1)_k + (q_1)_e - (q_1)_i\\(q_1)_k - (q_1)_j + (q_1)_i + (q_1)_e\end{bmatrix}\begin{bmatrix}(q_2)_e\\(q_2)_i\\(q_2)_j\\(q_2)_k\end{bmatrix}.$$

The quaternion number $q = q_e + iq_i + jq_j + kq_k$ is referred to as a vector $q = (q_e, q_i, q_j, q_k)$ in the 4-D real space R^4 with basic vectors $e = (1,0,0,0)$, i= (0,1,0,0), j= (0,0,1,0), and k= (0,0,0,1). The dot product of two quaternion numbers q_1 and q_2 is defined as

$$q_1 \cdot q_2 = |q_1||q_2|\cos(\vartheta) = (q_1)_e(q_2)_e + (q_1)_i(q_2)_i + (q_1)_j(q_2)_j + (q_1)_k(q_2)_k.$$

2.1. Color Image Models

In this section, we consider a few models of colors that are used in color imaging [44].

RGB Model: Three primary color components, R(ed), G(reen), and B(lue) of a pixel are transferred to three imaginary parts of quaternion numbers with dimensions i, j, and k, respectively. A discrete color image $f_{n,m}$ can therefore be transformed into the imaginary part of quaternion numbers, by considering the red, green, and blue components of the image as pure quaternions (with zero real part):

$$f_{n,m} = 0 + (r_{n,m}i + g_{n,m}j + b_{n,m}k).$$

Figure 1 shows the color map of the colors (r, g, b) into the quaternion space $(1, i, j, k)$.

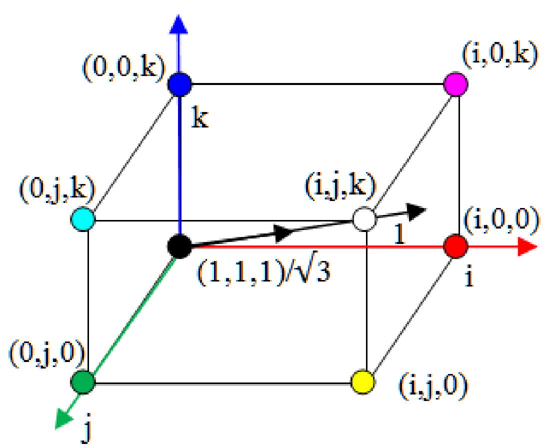


Figure 1. RBG color cube in the quaternion subspace.

The colors in this model are calculated by color components as $C = rR + gG + bB$. Practically, the color is expressed as the triplet (r, g, b), each component of which can vary from zero to a defined maximum value. For example, the triplet $(r, g, b) = (255,0,0)$ is expressed the red color $(1,0,0)$; the triplet $(0,255,0)$ expresses the green color $(0,1,0)$; the triplet $(0,0,255)$ expresses the blue color $(0,0,1)$. If the triplet (r, g, b) is $(0,255,255)$ the result is expressed the magenta color $(M = R + B = (1,0,0) + (0,0,1) = (1,0,1)$, if all components are at zero, the result is black; if all components are at maximum, the result is the brightest representable white. The red and green lights together produce the yellow. Approximately 65% of all cones in the retina are sensitive to the red light, 33% are sensitive to the green light and about 2% are sensitive to the blue light (most sensitive). This RGB color model was described by Thomas Young and Herman Helmholtz in their publication "*Theory of trichromatic color vision*" (first half of the 19th century) and by James Maxwell's (color triangle). RGB is a convenient color model for computer graphics and it is mostly used for recording colors in digital cameras/scanners, including still image and video cameras. There are various types of models based on commonly used RGB color model, for example, RGB ProPhoto RGB, scRGB, and CIE RGB and sRGB.

CMYK color model: The mixed colors in this model are the primary colors of pigment, which are C(yan), M(agenta), and Y(ellow). This model of colors covers a large part of the human color space. The primary colors from RGB color space are transferred to CMYK space by the following simple operations:

$$C = 1 - R, \qquad M = 1 - G, \qquad Y = 1 - B,$$

and the additional forth color, black, as $K = \min(C, M, Y)$ with the following change of colors: $C = C - K$, $M = M - K$, and $Y = Y - K$.

HSI color model: The Hue-Saturation-Intensity color model is a non-linear transformation of the RGB color space. The transformation of colors R, G, and B into the corresponding H, S, and I values in this model is calculated as follows:

$$H = \begin{cases} \vartheta, & \text{if } B \leq G \\ 360 - \vartheta, & \text{if } B > G \end{cases}$$

$$I = \frac{R + G + B}{3}$$

$$S = 1 - \frac{\min\{R, G, B\}}{I}.$$

Here, the angle (in degrees) is calculated by

$$\cos(\vartheta) = \frac{1}{2}\frac{2R - G - B}{\sqrt{(R - G)^2 + (R - B)(G - B)}}.$$

In quaternion space, these three components of the HSI model are defined in the following way [45]. The value (I) component is referred as the norm of the quaternion vector q on the gray axis (axis of real part of q), which is $(q \cdot \mu)\mu$, where for instance $\mu = (1 + j + k)/\sqrt{3}$. The saturation is referred to as the angle between the vectors corresponding to numbers q and μ. The hue is defined by a reference vector ν which is orthogonal to μ, for instance, a vector in red color direction. These three values of the color model can be calculated as

$$H = \operatorname{atan}\left(\frac{|q - \mu\nu q\nu\mu|}{|q - \nu q\nu|}\right),$$

$$V = (q - \mu q\mu)/2, \qquad S = |q + \mu q\mu|/2.$$

CIE XYZ color model: In the XYZ model, a mathematical formula is used to convert the RGB data to a system of positive integers as values X, Y, and Z, which are approximately correspond to red, green, and blue values, respectively. To obtain the XYZ tristimulus values from the primary colors R, G, and B, the following formula is used:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697}\begin{bmatrix} 0.49 & 0.31 & 0.2 \\ 0.17697 & 0.8124 & 0.01063 \\ 0 & 0.01 & 0.99 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

The transformation of values X, Y, and Z into the quaternion space is similar to the RGB color model, i.e., $(X, Y, Z) \rightarrow 0 + (iX\ + jY\ + kZ)$.

