I.J. Image, Graphics and Signal Processing, **2012, 2, 9-15**
Published Online March 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2012.02.02

Applying Quaternion Fourier Transforms for Enhancing Color Images

M.I. Khalil
Nuclear Research Center, Atomic Energy Authority, Cairo, Egypt
e-mail: magdi_nrc@hotmail.com

Abstract— **The Fourier transforms play a critical role in a broad range of image processing applications, including enhancement, analysis, restoration, and compression. Until recently, it was common to use the conventional methods to deal with colored images. These methods are based on RGB decomposition of the colored image by separating it into three separate scalar images and computing the Fourier transforms of these images separately. The computing of the Hypercomplex 2D Fourier transform of a color image as a whole unit has only recently been realized. This paper is concerned with frequency domain noise reduction of color images using quaternion Fourier transforms. The approach is based on obtaining quaternion Fourier transform of the color image and applying the Gaussian filter to it in the frequency domain. The filtered image is then obtained by calculating the inverse quaternion Fourier transforms.**

Index Terms— **Image processing, Fourier transforms, Hypercomplex 2D Fourier transform, quaternions, Bandpass Filter.**

I. INTRODUCTION

Fourier analysis is fundamentally a method for expressing a function as a sum of periodic components, and for inversely recovering the signal from those components. When both the function and its Fourier transform are replaced with discretized counterparts, it is called the discrete Fourier transform (DFT). Fourier transforms have been widely used in signal and image processing ever since the discovery of the Fast Fourier Transform (FFT) which made the computation of discrete Fourier transforms feasible using a computer [1]. It allows us to perform tasks which would be impossible to perform any other way; its efficiency allows us to perform other tasks more quickly. The Fourier transform provides, among other things, a powerful alternative to linear spatial filtering; it is more efficient to use the Fourier transform than a spatial filter for a large filter. As general, there are two main approaches to filtering an image. The first is the convolution of an image and kernel in the spatial domain. The second is the multiplication of an image's Fourier transform with a filter in the frequency domain. The fast Fourier transformation (FFT) algorithm, which is an example of the second approach, is used to obtain a frequency-filtered version of an image. Processing images by filtering in the frequency domain is a three-step process:

1. Perform a forward fast Fourier transform to convert a spatial image to its complex Fourier transform image. The complex image shows the magnitude frequency components. It does not display phase information.
2. Enhance selected frequency components of the image and attenuate other frequency components of the image by multiplying by a low-pass, high-pass, band-pass, or band-stop filter.
3. Perform an inverse fast Fourier transform to reconvert the image from the frequency domain to the spatial domain.

According to the convolution theorem, filtering in the frequency domain can be performed by element wise multiplication of the Fourier transform by a suitable “Filter matrix”. The Fourier transform also allows us to isolate and process particular image “frequencies”, and so perform low-pass and high-pass filtering with a great degree of precision.

With the advances in the computing power and memory, color image processing has attracted much interest over the past few years. In this subject area, color image filtering is still an elusive challenge. Due to the multichannel nature of the color images, the key issue is how to couple the information contained in the given color (*e.g.* red, green and blue) channels. Accordingly, conventional methods are used to separate a color image into three scalar images $\boldsymbol{(f_r(x,y), f_g\,(x,y)\ and\ f_b(x,y))}$ and compute the two dimensional fast Fourier transforms of these images separately $\boldsymbol{(F_r(u,v),\ F_g(u,v)\ \ and\, F_b(u,v))}$. Once these images are transformed into the frequency domain, filters can be applied to each image by convolutions. 2D FFT turns the complicated convolution operations into simple multiplications:

