Local quaternion Fourier transform and color image texture analysis

Dawit Assefa a,*, Lalu Mansinha b, Kristy F. Tiampo b, Henning Rasmussen c, Kenzu Abdella d

a Radiation Medicine Program, Princess Margaret Hospital, Department of Radiation Oncology, 610 University Ave, Rm. S–612 Toronto, Ontario, Canada M5G 2M9
b University of Western Ontario, Department of Earth Sciences, London, Ontario, Canada
c University of Western Ontario, Department of Applied Mathematics, London, Ontario, Canada
d Trent University, Department of Mathematics, Peterborough, Ontario, Canada

Article history:
Received 21 January 2009
Received in revised form 9 November 2009
Accepted 30 November 2009
Available online 11 December 2009

Abstract
Color images can be treated as two-dimensional quaternion functions. For analysis of quaternion images, a joint space-wavenumber localized quaternion S transform (QS) is presented in this study for a simultaneous determination of the local color image spectra. The QS transform uses a two-dimensional Gaussian localizing window that scales with wavenumbers. Rotation invariance, invertibility and computational aspects of the QS transform are discussed. The power map of the QS transform is presented here as a powerful tool in color image texture and pattern analysis. Examples are presented.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Fourier transforms have been widely used to analyze grayscale images in different applications [1]. Local spectral properties at any point on a grayscale image may be determined by joint space-wavenumber transforms such as the Gabor transform (space-localized Fourier transform) [2,3], the wavelet transform [4–6] and, more recently, the S transform [7–11]. For analyzing color images with multiple components [12,13] the application of these transforms is very limited. It is possible to separate a color image into three (or four) scalar images and analyze each component separately using these transforms. However, separate spectral analysis of each component color of an image leaves us ignorant of information about the coupled spectra of all color components.

The idea of computing the Fourier transform of color images as one quantity has been realized recently through use of quaternions and the quaternion Fourier transform has been introduced for efficient analysis of color images [14–17]. The use of a quaternion representation allows the analysis of a color image as a vector field. Design of quaternionic filters [18,19], color image vector correlations [17], and color image spectrum [14] are amongst specific applications where the quaternionic approach of analyzing color images proved to be vital. Other interesting studies with quaternions include the design of quaternion principal component analysis [20], and the quaternion matrix singular value decomposition [21].

In [22] quaternionic Gabor filters have been used for the classification of local image structures. The main drawback of the Gabor transform is the use of a fixed width localizing window which gives rise to a fixed resolution. The wavelet transform offers a better representation by having a progressive resolution. In that regard, the quaternion wavelet transform has been introduced [23,24]. The wavelet transform has two major limitations. First, the transform offers sparse information and second, it suffers from spectral overlaps. The S transform offers a very useful extension to the wavelets by having a wavenumber dependent progressive resolution as opposed to the arbitrary dilations used in wavelets.
The kernel of the S transform, as opposed to the wavelet counterpart, does not translate with the localizing window function. For that reason the S transform not only localizes the amplitude/power but also keeps the absolute phase information in the sense that the origin (0,0) is taken as a fixed reference point. This is not the case for the wavelet transform as it localizes only the amplitude/power spectrum but not the absolute phase. In addition, there is an easy and direct relation between the S transform and the natural Fourier transform which is not the case for wavelets. However, the S transform has limitations of its own. The major one is that it is not rotation invariant. In that regard, for an efficient use in color image analysis, this study presents the quaternion S transform based on a generalized and rotation invariant S transform.

One application of image analysis that has been studied for a long time is texture analysis [25–28]. While the word texture does not have a precise definition due to its wide variability, texture can be defined as the property that represents the surface or structure of an object (an image in our case). A texture is made of texture elements that are represented by a contiguous set of pixels. Texture is often defined subjectively with terms such as fine-grained, smooth, or coarse, for example. Attempts to objectively quantify texture generally rely on the intensity of the pixels that make up the texture and the spatial relationship between them, in conjunction with the total number of texture elements.

Various methods have been used in the literature for extracting textural properties from color images. Non-quaternion based methods for example have been used in color texture classification [29] and segmentation [30]. The quaternionic approach has also been used. For example, in [31–33] quaternions in conjunction with principal component analysis have been used in texture segmentation of color images. In [19] quaternions have been used in edge detection of color images. In this paper we suggest that the QS transform provides a powerful tool in extraction of textures and patterns in color images.