Since the color information of the image is transformed in quaternions, the discrete color image in the quaternion algebra is processed as a single matrix. In the traditional approach, the color image is processed separately by each color component. In other words, the processing of the color image is reduced to processing of three gray-scale images independently. It was shown in [36], that the use of quaternions type representation is that a color image is treated as a vector field or the hyper-complex Fourier transforms can handle color image pixels as vectors and thus offer scope to process color images holistically; rather than as separated luminance and chrominance, or separate color space components (example: red, green, blue). The use of the Fourier transform in color imaging is a new and interesting topic in image processing [24]-[27]. As the generalization of the traditional Fourier transform, the quaternion Fourier transform was first defined to process quaternion signals [22]. Later, some practical works related to the quaternion discrete Fourier transforms (QDFT) and their applications in color image processing were presented in [23] and [28].

3. MODIFIED COLOR IMAGE REPRESENTATION AND THE 2-D QDFT

In this section, we consider new methods of representation of color image in the quaternion space and their 2-D QDFTs. Different 2-D quaternion DFTs can be used in image processing, including the right-side and left-side DQFTs [23],[24],[27]. These two transforms are described similarly. Therefore, we consider the right-side 2-D DQFT.

The color image $f_{n,m}$ is considered to be of size $N \times M$. For the color image in the RGB color space $f_{n,m} = (r_{n,m}, g_{n,m}, b_{n,m})$ represented in the quaternion algebra as

$$f_{n,m} = i(r_{n,m}) + j(g_{n,m}) + k(b_{n,m}), \tag{1}$$

the right-side 2-D QDFTs are defined as

$$F_{p,s} = \sum_{n=0}^{N-1} \left(\sum_{m=0}^{M-1} f_{n,m} W_{\mu;M}^{ms} \right) W_{\mu;N}^{np}, \qquad p = 0\colon (N-1), s = 0\colon (M-1), \tag{2}$$

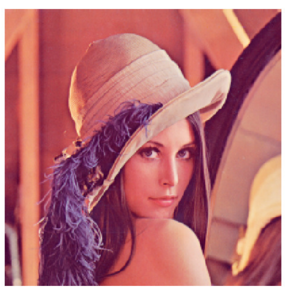
where μ is an unit pure quaternion $\mu = m_1 i + m_2 j + m_3 k$, $\mu^2 = -1$. The kernel of the transform is defined by the periodic exponential functions

$$W_{\mu;N}^{t} = \exp\left(-\mu \frac{2\pi t}{N}\right) = \cos\left(\frac{2\pi t}{N}\right) - \mu \sin\left(\frac{2\pi t}{N}\right), \ \ t = 0\colon (N-1),$$

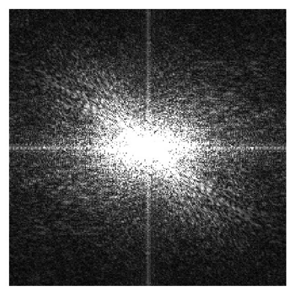
and $W_{\mu;M}^{t}$ defined similarly. The inverse 2-D QDFT is calculated by

$$f_{n,m} = \frac{1}{NM} \sum_{s=0}^{M-1} \left(\sum_{p=0}^{N-1} F_{p,s} W_{\mu;N}^{-np} \right) W_{\mu;M}^{-ms}, \qquad n = 0\colon (N-1), m = 0\colon (M-1). \tag{3}$$

As an example, the color "Lena" image of size 256×256 is shown in Figure 2 in part a.



(a)



(b)

Figure 2. (a) Color image and (b) 2-D QDFT of the quaternion the image.
The 2-D QDFT of the quaternion image $f_{n,m}$ in absolute scale and shifted to the center in part b.

3.1. Model with Gray-Scale Average Image

In this section, we consider a few models which are used in our study for color image enhancement. A quaternion number has four components, and when transforming the color image $f_{n,m}$ from the RGB color space into the quaternion algebra, the color image is presented as $f_{n,m} = (r_{n,m}i + g_{n,m}j + b_{n,m}k)$, i.e., with the real part equal zero. Color imaged can be represented in different color model for different applications. A color model is an abstract mathematical model describing a way the colors can be represented as n-tuple (ordered list of elements) of numbers (e.g. (red, green, blue) in the RGB color model and (hue, saturation, intensity) in HSI model, or four in CMYK (cyan, magenta, yellow and black). Another question arises here how to handle the 4-tuple (CMYK) cases, and what is a best way to plug the primary colors into the quaternion representation. Since the 2-D QDFT is defined not only to process color images in the frequency domain, and quaternion images with non zero real parts, we suggest to fill the real part of the quaternion image by a gray-scale image and use the complete 2-D QDFT. Figure 3 shows the threefold complex plane C^6 or three complex planes intersected between themselves along one real line R^1 in part a. This is a space for all quaternion numbers. These three complex planes C^2 of the threefold complex space are colored in the primary colors, red, green, and blue, since we want to use these planes for the RGB color model. The traditional representation of color images from the RGB color space into the quaternion subspace of numbers with zero real parts is shown in part b. In part c, the mapping of quaternions into a subset of numbers with non zero real parts is given.