$$\boldsymbol{(F_r(u,v)H(u,v),\ \ F_g(u,v)H(u,v)\ \ and\ \ F_b(u,v)H(u,v)).}$$

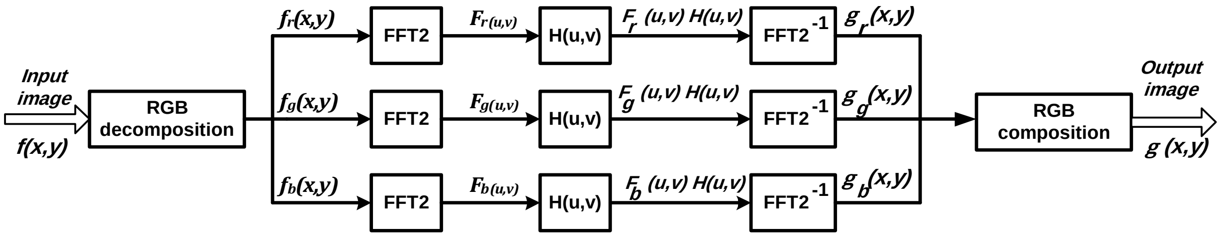


Fig.1 : filtering of color image in the frequency domain based on RGB decomposition

The inverse Fourier transform (***FFT2⁻¹***) can then be computed for each channel yielding three output components ($g_r(x,y), g_g(x,y)$ ***and*** $g_b(x,y)$). These components can be used to compose the filtered image ***g(x,y)*** (Fig.1). However, this approach fails to notice the inherent correlation between the components and results in color artifacts or blending. To avoid this, noise reduction process should be performed in a common and coherent way. The computing of the Hypercomplex 2D Fourier transform of a color image as a whole unit has only recently been realized. The quaternions have been proposed to be used to model color images [2,3], and it has been successfully applied in digital image processing [4-6]. Quaternions provide a new way to process color in combination. The advantage of the quaternion representation for color, as proposed by Sangwine [7,8], is that it combines a color 3-tuple (RGB) into a single hypercomplex number. A color can then be processed as a unit, rather than as three separate channels. Using quaternion operations, higher color information accuracy can be achieved because a color is treated as an entity. In this paper, the quaternion framework will be adopted for filtering color images since it offers scope to process color images holistically, rather than as separate color space components, and thereby handles the coupling between the color channels naturally. In this paper, firstly, a mathematical background concerning quaternions will be reviewed. This leads to the quaternion or complex representation of color image. The discrete quaternion Fourier transforms will be discussed as well. The second part of this paper address the problem of filtering color image in frequency domain using quaternion Fourier transforms. The Gaussian filter will be applied for color image filtering in the frequency domain. At last, evaluation results of typical filtering algorithms in MATLAB environment are summarized and presented.

II. BACKGROUND

A. Quaternions

The quaternions were discovered by Hamilton in 1843 [11]. They combine by the normal rules of algebra with the exception that multiplication is not commutative. A quaternion has four components, one real and three imaginary. The usual notation, extended from that of the complex numbers is:

$$q = w + xi + yj + zk \tag{1}$$

where *w*, *x*, *y* and *z* are real, and ***i, j*** and ***k*** are complex operators which obey the following rules:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2}$$
$$ij = k, \quad jk = i, \quad ki = j \tag{3}$$
$$ji = -k, \quad kj = -i, \; ik = -j \tag{4}$$

The pattern of signs in the products of different operators is easily remembered if the operators i, j *and* k are imagined on a clock face, arranged clockwise in alphabetical order. Multiplication of any pair of operators in a clockwise sequence produces a positive product, while multiplication in anti-clockwise sequence yields a negative product. The quaternion conjugate is:

$$\bar{q} = w - xi - yj - zk \tag{5}$$

And the modulus of a quaternion is given by:

$$|q| = \sqrt{w^2 + x^2 + y^2 + z^2} \tag{6}$$

A quaternion with zero real part is called a *pure* quaternion, and a quaternion with unit modulus is called a *unit* quaternion. The imaginary part of a quaternion has three components and may be associated with a 3-space vector. For this reason, it is sometimes useful to consider the quaternion as composed of a vector part and a scalar part, thus: ***q = S(q) + V(q),*** where the scalar part, ***S(q),*** is the real part, and the vector part is a composite of the three imaginary components, $V(q) = xi + yj + zk$.
The product of two quaternion expressed in terms of their scalar and vector parts is given by:

$$qr = S(q)\,S(r) - V(q).V(r) + S(q)V(r) + S(r)V(q) + V(q)x\,V(r) \tag{7}$$

Where **.** and *x* operators denote the vector dot and cross products respectively. It follows from this that the dot and cross products of two *pure* quaternions, ***u*** and ***v*** are given by:

$$u\,x\,v = \frac{1}{2}\,(uv - vu) \tag{8}$$
$$u\,.\,v = -\frac{1}{2}\,(uv + vu) \tag{9}$$