The rest of the paper is organized into sections. The quaternion Fourier transform is introduced in Section 2 while the localized QS transform is defined with a scalable 2D Gaussian localizing window function in Section 3. In addition, a texture algorithm is developed based on the definition of the power map. Examples are presented in Section 4 followed by a discussion of some additional issues in Section 5. Finally, concluding remarks are presented in Section 6.

2. Quaternions and the quaternion Fourier transform

A quaternion $q$ has one real and three imaginary components [34]. Thus

$$q = a + ib + jc + kd$$

(1)

where $a$, $b$, $c$ and $d$ are real numbers and $i$, $j$ and $k$ are orthonormal complex operators satisfying

$$i^2 = j^2 = k^2 = ijk = -1$$

The skew field of quaternions, denoted by $\mathbb{Q}$, is a 4D, non-commutative field over the field of real numbers $\mathbb{R}$ with four base elements $\{1, i, j, k\}$ where 1 is the multiplicative identity element.

Quaternion algebra is distinct from both real and complex algebra [34]. For example multiplication is not commutative in quaternion space. Below we review some properties of quaternions.

2.1. Properties of quaternions

(a) We can express a given quaternion number $q = a + ib + jc + kd$ as sum of a real part and a vector part as $q = S(q) + V(q)$ where $S(q) = a$ is the real part of the quaternion and $V(q) = ib + jc + kd$ the vector part.

(b) Any quaternion $q = a + ib + jc + kd$ can be written in polar form as [35]

$$q = |q|e^{i\phi}$$

(3)

where $|q| = \sqrt{a^2 + b^2 + c^2 + d^2}$ is the amplitude (modulus), $\mu = V(q)/|V(q)|$ is the eigen axis, and $\phi = \text{arctan}(\mu)$ is the eigen angle (phase).

When $|q| = 1$ $q$ is a unit quaternion and when $a = 0$ it is a pure quaternion. Euler’s and De Moivre’s formulae still hold in quaternion space; i.e. for a pure unit quaternion $\mu$ the following holds:

$$e^{i\phi} = \cos(\phi) + \mu \sin(\phi)$$

(4)

$$e^{i\phi} + e^{i\phi} = e^{i2\phi} = \cos(n\phi) + \mu \sin(n\phi)$$

(5)

(c) Quaternion conjugate is given by $\overline{q} = a - ib - jc - kd$.

(d) A complex number can be considered as a special quaternion number with one real and one imaginary part.

(e) The skew field of quaternions $\mathbb{Q}$ is also identified as 2D complex space $\mathbb{C}_2$. For example if $q = a + ib + jc + kd$ then $q = S_1 + S_2 j$ where $S_1 = a + ib \in \mathbb{C}_i$ and $S_2 = c + id \in \mathbb{C}_i$. Similarly $q$ can be identified as 2D complex space $\mathbb{C}_1$ or $\mathbb{C}_2$. For that reason the quaternions are sometimes termed hypercomplex numbers.

2.2. The quaternion Fourier transform

The quaternion (or hypercomplex) Fourier transform (QFT) is defined similar to the standard Fourier transform of 2D functions. The non-commutative nature of quaternion multiplication allows for the definition of different types of QFT [14–16,35–42]. The dominant ones in literature are listed below.

The QFT of type I [35,38,39] is given by

$$Q(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi i f(x,y)} e^{-ik_1 u} e^{-ik_2 v} dx dy$$

(6)

where $x$ and $y$ represent spatial variables while $u$ and $v$ are the wavenumbers (inverse of wavelengths) in the respective directions and $f(x,y)$ is a quaternion. The inverse type I QFT differs from the forward only at the sign of the exponents in the two exponential terms.
The discrete variant of (6) was first suggested in [43] as an alternative to reduce the computational complexity of the complex discrete Fourier transform.

A QFT of type II \([14,39,42]\) is given by

\[
Q(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\mu x^2 + \nu y^2} f(x, y) \, dx \, dy
\]  

(7)

where \(\mu\) is any pure unit quaternion, \(\mu^2 = -1\). The formula for the inverse type II QFT differs from the forward one only at the sign of the exponent in the exponential term. The Fourier transform is a special case in which \(\mu = i\). The parameter \(\mu\) is chosen arbitrarily. For an RGB color image \(f(x, y)\) for example, where the image has three components which can be written as a quaternion, \(f(x, y) = R(x, y) + G(x, y)j + B(x, y)k\),\(^1\) we customarily use \(\mu = (i+j+k)/\sqrt{3}\). This choice coincides with the gray line in the RGB space with all three components equal.