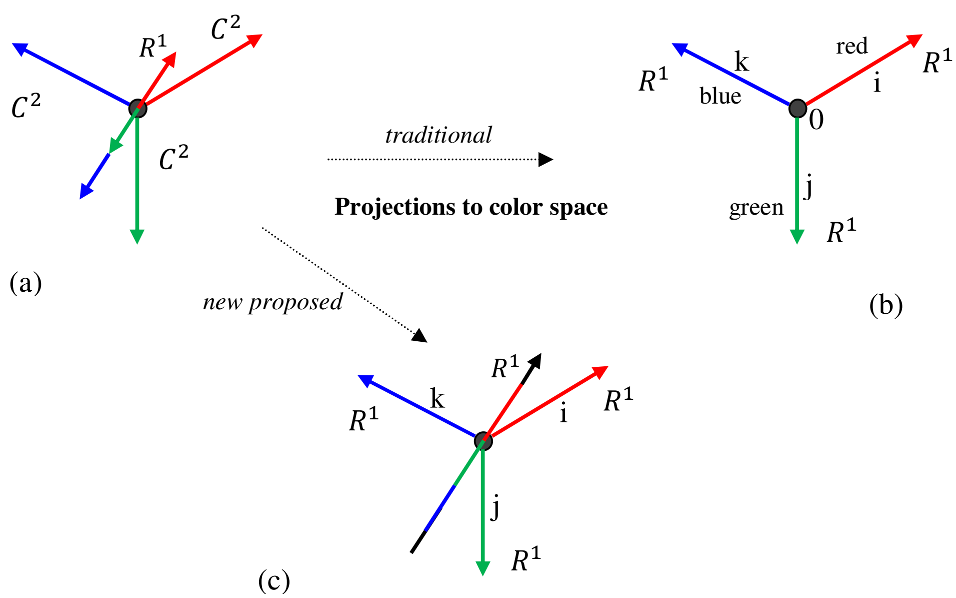


Figure 3. Transformations from the 6-D complex space: (a) The threefold complex plane ($(C^2)^3$ or C^6) of quaternions, (b) the subset (R^3) of quaternions for color images in RGB model, and (c) a new subset (R^4) of quaternions for the model of color images with nonzero gray images.

For model shown in c, the image $a_{n,m} = (r_{n,m} + g_{n,m} + b_{n,m})/3$ can be considered as such gray-scale image. Our preliminary results in image enhancement by the quaternion discrete Fourier transform show, that this real gray-scale component of the quaternion image can be enhanced together with the color image [28]. This enhancement differs from the gray-scale image calculated as the average of processed three color components. Therefore, we define the quaternion-color image by

$$q_{n,m} = a_{n,m} + f_{n,m} = \frac{r_{n,m} + g_{n,m} + b_{n,m}}{3} + \left(r_{n,m}i + g_{n,m}j + b_{n,m}k\right). \qquad (4)$$

This quaternion image can be written as a sum of three complex images

$$q_{n,m} = \left(\frac{r_{n,m}}{3} + r_{n,m}i\right) + \left(\frac{g_{n,m}}{3} + g_{n,m}j\right) + \left(\frac{b_{n,m}}{3} + b_{n,m}k\right) \qquad (5)$$

and

$$q_{n,m} = \left(\frac{1}{3} + i\right) r_{n,m} + \left(\frac{1}{3} + j\right) g_{n,m} + \left(\frac{1}{3} + k\right) b_{n,m}\,. \qquad (6)$$

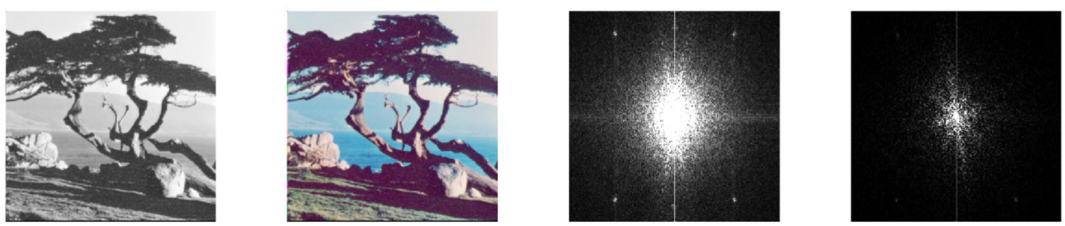
The right-side 2-D QDFT over the quaternion image $q_{n,m}$ is defined as

$$Q_{p,s} = \sum_{n=0}^{N-1} \left(\sum_{m=0}^{M-1} q_{n,m} W_{\mu;M}^{ms} \right) W_{\mu;N}^{np}, \quad p = 0{:}\,(N-1), s = 0{:}\,(M-1). \qquad (7)$$

This also can be written as the modified QDFT (mQDFT)

$$Q_{p,s} = \sum_{n=0}^{N-1} \left(\sum_{m=0}^{M-1} a_{n,m} W_{\mu;M}^{ms} \right) W_{\mu;N}^{np} + F_{p,s}\,. \qquad (8)$$

As an example, Figure 4 shows the gray and color tree images in part a and b, respectively.



(a) (b) (c) (d)

Fig. 4. (a) The gray-scale tree image, (b) color three image, (c) 2-D QDFT of the quaternion tree image, and (d) the difference of 2-D QDFTs of the quaternion and color tree images (in absolute scale).

In this case, the real part $a_{n,m}$ of the quaternion image is the image in a and the imaginary part is the color image $f_{n,m}$ in b. The 2-D QDFT of the quaternion tree image $g_{n,m}$ in absolute scale and shifted to the center is shown in part c, and the difference of 2-D QDFTs of the quaternion and color tree images in d. The processing of the quaternion image will result in not only a new color image and a new gray-scale image as well. (An example of processing different gray-scale and color images in one quaternion image is given in Section IV.)