B. Complex or quaternion representation of color images

It is possible to utilize hypercomplex numbers, in particular, quaternions, and to represent color image pixels as quaternions. Color image pixels have three components, and to model color images in quaternions, the real part is set to zero and the three color components are assigned to the imaginary parts. For example, a pixel at image coordinates ***(x,y)*** in an RGB image can be represented as:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{i} + \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{j} + \boldsymbol{b}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{k} \qquad \textbf{(10)}$$

Where $\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})$ is the red component and $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{b}(\boldsymbol{x},\boldsymbol{y})$ are the green and blue components respectively. The image here is represented as a multivariate image of three vector components. So the resulting representation is a quaternion vectors of dimension the same as the number of pixels in the image.

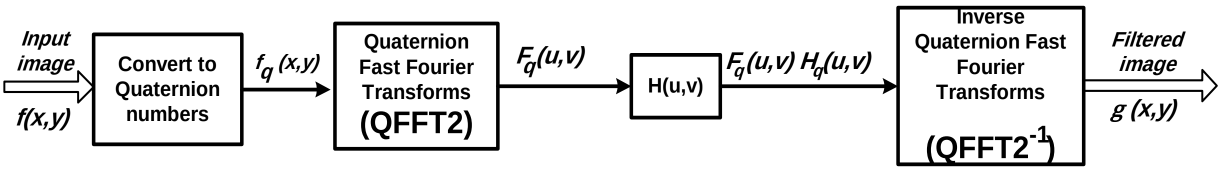

Input image
f(x,y)
Convert to Quaternion numbers
f_q (x,y)
Quaternion Fast Fourier Transforms
(QFFT2)
F_q(u,v)
H(u,v)
F_q(u,v) H_q(u,v)
Inverse Quaternion Fast Fourier Transforms
(QFFT2^-1)
Filtered image
g (x,y)

Figure 2: Filtering of color image in frequency domain based on quaternion Fourier transforms

C. Discrete quaternion Fourier transforms

Based on the concept of quaternion multiplication and exponential, the Quaternion Fourier Transform (QFT) has been introduced. Due to the non-commutative property of the quaternion, there are three different types of QFT defined: the left side QFT, the right side QFT and the two sides QFT. The earliest definition of QFT is the two-side form as following [9].

$$\boldsymbol{F_q}(\boldsymbol{u},\boldsymbol{v}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{e^{-iux}f}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{e^{-jvy}dxdy} \qquad \textbf{(11)}$$

And this equation can be generalized as:

$$\boldsymbol{F_R}(\boldsymbol{u},\boldsymbol{v}) = \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{MN}}} \sum_{\boldsymbol{x=0}}^{\boldsymbol{M-1}}\sum_{\boldsymbol{y=0}}^{\boldsymbol{N-1}} \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{e}^{-\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{xu}}{\boldsymbol{M}}+\frac{\boldsymbol{yv}}{\boldsymbol{N}}\right)} \qquad \textbf{(12)}$$

Two-side DQFT:

$$\boldsymbol{F_{L-R}}(\boldsymbol{u},\boldsymbol{v}) = \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{MN}}} \sum_{\boldsymbol{x=0}}^{\boldsymbol{M-1}}\sum_{\boldsymbol{y=0}}^{\boldsymbol{N-1}} \boldsymbol{e}^{-\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{xu}}{\boldsymbol{M}}\right)}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{e}^{-\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{vy}}{\boldsymbol{N}}\right)} \textbf{(13)}$$

Similarly, the Inverse Discrete Quaternion Fourier Transforms (IDQFT) can be defined for the three types of QFT respectively as Left-side IDQFT:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{MN}}} \sum_{\boldsymbol{x=0}}^{\boldsymbol{M-1}}\sum_{\boldsymbol{y=0}}^{\boldsymbol{N-1}} \boldsymbol{e}^{\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{xu}}{\boldsymbol{M}}+\frac{\boldsymbol{yv}}{\boldsymbol{N}}\right)} \boldsymbol{F_L}(\boldsymbol{u},\boldsymbol{v}) \qquad \textbf{(14)}$$