The discrete counterparts of (7) have also been used previously \([44–48]\). In [35] the double-complex algebra has been defined very similar to the quaternions but with commutative multiplication. In [49] the commutative hypercomplex numbers (Davenport’s hypercomplex numbers) and the associated algebra was proposed and this gives rise to the commutative hypercomplex Fourier transform \([40,44]\). While the commutative property in the RGB space with all three components equal, the double-complex algebra and Davenport’s hypercomplex numbers have a crucial advantage over Hamilton’s quaternions, the major drawback is that it is not a division algebra. Not all nonzero double-complex numbers \([35]\) and hypercomplex numbers \([49]\) have multiplicative inverses. That means care is needed during computation and this can cause problems in defining and computing Fourier transforms in these algebra. Still not a division algebra, the commutative reduced biquaternions and their Fourier transform are defined in \([50]\).

3. The local quaternion Fourier transform

Analogous to the 2D Fourier transform, the QFT (6) (or (7)) gives the wavenumber spectrum of the entire color image. But for images with spatially varying spectra the whole image QFT provides inadequate information. This calls for a space-localized quaternion Fourier (QS) transform.

3.1. The QS transform

Here in defining the local transform we adopt the Type II QFT definition as many operations such as convolution are easier with the type II QFT than the type I QFT (see Appendix, for example). Without loss of generality we treat here only RGB color images. The generalization to other kinds of color images is not difficult.

Let \(f(x, y) = R(x, y)i + G(x, y)j + B(x, y)k\). Similar to the definition of the 2D S transform \([7–11]\), \(f(x, y)\) is multiplied with a translating scalable Gaussian window. The 2D S transform is given by

\[
S(X, Y, u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi(uX + vY)} \frac{|uv|}{2\pi} e^{-|uv|^2/2} f(x, y) \, dx \, dy
\]

(8)

As opposed to the Gabor transform \([2,3]\) which uses a fixed width localizing window, the S transform permits the window width to scale with wavenumber giving rise to a progressive resolution in the complex space. This idea is extended into the quaternion space below.

The 2D Gaussian function is given by

\[
w(x, y, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-x^2/2\sigma_x^2} e^{-y^2/2\sigma_y^2}
\]

(9)

The window widths along the \(x\) direction (\(\sigma_x\)) and along the \(y\) direction (\(\sigma_y\)) are proportional to the inverse wavenumbers in the respective directions such that \(\sigma_x = k_1/|u|\), and \(\sigma_y = k_2/|v|\), for some positive constants \(k_1\) and \(k_2\). The location of the Gaussian is given by \((X, Y)\). Varying the values of \(k_1\) and \(k_2\) alter the resolution of the space-wavenumber plane. Then the QS transform is given by

\[
QS(X, Y, u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi(uX + vY)}
\]

\[
\times w\left(X-x, Y-y, \frac{k_1}{|u|}, \frac{k_2}{|v|}\right) f(x, y) \, dx \, dy
\]

(10)

The QS transform is derived to have progressive resolution in both the spatial as well as the wavenumber domains with an assumption similar to the definition of the 2D S transform, though the two transforms are defined in distinct spaces. Such a setting, however, has a drawback and that limitation is discussed and a solution is suggested below.

The energy (amplitude), for example, of an image should be the same irrespective of the orientation of the image and any transform applied to this image should essentially be able to demonstrate that. This property is rotation invariance. Unfortunately, the QS transform (10) is not rotation invariant. If we rotate the image by any angle, the amplitude of the QS transform will change. This effect is due to the independent scaling of the Gaussian localizing window in the \(x\) and \(y\) directions, proportional to the wavenumbers \(u\) and \(v\) in the respective directions. The lack of rotation invariance of local transforms was first observed in \([11]\) for the 2D S transform.

The width of the Gaussian window in the \(x\) and \(y\) directions is proportional to the wavenumbers \(u\) and \(v\) in the respective directions. Hence the 2D Gaussian localizing window function generally has an elliptical horizontal cross section. Whenever either of the two wavenumbers \(u\) or \(v\) goes to zero the window width in that direction goes to infinity. Special definition of (10) is then required for the limiting case \(u = 0\) (or \(v = 0\)).