The number of operations for calculating this 2-D QDFT will increase on the amount required for calculating N complex M-point 1-D QDFTs instead of real M-point 1-D QDFTs. Here, we remind that the complex M-point 1-D QDFT can be accomplished by two complex M-point DFTs, and the real M-point 1-D QDFT can be accomplished by one complex and one real M-point DFTs, for which fast algorithms can be used [13]-[21].The time difference for calculating the 2-D QDFTs $Q_{p,s}$ and $F_{p,s}$ is therefore small, as shown in Table 1 for a few cases when $M = N$ and N is a power of two. The transforms were calculated in MATLAB on a personal computer with Intel(R) Core(TM) i3 CPU Processor at 3.20GHz speed.

Table 1: Time data for calculating the $N \times N$-point real and complex 2-D QDFTs.

N	64	128	256	512	1024	2048
2-D QDFT ($F_{p,s}$)	0.018015s	0.051563s	0.123083s	0.388868s	1.333129s	5.716369s
2-D QDFT ($G_{p,s}$)	0.026308s	0.059262s	0.139223s	0.409061s	1.454468s	6.195034s
time difference	0.0083s	0.0077s	0.0161s	0.0202s	0.1213s	0.4787s

3.2. Model with Gray-Scale Image

Other models of complete quaternion images composed from the color image can also be considered for the 2-D QDFT. For example, the following quaternion image being a sum of three complex images can be taken:

$$q_{n,m} = \left(\frac{1}{3} + i\frac{2}{3}\right) r_{n,m} + \left(\frac{1}{3} + j\frac{2}{3}\right) g_{n,m} + \left(\frac{1}{3} + k\frac{2}{3}\right) b_{n,m}. \tag{9}$$

In this model, all three color components of the image are distributed between the real and imaginary parts in the same way. The color image can be calculated from this quaternion image as

$$f_{n,m} = \frac{3}{2}\left[q_{n,m} - \mathrm{Real}(q_{n,m})\right] = \frac{3}{2}\left[q_{n,m} - \frac{r_{n,m} + g_{n,m} + b_{n,m}}{3}\right]. \tag{10}$$

If we denote three imaginary components of the quaternion image $q_{n,m}$ by $(q_{n,m})_i$, $(q_{n,m})_j$, and $(q_{n,m})_k$, the color image can be defined as

$$r_{n,m} = \frac{3}{2}(q_{n,m})_i, \quad b_{n,m} = \frac{3}{2}(q_{n,m})_j, \quad b_{n,m} = \frac{3}{2}(q_{n,m})_k.$$

We now consider a general model of the color image in the quaternion space. Let a_1, a_2, a_3, and β_1, β_2, and β_3 be some numbers from the interval (0,1). The color image $f_{n,m}$ can be represented as the following quaternion image:

$$q_{n,m} = (a_1 + i\beta_1) r_{n,m} + (a_2 + j\beta_2) g_{n,m} + (a_3 + k\beta_3) b_{n,m} \tag{11}$$

or

$$q_{n,m} = \left(a_1 r_{n,m} + a_2 g_{n,m} + a_3 b_{n,m}\right) + i\beta_1 r_{n,m} + j\beta_2 g_{n,m} + k\beta_3 b_{n,m}.$$

To reconstruct the color image, the following calculations can be used when $\beta_n \neq 0$, $n = 1,2,3$:

$$r_{n,m} = \frac{1}{\beta_1}(q_{n,m})_i, \qquad g_{n,m} = \frac{1}{\beta_2}(q_{n,m})_j, \qquad b_{n,m} = \frac{1}{\beta_3}(q_{n,m})_k,$$

where $n = 0{:}(N-1)$ and $m = 0{:}(M-1)$. Thus, we have a parameterized representation of the color image in the quaternion space, or threefold complex space C^6. For instance, the coefficients a_1, a_2, a_3, and β_1, β_2, and β_3 can be chosen in such a way that $a_n + \beta_n = 1$ for $n = 1,2,3$. When the coefficients $a_1 = a_3 = a_3 = 0$, the quaternion image $q_{n,m}$ is referred to as the tradition representation of the color image. The $a_1 = a_3 = a_3 = 1$ case corresponds to the gray-scale image $q_{n,m} = (r_{n,m} + g_{n,m} + b_{n,m})/3$.

It should be mentioned, that in the quaternion space, we can consider and process simultaneously two different images, gray-scale $v_{n,m}$ and color $f_{n,m}$ images, by combining them into a quaternion image, for instance, as follows:

$$q_{n,m} = (v_{n,m}, f_{n,m}) = v_{n,m} + (ir_{n,m} + jg_{n,m} + kb_{n,m}). \tag{12}$$

Then, after processing this image $q_{n,m} \to \hat{q}_{n,m} = (\hat{v}_{n,m}, \hat{f}_{n,m})$ the output gray-scale and color images are considered to be

$$\hat{v}_{n,m} = \mathrm{Real}\big(\hat{q}_{n,m}\big), \quad \big(i\hat{r}_{n,m} + j\hat{g}_{n,m} + k\hat{b}_{n,m}\big) = \mathrm{Imag}\big(\hat{q}_{n,m}\big),$$

and color components of the new color image $\hat{f}_{n,m}$ are calculated as

$$\hat{r}_{n,m} = (\hat{q}_{n,m})_i, \quad \hat{g}_{n,m} = (\hat{q}_{n,m})_j, \quad \hat{b}_{n,m} = (\hat{q}_{n,m})_k \,.$$

As an example, Figure 5 shows the gray-scale “Lena” image in part a and color tree image in b. These two images compose one quaternion image with four components. In parts c and d, the results of enhancement of the quaternion image are shown. The real component of $q_{n,m}$ is shown in c and the image composed by three color components of the imaginary part in d. “Lena” image and color tree image were enhanced by a single operator in the quaternion space.