Right-side IDQFT:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{MN}}} \sum_{\boldsymbol{x=0}}^{\boldsymbol{M-1}}\sum_{\boldsymbol{y=0}}^{\boldsymbol{N-1}} \boldsymbol{F_R}(\boldsymbol{u},\boldsymbol{v})\boldsymbol{e}^{\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{xu}}{\boldsymbol{M}}+\frac{\boldsymbol{yv}}{\boldsymbol{N}}\right)} \qquad \textbf{(15)}$$

Two-side IDQFT:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{MN}}} \sum_{\boldsymbol{x=0}}^{\boldsymbol{M-1}}\sum_{\boldsymbol{y=0}}^{\boldsymbol{N-1}} \boldsymbol{e}^{\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{xu}}{\boldsymbol{M}}\right)} \boldsymbol{F_{L-R}}(\boldsymbol{u},\boldsymbol{v})\boldsymbol{e}^{\boldsymbol{\mu 2\pi}\left(\frac{\boldsymbol{vy}}{\boldsymbol{N}}\right)} \textbf{(16)}$$

III. FILTERING WITH QUATERNION FOURIER TRANSFORMS

The process used to apply filters to an image is known as convolution, and may be applied in either the spatial or frequency domain. Within the frequency domain, convolution can be performed by multiplying the FFT (Fast Fourier Transform) of the image by the FFT of the kernel, and then transforming back into the spatial domain. For a digital image ***f(x,y)*** the basic filtering equation in the frequency domain is:

$$\boldsymbol{g(x, y)} = \Im^{-1}\{\boldsymbol{H(u,v)F(u,v)}\} \qquad \textbf{(17)}$$

Where ***F(u,v)*** and ***H(u,v)*** are DFTs of the image and of the filter respectively. Their product is defined by array (element-by-element) multiplication. The quaternion convolution operation in the spatial domain corresponds to the product operation in the frequency domain [10-14]. This is the same as the case of the conventional convolution. The basic model of filtering in the frequency domain is shown in Fig.2. Each RGB pixel of the input color image ***f(x,y)*** is transferred to a quaternion number, and then the 2D Fast Quaternion Fourier Transform (FQFT2) is applied to the image as whole unit yielding to $\boldsymbol{F_q}(\boldsymbol{u},\boldsymbol{v}$. The selected quaternion filter transfer function $\boldsymbol{H_q}(\boldsymbol{u},\boldsymbol{v})$ is then multiplied (element-by-element) by $\boldsymbol{F_q}(\boldsymbol{u},\boldsymbol{v})$. The filtered image ***g(x,y)*** can then be obtained simply by computing the inverse quaternion Fourier transform of the product $\boldsymbol{F_q}(\boldsymbol{u},\boldsymbol{v})\boldsymbol{H_q}(\boldsymbol{u},\boldsymbol{v})$**.**

To illustrate the practical application of quaternion Fourier transforms in the field of image processing, two filtering techniques will be investigated here as examples of noise reduction of color images. The first one, the Gaussian band-pass filter, deals with the problem of reducing the noise in color images. The Gaussian filter will be used as an example of noise reduction techniques. The second technique will address the problem of edge detection in color images.

A. *Noise reduction using gaussian filter and quaternion fourier transforms*

The transfer function of the Gaussian low pass filter in two-dimensions is given by:

$$\boldsymbol{H(u,v) = e^{-D^2(u,v)/2\sigma^2}} \qquad \textbf{(18)}$$

Where $\boldsymbol{\sigma}$ measure of spread about the center. Let $\boldsymbol{\sigma = D_0}$, then :

$$\boldsymbol{H(u,v) = e^{-D^2(u,v)/2{D_0}^2}} \qquad \textbf{(19)}$$

where $\boldsymbol{D_0}$ is the cutoff frequency.