In (10) we had assumed \(\sigma_x \sim 1/|u|\) and \(\sigma_y \sim 1/|v|\). To achieve rotation invariance we used a circularly symmetric Gaussian with window width inversely proportional to the wavenumbers such that \(\sigma = |r|/\sqrt{u^2 + v^2}\).
where \( r \) is a constant. The QS transform is now
\[
\text{QS}(X, Y, u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i 2\pi(uX + vy)} u^2 + v^2 \frac{1}{2\pi^2} f(x, y) \, dx \, dy
\]
(11)

In this work, a value of unity has been assumed for \( r \). However, by varying the value of \( r \) we can change the resolution of the space-wavenumber plane as required.

**Theorem 1.** The QS transform is rotation invariant.

**Proof.** To show that the modified QS transform (11) is rotation invariant with the modified Gaussian, let the original image \( f(x, y) \) be rotated by an angle \( \theta \) such that
\[
x' = x \cos(\theta) + y \sin(\theta), \quad y' = -x \sin(\theta) + y \cos(\theta)
\]
\[
u' = u \cos(\theta) + v \sin(\theta), \quad v' = -u \sin(\theta) + v \cos(\theta)
\]
\[X' = X \cos(\theta) + Y \sin(\theta), \quad Y' = -X \sin(\theta) + Y \cos(\theta)
\]
Then the following is true:
\[
u'^2 + v'^2 = u^2 + v^2
\]
(12)
\[(X' - X)^2 + (Y' - Y)^2 = (X - x)^2 + (Y - y)^2
\]
(13)
\[u'v' + v'u' = u\cos(\theta) + v\sin(\theta)
\]
(14)

Hence from the above we can see that
\[
\cos(2\pi u'x' + 2\pi v'y') = \cos(2\pi ux + 2\pi vy)
\]
(15)
\[
\sin(2\pi u'x' + 2\pi v'y') = \sin(2\pi ux + 2\pi vy)
\]
(16)

Moreover, we have
\[
dx'\,dy' = \left| \frac{\partial(x', y')}{\partial(x, y)} \right| \, dx \, dy
\]
(17)

But
\[
\left| \frac{\partial(x', y')}{\partial(x, y)} \right| = \left| \begin{array}{cc}
\frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\
\frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y}
\end{array} \right| = \left| \begin{array}{cc}
\cos(\theta) & \sin(\theta) \\
-\sin(\theta) & \cos(\theta)
\end{array} \right| = 1
\]

Hence \( dx'\,dy' = dx\,dy \) and the QS transform is rotationally invariant. \( \square \)

The following theorem demonstrates the invertibility of the QS transform.

**Theorem 2.** The QS transform is invertible.

**Proof.** Taking the double space integral of the QS transform as given in Eq. (11), we obtain the following:
\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{QS}(X, Y, u, v) \, dX \, dY = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i 2\pi(uX + vy)} f(x, y) \, dx \, dy
\]
(18)

But the RHS of Eq. (18) is the QFT of the image function \( f(x, y) = R(x, y) + G(x, y) + B(x, y)k \). Thus
\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{QS}(X, Y, u, v) \, dX \, dY = Q(u, v)
\]
(19)

Taking the inverse QFT of both sides of (19), we obtain the original quaternion valued image function. Hence we can recover the original image function from the QS transform using
\[
f(x, y) = R(x, y) + G(x, y) + B(x, y)k
\]
(20)

An example is presented in Fig. 1 to show why the circular symmetry of the Gaussian is important. We generated a 2D \( N \times N \) gray scale image with frequency changing only in the \( x \) direction,
\[
f(x, y) = 10\cos(2\pi 0.15x^2/N)
\]
(21)

for \( x = 0, \ldots, N-1 \). Multiplying (21) with a normalized Gaussian function gives an image that goes to zero towards the edges (Fig. 1(a)). Several slices of the QS transform corresponding to \( v = 0 \) and different values of \( u \) are shown in Fig. 1(b) and (c). The amplitude spectrum of the QS transform should go to zero towards the edges. In Fig. 1(b) the QS transform is computed using (10) and we see a nonzero QS amplitude spectrum at the spatial boundaries which is an artifact caused by infinite window width at small wavenumbers, and (c) amplitude of the discrete version of the rotationally invariant QS transform giving close to zero spectrum at the edges.

3.2. Computing the QS transform

The QS transform in (11) can be expressed as a convolution of the circularly symmetric Gaussian window \( w(x, y) \) and the product of \( f(x, y) \) with the kernel of the QFT. Since the Gaussian window is a real valued even function,
the convolution in quaternion space is similar to convolution in the standard complex Fourier space (see Appendix).