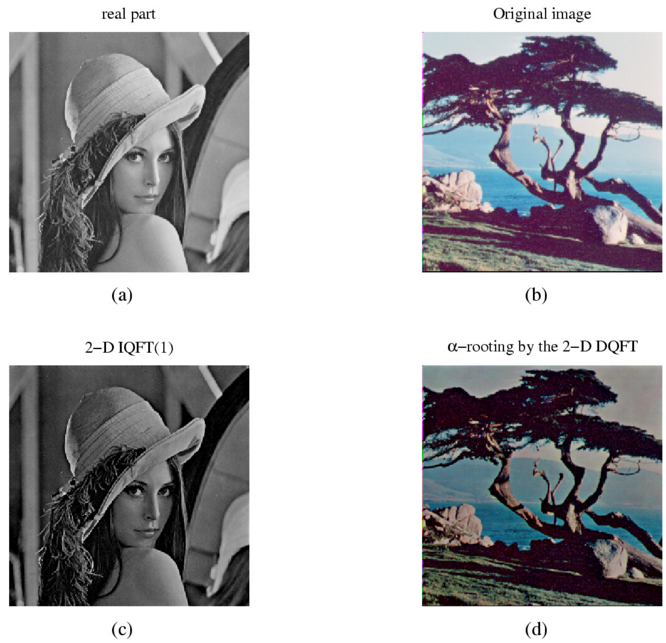


Figure 5. (a) The gray-scale image and (b) color image before and (c) gray-scale image and (d) color image after processing together in the quaternion space.

4. 2-D mQDFT in Image Enhancement

In this section, we consider application of the proposed models of color images in the quaternion space for image enhancement in the frequency domain. The enhancement by the 2-D mQDFT can be described as shown in Figure 6. The 2-D discrete QDFT of the color image is calculated and its amplitude only changes by using an operator M, and then, the inverse 2-D QDFT is calculated,

$$f_{n,m} \to \{F_{p,s} = (|F_{p,s}|, \vartheta_{p,s})\} \to \{\hat{F}_{p,s} = (M[|F_{p,s}|], \vartheta_{p,s})\} \to \{\hat{f}_{n,m}\} \qquad (13)$$

Here, $\vartheta_{p,s}$ is the phase and $(|F_{p,s}|, \vartheta_{p,s})$ is a polar representation of $F_{p,s}$.

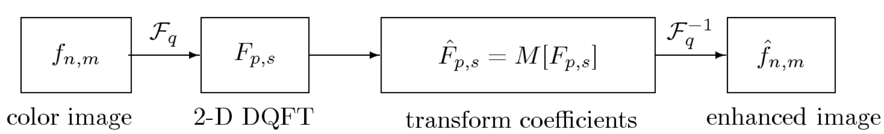


Figure 6. Block-diagram of the image enhancement.

We consider the well-known method of α-rooting for enhancement of images [20],[29]-[31], [39], when the magnitude of the quaternion Fourier transform of the image is transformed as

$$F_{p,s} \to M[|F_{p,s}|] = |F_{p,s}|^{\alpha}$$

for each frequency-point (p, s). The value of α is taken from the interval $(0,1)$ and can be selected by the user, or can be found automatically [19],[20],[28],[37],[41].

To select values of α for image enhancement, we can analyze the color image, by using the measure CEME introduced in [41]. The discrete color image $f_{n,m}$ of size $N \times M$ is divided by $k_1 k_2$ blocks of size $L_1 \times L_2$ each, where integers $k_n = \lfloor N_n/L_n \rfloor$, $n = 1,2$. Here, $\lfloor \cdot \rfloor$ denotes the floor function. The overlapping of these blocks can also be considered [42].The quantitative measure of enhancement of the color image processed by the 2-D QDFT transform,

$$f = (f_R, f_G, f_B) \to \hat{f} = (\hat{f}_e, \hat{f}_R, \hat{f}_G, \hat{f}_B),$$

is defined as follows:

$$E_q(\alpha) = CEME_\alpha(\hat{f}) = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \log_{10} \left[\frac{\max_{k,l}(\hat{f}_R, \hat{f}_G, \hat{f}_B)}{\min_{k,l}(\hat{f}_R, \hat{f}_G, \hat{f}_B)} \right]. \tag{14}$$

Here, $\max_{k,l}(\hat{f})$ and $\min_{k,l}(\hat{f})$ respectively are the maximum and minimum of colors of the image $\hat{f}_{n,m}$ inside the (k, l)th block, and α is a parameter of the enhancement algorithm. $CEME_\alpha(\hat{f})$ is called *a measure of enhancement*, or *measure of improvement of the image* $f_{n,m}$. The "best" image enhancement parameter α is considered to be the one which maximizes the value of the $CEME$, i.e., $CEME_\alpha(\hat{f}) = \max CEME(\hat{f})$. When considering the quaternion image $\hat{f}$ with non zero real part, the enhancement measure CEME is calculated as

$$E_q(\alpha) = CEME_\alpha(\hat{f}) = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \log_{10} \left[\frac{\max_{k,l}(\hat{f}_e, \hat{f}_R, \hat{f}_G, \hat{f}_B)}{\min_{k,l}(\hat{f}_e, \hat{f}_R, \hat{f}_G, \hat{f}_B)} \right]. \tag{15}$$

Now, we consider an example of image enhancement by using the measure $CEME$. Figure 7 shows the color image of size 240×320 in part a. The image has the measure CEME equal 14.7848 when calculated with blocks of size 7×7.