Since Gaussian filters have the very nice mathematical property that a Fourier transform of a Gaussian is a Gaussian, we should get exactly the same results as when using a linear Gaussian spatial filter. And therefore, it does not incur the ringing effect in the spatial domain of the filtered image. The derivation of a Gaussian band-pass filter starts from the low-pass filter:

$$\boldsymbol{H_{LP}(u,v) = e^{-D^2(u,v)/2D_L^2}} \qquad \textbf{(20)}$$

And the high-pass filter:

$$\boldsymbol{H_{HP}(u,v) = 1 - e^{-D^2(u,v)/2D_H^2}} \qquad \textbf{(21)}$$

And, the band-pass filter:

$$\boldsymbol{H_{BP}(u,v) = H_{LP}(u,v) * H_{HP}(u,v), \quad D_L > D_H} \qquad \textbf{(22)}$$

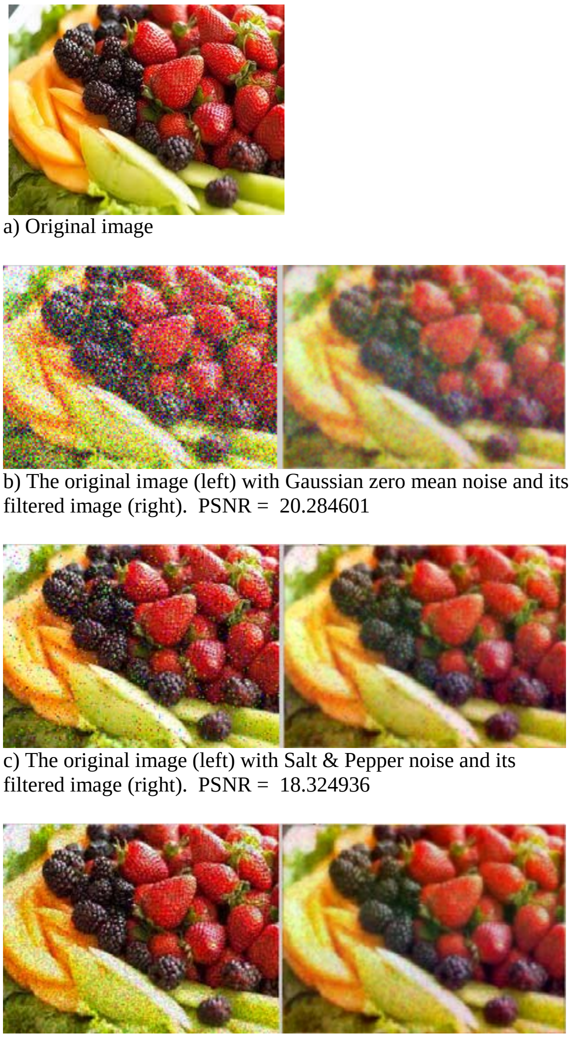
Where $\boldsymbol{D_L}$ and $\boldsymbol{D_H}$ are the cutoff frequencies of the low and high pass filters respectively.

Images are often degraded by noises. Noise can occur during image capture, transmission, etc. Noise removal is an important task in image processing. Image Noise is classified as Amplifier noise (Gaussian noise), Salt-and-pepper noise (Impulse noise), Shot noise, Quantization noise (uniform noise), Film grain, on-isotropic noise, Speckle noise (Multiplicative noise) and Periodic noise. Accordingly, the images presented in Fig.3 demonstrate the effect of applying the Gaussian filter to these images using the previously described quaternion Fourier transform method. The image in Fig.3a is used as test image and is exposed to different types of noise as following:

- In Fig.3b-1, ***Gaussian zero mean noise*** with 0.01 variance is added to the original image. The produced image is transferred and represented as quaternions numbers which in turn is transferred to the frequency domain using the fast quaternion Fourier transforms. The product is then multiplied element-by-element with the Gaussian filter. The filtered image is then obtained simply by computing the inverse quaternion Fourier transform of the obtained product yielding to the image represented in Fig.3b-2.

- In Fig.3c-1, ***Salt & pepper noise*** with 0.05 noise density is added to the original image. The previous procedure is applied to this image yielding to the image represented in Fig.3c-2.