There are fast QFT algorithms discussed in the literature (42,51) for type I QFT and (44,52) for type II QFT, for example. But those are designed based on the standard fast Fourier transform which requires decomposition of the QFT into complex Fourier transforms. In this study we used the 2D Hartley transform (53–59) instead without the need to decompose the QFT by applying Hartley convolutions on each component of the QS transform.

We used the following simplifications to rewrite the QS transform with its real and imaginary parts separated. Let \( \theta = 2\pi(u_x + iv_y) \), and the image function \( f(x, y) = iR(x, y) + jG(x, y) + kB(x, y) \) which \( \mu = (i + j + k)/\sqrt{3} \). Then \( e^{-\mu \theta}f(x, y) = e^{-[(i + j + k)/\sqrt{3}]\theta}iR(x, y) + jG(x, y) + kB(x, y) \). But the quaternion \( \mu \) satisfies \( e^{-\mu \theta} = \cos(\theta) - j\mu \sin(\theta) \). Hence we have the following:

\[
e^{-\mu \theta}f(x, y) = \left( \cos(\theta) - \frac{i + j + k}{\sqrt{3}} \sin(\theta) \right) iR(x, y) + jG(x, y) + kB(x, y)
\]

\[= q_0(x, y) + iq_1(x, y) + jq_2(x, y) + kq_3(x, y) \quad (22)\]

where

\[q_0(x, y) = \frac{\sin(\theta)}{\sqrt{3}} (R(x, y) + G(x, y) + B(x, y)) \quad (23)\]

\[q_1(x, y) = R(x, y)\cos(\theta) + (G(x, y) - B(x, y))\frac{\sin(\theta)}{\sqrt{3}} \quad (24)\]

\[q_2(x, y) = G(x, y)\cos(\theta) + (B(x, y) - R(x, y))\frac{\sin(\theta)}{\sqrt{3}} \quad (25)\]

\[q_3(x, y) = B(x, y)\cos(\theta) + (R(x, y) - G(x, y))\frac{\sin(\theta)}{\sqrt{3}} \quad (26)\]

Using the above, the QS transform can be written as follows:

\[\text{QS}(X, Y, u, v) = \frac{u^2 + v^2}{2\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( q_0(x, y) + iq_1(x, y) + jq_2(x, y) + kq_3(x, y) \right) e^{-i(X-x)^2 + i(Y-y)^2 + i(u^2 + v^2)/2r^2} \, dx \, dy \quad (27)\]

Each (real valued) component of the QS transform in (27) can be written as a convolution of two functions in the Hartley space [60,61]. The 2D Hartley transform is defined as

\[H(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cos(2\pi ux + 2\pi vy) \, dx \, dy \quad (28)\]

while the inverse is given by

\[f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(u, v) \cos(2\pi ux + 2\pi vy) \, du \, dv \quad (29)\]

where \( \text{cas}(x) = \cos(x) + \sin(x) \).

The convolution formula in the Hartley space is given by

\[C(u, v) = \frac{H_1(u, v)H_2(u, v) + H_1(u, v)H_2(-u, -v)}{2} + \frac{H_1(-u, -v)H_2(u, v) - H_1(-u, -v)H_2(-u, -v)}{2} \quad (30)\]

where \( H_1 \) and \( H_2 \) are the Hartley transforms of two image functions \( h_1 \) and \( h_2 \), respectively, and \( C \) is the Hartley transform of the convolution. Consider the real term of the QS transform (QS\(_0\)), for example. It can be written as a convolution as follows:

\[\text{QS}_0(X, Y, u, v) = q_0(x, y) \ast d(x, y) \quad (31)\]

where \( d(x, y) = \left[ (u^2 + v^2)/2\pi^2 \right] e^{-(X-x)^2 + (Y-y)^2/(u^2 + v^2)/2r^2} \), \( q_0 \) is as defined above, and \( \ast \) denotes convolution in the Hartley space. One of the convolved functions, \( d \), is even. Hence the 2D Hartley convolution in (30) reduces to \( H_1(u, v)H_2(u, v) \), where \( H_1 \) is the Hartley transform of \( q_0 \) and \( H_2 \) is the Hartley transform of \( d \) (from \( x \), \( y \) to \( \alpha, \beta \)) given by

\[H_2(u, v, \alpha, \beta) = e^{-2\pi^2r^4(x^2 + \beta^2)/(u^2 + v^2)} \]

Then QS\(_0\) is simply the inverse Hartley transform (which is identical to the forward Hartley transform) of \( H_1(u, v)H_2(u, v) \).

\[\text{QS}_0(X, Y, u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H_1(x, \alpha, u, v)e^{-2\pi^2r^4(x^2 + \beta^2)/(u^2 + v^2)} \times \cos(2\pi x \alpha + 2\pi \beta Y) \, dx \, d\beta \quad (33)\]

Similarly the three imaginary components of the QS transform can be computed very easily. Computationally this operation is extremely cheaper (four real multiplications and three real additions) than the corresponding quaternion convolution given in the Appendix (product of two quaternions).