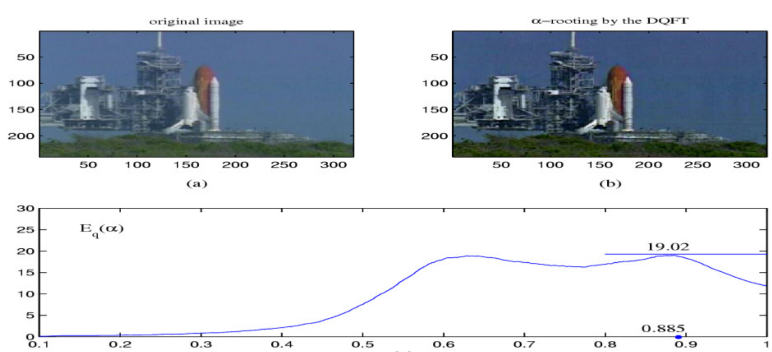


Figure 7. (a) The color image, (b) the 0.8850-rooting by the 2-D DQFT, and (c) the enhancement function of the image.

The curve of the function $E_q(\alpha)$ for this image is given in part c, when α runs the interval $[0.1,1)$. The maximum value of the enhancement is 19.0166 at the point $\alpha = 0.8850$. The corresponding 0.8850-rooting of the color image is shown in part b.

We also consider an example of enhancement of a quaternion image with non zero real part. Figure 8 shows the color tree image in part a, and the enhanced image in b, when the 2-D QDFT-based 0.96-rooting is applied.

Figure 8. (a) Color image and (b) image enhanced by 0.96-rooting.

The enhancement was performed over the image with the real part shown in part a of Figure 9. The CEME measure of the image has a high value at point $\alpha = 0.96$. The real part of the inverse 2-D QDFT after α-rooting is shown in b. For comparison, the gray-scale image calculated from the last three components as their average is shown in c. One can observe that after processing simultaneously the gray-scale and color tree images, the result of processing of the gray-scale image is better, than the average of the color components of the quaternion image.

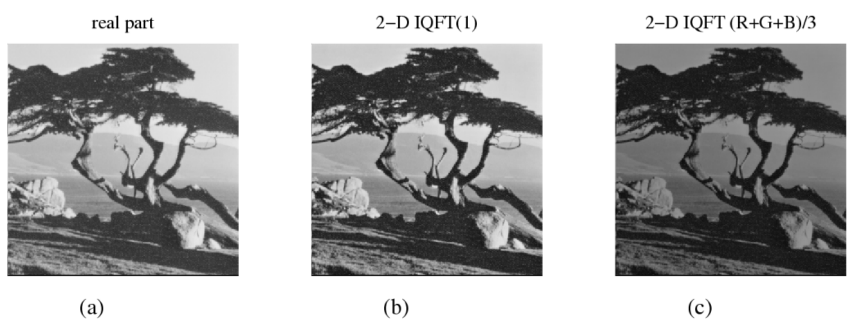


Figure 9. (a) Gray-scale tree image in the quaternion image, (b) enhanced real part of the image, and (c) average of three imaginary components of the enhanced quaternion image.

Recently, gradient based gray level image enhancement has been introduced [46]-[48]. For color image in the quaternion space, we apply the following measure of enhancement calculated on the image gradients:

$$CEME(\hat{f}) = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \log_{10} \left[\frac{\max_{k,l} G_{x+y}(\hat{f})}{\min_{k,l} G_{x+y}(\hat{f})} \right], \qquad (16)$$

where the gradient operators $G_{x+y} = G_x + G_y$ or $G_{x+y} = (G_x, G_y)$. Here, the gradients along the x- and y-axes are calculated over the components of the quaternion image as $\left(G_x[\hat{f}_R], G_x[\hat{f}_G], G_x[\hat{f}_B]\right)$ and $\left(G_y[\hat{f}_R], G_y[\hat{f}_G], G_y[\hat{f}_B]\right)$, respectively.

Different gradient operators are used in digital image processing. We apply the following well-known operations [43],[44]:

Sobel's gradients3×3:

$$G_x = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad G_y = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & -1 \end{bmatrix}.$$

Prewitt's gradients 3×3:

$$G_x = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \qquad G_y = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & -1 \end{bmatrix}.$$

Robert's gradients 3×3:

$$G_x = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad G_y = \frac{1}{4}\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Frei-Chen's gradients 3×3:

$$G_x = \frac{1}{1+\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}, \qquad G_y = \frac{1}{1+\sqrt{2}}\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & -1 \end{bmatrix}.$$

Agaian-Frei-Chen's gradients 5×5:

$$G_x = \begin{bmatrix} 1 & \sqrt{2} & 0 & -\sqrt{2} & -1 \\ \sqrt{2} & 2 & 0 & -2 & -\sqrt{2} \\ 2 & \sqrt{8} & 0 & -\sqrt{8} & -2 \\ \sqrt{2} & 2 & 0 & -2 & -\sqrt{2} \\ 1 & \sqrt{2} & 0 & -\sqrt{2} & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & \sqrt{2} & 2 & \sqrt{2} & 1 \\ \sqrt{2} & 2 & \sqrt{8} & 2 & \sqrt{2} \\ 0 & 0 & 0 & 0 & 0 \\ -\sqrt{2} & -2 & -\sqrt{8} & -2 & -\sqrt{2} \\ -1 & -\sqrt{2} & -2 & -\sqrt{2} & -1 \end{bmatrix}.$$

Many other gradient operators, including the extension of Frei-Chen's gradients of large sizes, can be found in [49]-[52].

5. CONCLUSION

In this paper, a new view of expressing color images using quaternion-based representation was provided. In this work, we consider a full model for representation and processing color images in the quaternion algebra. We have presented a fully quaternion-based color processing framework in which several color analysis problems may be solved. We have shown how a particular image enhancement in the framework of proposed model leads to an excellent color enhancement (better than other algorithms tested). Many other color processing algorithms in the framework of the proposed model can be expressed, including filtration and restoration.