- In Fig.3d-1, ***Speckle uniformly distributed noise*** with mean of 0 and variance 0.04 is added to the original image. The previous procedure is applied to this image yielding to the image represented in Fig.3d-2.



a) Original image

b) The original image (left) with Gaussian zero mean noise and its filtered image (right). PSNR = 20.284601

c) The original image (left) with Salt & Pepper noise and its filtered image (right). PSNR = 18.324936

d) The original image (left) with Speckle uniformly distributed noise and its filtered image (right). PSNR = 20.284069

Fig.3: The results of applying the Gaussian filter to color image based on the quaternion Fourier transforms.

B. Image qualty estimation

To estimate the quality of the filtered image, the Peak Signal to Noise Ratio (PSNR) is computed. PSNR is the value of the filtered image with respect to that of the noisy image. The value of PSNR and RMSE (Root Mean Square Error) for the proposed method is found out experimentally. The PSNR and the Root Mean Square Error of the retrieved image can be calculated by using the following equations:

$$\boldsymbol{PSNR = 20\, log_{10}(\frac{255}{RMSE})} \qquad \textbf{(23)}$$

where

$$\boldsymbol{RMSE = \sqrt{\frac{\sum_{i=1}^{r\,c} |x_i - \hat{x}_i\,|}{3\,r\,c}}}$$

$\boldsymbol{x_i}$ and $\boldsymbol{\hat{x}_i}$ represent the original and the estimated image vectors respectively. r and c represent number of rows and columns of the image respectively.

Figure 5 shows the results of applying the Gaussian band-pass filter in the quaternion spectral domain to the quaternion fast Fourier transform of an image (Fig.4). The higher cutoff frequency of the filter is changed gradually from 110 down to 20. The filtered images are shown in Fig.4-1 to Fig.4-10 accompanied with the image of the filter.

The results show that the PSNR value decreases with each decrease in the higher cutoff frequency of the Gaussian filter. This can be explained as: the noise pixels have high frequency components in the frequency domain, the number of rejected pixels will increase when the cutoff frequency decreases. This leads to a higher difference between the noisy image and the filtered image and accordingly lower PSNR value.

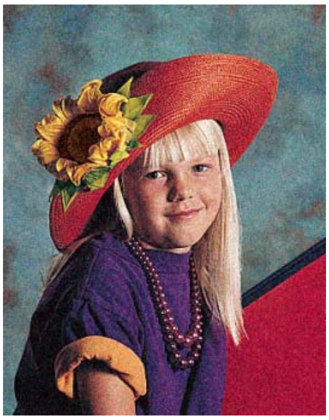


Fig.4 The original noisy image used as test image yielding to the images in Fig.5