REFERENCES

[1] H.J. Trussell, E. Saber, and M. Vrhel, (2005) "Color image processing," IEEE Signal Processing Magazine, vol. 22, no.1, pp.14– 22.

[2] A. Tremeau, S. Tominaga, and K.N. Plataniotis, (2008) "Color in image and video processing: Most recent trends and future research directions," EURASIP Journal on Image and Video Processing, vol. 2008, 26 p., doi:10.1155/2008/581371.

[3] W.R. Hamilton, (1843) Lectures on quaternions: Containing a systematic statement of a new mathematical method, Dublin: Hodges and Smith.

[4] K. Shoemake, (1985) "Animating rotation with quaternion curves," SIGGRAPH Comput. Graph. vol. 19, no. 3, pp. 245–254.

[5] J.B. Kuipers, (2002) Quaternions and rotation sequences: A primer with applications to orbits, aerospace and virtual reality. Princeton University Press, Princeton.

[6] S.C. Pei and C.M. Cheng, (1996) "A novel block truncation coding of color images by using quaternion-moment-preserving principle," In: IEEE International Symposium on Circuits and Systems, ISCAS 1996, "Connecting the World," vol. 2, pp. 684–687.

[7] S. Sangwine, (1996) "Fourier transforms of colour images using quaternion or hypercomplex numbers," Electronic Letters, vol. 32, no. 21, pp. 1979–1980.

[8] A.R. Leon-Garcia, (1994) Probability and random processes for electrical engineering, Addison-Wesley, Mass.

[9] C.E. Moxey, S.T. Sangwine, and T.A. Ell, (2002) "Hypercomplex operators and vector correlation," in: Proceedings of the 11th European Signal Processing Conference (EUSIPCO). Toulouse, France, vol.3, pp. 247-250.

[10] P. Denis, P. Carre, and C. Fernandez-Maloigne, (2007) "Spatial and spectral quaternionic approaches for colour images," Comput. Vis. Image Und., vol. 107, pp. 74–87.

[11] F.-Y. Lang,etc., (2007) "Quaternion and color image edge detection," Computer Science," vol. 34, no. 11, pp. 212-216.

[12] Ö.N. Subakan, B.C. Vemuri, (2011) "A Quaternion framework for color image smoothing and segmentation," Int. Journal Comput. Vis, vol. 91, pp. 233–250.

[13] A.M. Grigoryan and S.S. Agaian, (2003) Multidimensional Discrete Unitary Transforms: Representation, Partitioning, and Algorithms, New York: Marcel Dekker.

[14] A.M. Grigoryan and S.S. Agaian, (2000) "Efficient algorithm for computing the 2-D discrete Hadamard transform," IEEE Transactions on Circuits and Systems II, vol. 47, no. 10, pp. 1098-1103.

[15] A.M. Grigoryan, and S.S. Agaian, (2000) "Split manageable efficient algorithm for Fourier and Hadamard transforms," IEEE Trans. on Signal Processing, vol. 48, no. 1, pp. 172-183.

[16] S.S. Agaian, (1990) "Advances and problems of the fast orthogonal transforms for signal-images processing applications (Part 1)," in: Pattern Recognition, Classification, Forecasting. Yearbook, The Russian Academy of Sciences, Nauka, Moscow, no. 3, pp. 146-215..

[17] S.S. Agaian, H.G. Sarukhanyan, K.O. Egiazarian, and J. Astola, (2011) Hadamard transforms, SPIE Press.

[18] A.M. Grigoryan, (2010) "Multidimensional Discrete Unitary Transforms," chapter 19, in Transforms and Applications Handbook (3rd edition, A. Poularikas), CRC Press, p. 69.

[19] S.S. Agaian, K. Panetta, and A.M. Grigoryan, (2001) "Transform-based image enhancement algorithms," IEEE Trans. on Image Processing, vol. 10, pp. 367-382.

[20] A.M. Grigoryan and S.S. Agaian, (2004) "Transform-based image enhancement algorithms with performance measure," Advances in Imaging and Electron Physics, Academic Press, vol. 130, pp. 165-242.

[21] R. Narayanam, P. Parimal, and A.M. Grigoryan, (2011) "Performances of Texas Instruments DSP and Xilinx FPGAs for Cooley-Tukey and Grigoryan FFT algorithms," SOURCE Journal of Engineering & Technology, vol. 1, no. 2, p.83.

[22] T.A. Ell, (1993) "Quaternion-Fourier transforms for analysis of 2-dimensional linear time-invariant partial differential systems," in Proc. of the 3nd IEEE Conference on Decision and Control, vol. 1-4, pp. 1830-1841, San Antonio, Texas, USA.

[23] M.H. Yeh, (2008) "Relationships among various 2-D quaternion Fourier transforms," IEEE Signal Processing Letters, vol. 15, pp. 669-–672.

[24] S.J. Sangwine, (1996) "Fourier transforms of colour images using quaternion, or hypercomplex, numbers," Electronics Letters, vol. 32, no. 21, pp. 1979-1980.

[25] S.J. Sangwine, (1997) "The discrete quaternion-Fourier transform," IPA97, Conference Publication, 443, pp. 790-793.

[26] S.J. Sangwine, T.A. Ell, J.M. Blackledge, and M.J. Turner, (2000) "The discrete Fourier transform of a color image," in Proc. Image Processing II Mathematical Methods, Algorithms and Applications, pp. 430-441.

[27] S.J. Sangwine and T.A. Ell, (2001) "Hypercomplex Fourier transforms of color images," in Proc. IEEE Intl. Conf. Image Processing, vol. 1, pp. 137–140.