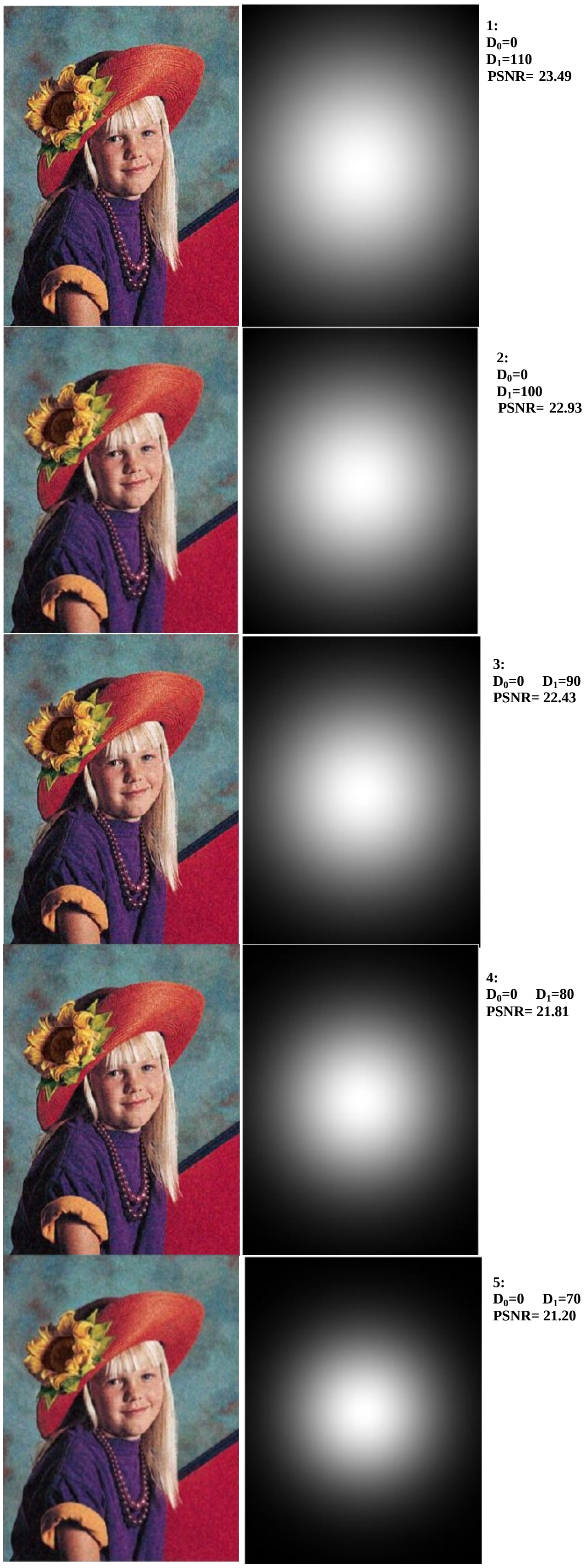


Fig.5 The results of applying the Gaussian band-pass filter to the noisy image of Fig.4 with different cutoff values (part-1).

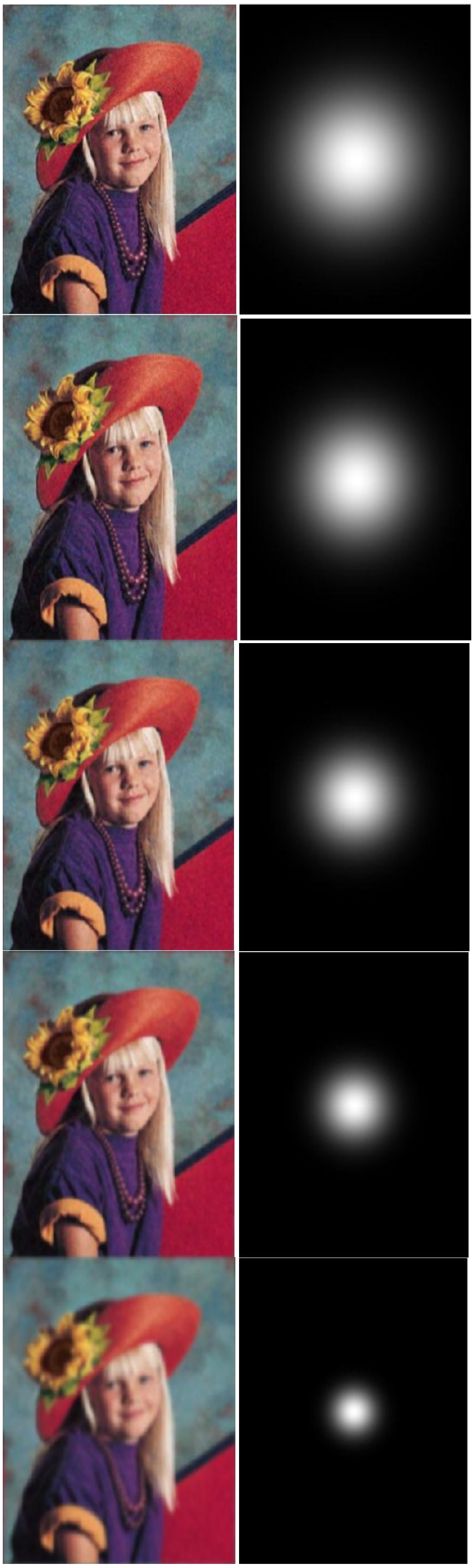


Fig.5 The results of applying the Gaussian band-pass filter to the noisy image of Fig.4 with different cutoff values (part-2).