[28] A.M. Grigoryan and S.S. Agaian, (2014) "Alpha-rooting method of color image enhancement by discrete quaternion Fourier transform," [9019-3], in Proc. SPIE 9019, Image Processing: Algorithms and Systems XII, 901904; 12 p., doi: 10.1117/12.2040596.

[29] J.H. McClellan, (1980) "Artifacts in alpha-rooting of images," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, pp. 449-452.

[30] A. Grigoryan, and S. Agaian, (2003) "Tensor form of image representation: enhancement by image-signals," in Proc. IS&T/SPIE's Symposium on Electronic Imaging Science & Technology, San Jose, CA.

[31] F.T. Arslan and A.M. Grigoryan, (2006) "Fast splitting alpha-rooting method of image enhancement: Tensor representation," IEEE Trans. on Image Processing, vol. 15, no. 11, pp. 3375-3384.

[32] Z. Yang and S.-I. Kamata, (2011) "Hypercomplex polar Fourier analysis for color image," in Proc. International Conference on Image Processing, ICIP 2011.

[33] A. Greenblatt, C. Mosquera-Lopez, S.S. Agaian, (2013) "Quaternion neural networks applied to prostate cancer Gleason grading," in Proc. Man and Cybernetics, SMC 2013. IEEE International Conference, Manchester, pp. 1144-1149, 2013.

[34] L. Guo, M. Dai, and M. Zhu, (2014) "Quaternion moment and its invariants for color object classification," Information Sciences, vol. 273, pp. 132–143.

[35] S.C. Pei, J.J. Ding, and J.H. Chang, (2001) "Efficient implementation of quaternion Fourier transform, convolution and correlation by 2-D complex FFT," IEEE Trans. Signal Processing, vol. 49, pp. 2783-2797.

[36] T.A. Ell and S.J. Sangwine, (2007) "Hypercomplex Fourier transforms of color images," IEEE Trans. Image Processing, vol. 16, no. 1, pp. 22-35.

[37] A.M. Grigoryan and M.M. Grigoryan, (2009) Brief notes in advanced DSP: Fourier analysis with MATLAB, CRC Press Taylor and Francis Group.

[38] S. Sangwine and N.L. Bihan, Quaternion toolbox for MATLAB. Available online: http://qtfm.sourceforge.net/

[39] S.S. Agaian, K. Panetta, and A.M. Grigoryan, (2000) "A new measure of image enhancement," in Proc. IASTED Int. Conf. Signal Processing Communication, Marbella, Spain.

[40] A.M. Grigoryan, (2001) "2-D and 1-D multi-paired transforms: Frequency-time type wavelets," IEEE Trans. on Signal Processing, vol. 49, no. 2, pp. 344-353.

[41] A.M. Grigoryan and S.S. Agaian, (2014) "Alpha-rooting method of color image enhancement by discrete quaternion Fourier transform," [9019-3], SPIE proceedings, 2014 Electronic Imaging: Image Processing: Algorithms and Systems XII, February 2-6, San Francisco, California.

[42] M. Trivedi, A. Jaiswal, and V. Bhateja, (2012) "A novel HSV based image contrast measurement index," in Proc. of the Fourth International Conference on Signal and Image processing 2012 (ICIP 2012), pp. 545-555.

[43] W.K. Pratt, (2000) Digital Image Processing, New York: Wiley.

[44] R.C. Gonzalez and R.E. Woods, (2002) Digital Image Processing, 2nd Edition, Prentice Hall.

[45] P. Denis, P. Carre, and C.F. Maloigne, (2007) "Spatial and spectral quaternionic approaches for color images," Computer Vision and Image Understanding, vol. 107, pp. 74-87.

[46] S. Nercessian, K. Panetta, and S. Agaian, (2009) "A generalized set of kernels for edge and line detection," Proc. SPIE, vol. 7245, 72450U.

[47] E.J. Wharton, K. Panetta, and S.S. Agaian, (2007) “Logarithmic edge detection with applications,” Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, pp. 3346-–3352, Montreal, 7-10.

[48] K. Panetta, S. Qazi, and S. Agaian, (2008) “Techniques for detection and classification of edges in color images,” Proc. SPIE 6982, Mobile Multimedia/Image Processing, Security, and Applications 2008, 69820W; doi:10.1117/12.777703.

[49] S. Nercessian. S.S. Agaian, and K.A. Panetta, (2012) “Multi-scale image enhancement using a second derivative-like measure of contrast,” Proc. SPIE 8295, Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II, 82950Q; doi:10.1117/12.906494.

[50] S. Agaian, (1999) “Visual morphology,” Proc. SPIE, vol. 3646, pp. 139-150.

[51] K. Panetta, C Gao, and S. Agaian, (2013) “No reference color image contrast and quality measures,” IEEE Trans. on Consumer Electronics, vol. 59, no. 3, pp. 643-651.

[52] S. Nercessian, K.A. Panetta, and S.S. Agaian, (2013) “Non-linear direct multi-scale image enhancement based on the luminance and contrast masking characteristics of the human visual system,” IEEE Trans. on Image Processing, vol. 22, no. 9.

[53] Ö.N. Subakan and B.C. Vemuri, (2009) “Color image Segmentation in a quaternion framework, energy minimization methods,” Comput. Vis. Pattern. Recognition, pp. 401–414.

Authors

Artyom M. Grigoryan is Associate Professor in the Electrical and Computer Engineering at the University of Texas, San Antonio. He is the author of three books, three book-chapters, two patents, and many journal papers and specializing in the theory and application of fast Fourier transforms, image enhancement, computerized tomography, processing biomedical images, and image cryptography.

Sos S. Agaian is Professor of Electrical and Computer Engineering at the University of Texas, San Antonio. He has seven books, 500 scientific papers, and holds 14 patents. He is a Fellow of the International Society for Photo-Optical Instrumentation Engineers and Fellow Imaging Sciences and Technology (IS\&T).