IV. CONCLUSION

The Fourier transform of Hypercomplex provides a tool to analyze color images in the whole view. Color images can be represented mathematically as quaternion valued images which can be transformed into the frequency domain using the Quaternion Fast Fourier Transform (QFFT) allowing different image processing techniques to be carried out. The advantage of the quaternion representation for color is that it combines a color 3-tuple (RGB) into a single Hypercomplex number. This work demonstrates the application of the Gaussian band-pass filter on the basis of QFT. Filtering in quaternion frequency domain allows higher color information accuracy to be achieved because a color is treated as an entity.

REFERENCES

[1] Cooley, James W., and John W. Tukey, 1965", An algorithm for the machine calculation of complex Fourier series", *Math. Comput, Vol.*19, 1965, pp. 297-301.

[2] Sangwine, S., Ell, T.A., "Hypercomplex Fourier Transforms of Color Images", IEEE International Conference on Image Processing (ICIP), 2001, Vol. 1, pp. 137–140.

[3] Pei, S., Cheng, C., "A novel block truncation coding of color images by using quaiemion-moment preserving principle", IEEE International Symposium on Circuits and systems, 1996, Vol. 2, pp. 684–687.

[4]. Sangwine, S., Ell, T, "Hypercomplex fourier transforms of color images", IEEE International Conference on Image Processing, 2001, Vol.1, pp. 137–140.

[5] Bihan, N.L., Sangwine, S.J., "Quaternion principal component analysis of color images", IEEE International Conference on Image Processing, 2003, Vol, 1, pp. 809–812.

[6] Ja-Han Chang, S.C.P., Ding, J.J., "2d quaternion fourier spectral analysis and its applications", IEEE International Symposium on Circuits and Systems, 2004, Vol. 3, pp. 241–244.

[7] S. J. Sangwine, "Fourier transforms of colour images using quaternion, or hypercomplex, numbers", Electron. Lett. Vol. 32, No. 21, 1996, pp.1979–1980..

[8] Ell T.A., "Quaternion-Fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems," in Proc. 32nd Con. Decision Contr., Dec. 1993, pp. 1830-1841.

[9] S. J. Sangwine and T. A. Ell, "The discrete Fourier transform of a colour image", in Proc. Image Processing II Mathematical Methods, Algorithms and Applications, J. M. Blackledge and M. J. Turner, Eds., Chichester, U.K., 2000, pp. 430–441.

[10] Sangwine S., "Colour image edge detector based on quaternion convolution", Electronic Lett, Vol. 34, No. 10, 1998; pp. 969–971.

[11] Moxey C, Sangwine S, "Ell T. Hypercomplex correlation techniques for vector images", IEEE Transactions on Signal Processing, Vol. 51, No.7, 1941.

[12] Subakan ÖN, Vemuri BC, "Image segmentation via convolution of a level-set function with a Rigaut kernel",

IEEE Conference on Computer Vision and Pattern Recognition; Anchorage, Alaska. June 2008.

[13] Chan TF, Yezrielev B, Vese LA., "Active contours without edges for vector-valued", images. Journal of Visual Communication and Image Representation, Vol. 11, 2000, pp. 130–141.

[14] Sangwine, S.J., "Colour in Image Processing", Electronics and Communication Engineering Journal, Vol. 12, No. 5, pp.211-219, October 2000.

Magdi Ibrahim Khalil El-Sharkawy, male, received his B.Sc degree in Computer and Automatic Control Engineering from Ain Shams University, Cairo, Egypt, in 1983, M.Sc degree in Computer Engineering from Tanta University,Tanta, Egypt, in 2003 and Ph.D degree in Computer System Engineering from Benha University, Cairo, Egypt, in 2005. He is currently working as Assistant Professor in Department of Networking and Communication systems at the Faculty of Computer and Information Sciences, Princess Noura Bent Abdulrahman University, Riyadh, KSA. He has 15 years of previous experience at the Reactor physics Department, Nuclear Research Center, Cairo, Egypt in the field of Data Acquisition and Interface Design. His area of interest includes image processing and digital signal processing.