The 2D complex Fourier transform cannot be used directly for the computation, because the Fourier transform is itself complex and has a different interpretation in quaternion space. However, the 2D Hartley transform is real valued and can be applied to each component of the QS transform directly. Moreover, the forward and the inverse Hartley transforms use the same kernel, which makes the transform more convenient in applications.

In order to compute the inverse QS transform in (20), we can employ the fast QFT algorithms referenced earlier in this section and that requires the decomposition of the QFT into complex Fourier transforms. However, it is not clear if we still can do the inverse QS computation in the Hartley space.

### 3.3. The power map of the QS transform

The power (square amplitude) spectrum, \( P \), of a color image can be computed from its QS transform as

\[P(X, Y, u, v) = \text{QS}^2_0 + \text{QS}^2_1 + \text{QS}^2_2 + \text{QS}^2_3 \quad (34)\]

where \( \text{QS}_0, \text{QS}_1, \text{QS}_2, \text{QS}_3 \) are the real and imaginary parts of the QS transform, respectively. The resultant power, \( RP \), at each pixel point is given by the sum of the powers contributed by all wavenumber at that point. Thus

\[RP(X, Y) = \sum_{u} \sum_{v} P(X, Y, u, v) \quad (35)\]
Then the power map is defined in terms of the logarithm of $RP$ normalized with the maximum value of powers given by

$$\frac{\log(1 + RP(X, Y))}{\log(1 + \max(RP(X, Y)))} \quad (36)$$

For an $N \times M$ quaternion image, the discrete QS transform is of size $(N \times M)^2$ and its power map is an $N \times M$ real valued matrix. In practice, the power map is computed by summing the square of the QS transform values at each wavenumber without the need to compute the entire 4D QS transform matrix upfront. Each entry of the power map is the effective local power of the corresponding pixel. Pixels or different groups of pixels can have same or distinct effective powers and we attribute this similarity and variation to their texture/pattern. Hence we are proposing here the power map as an effective way of extracting different textural regions and their patterns.

4. Examples

Fig. 2 shows a synthetic color image and its amplitude spectrum computed from the QS transform. The synthetic color image is generated with a Matlab program. In each color band the wavenumber changes only in the $x$ direction. The first component has an increasing chirp signal in the first half $x$ interval, the second component has a low wavenumber on the entire $x$ interval and a pulse in the second half $x$ interval while the third component has a medium wavenumber in the second half $x$ interval.

The 4D quaternion valued QS transform matrix is calculated and a slice of the amplitude spectrum corresponding to $v = 0$ and $y = 0$ is plotted in Fig. 2(b). All the wavenumber components are well resolved on the spectrum which is plotted in grayscale. In Fig. 2(c) the amplitude spectrum corresponding to $v = 0$ and $x = 40$ is plotted showing no variation along the $y$-axis. The QS spectral approach has two major
advantages over monochromatic analysis of each color component. The first one is that the whole spectral information can be visible. The second advantage is computational. Using convolutions in the Hartley space as explained in the previous section, we only need four 4D real matrices to store the QS transform values, as compared to the six 4D real matrices required to store complex Fourier transform of each monochromatic component separately. This is a considerable memory saving particularly when dealing with large size images.

Subsequent examples are presented below in order to demonstrate the use of the QS transform in image texture extractions. This procedure utilizing the power map of the QS transform is a vital step which can help in further analysis such as image auto-segmentation and texture based classification. The robustness of texture based image processing applications such as segmentation mainly relies on how good we can extract the different textures and patterns that the image is comprised of. This task is even more challenging when the color image under consideration has difficult patterns such as, for example, if the color variations within the image are blended together with textures (see for example Figs. 5(a) and 6(a), and also the top left and bottom right regions in Fig. 3(a)).

A first example color image (from VisTex database) is shown in Fig. 3(a) and the corresponding power map is shown in Fig. 3(b). The original image contains four distinct oriented textural regions with the top right and bottom left regions having similar colors and the computed power map picked all regions correctly.

Another example (reprinted from [32]) and its power map are shown in Fig. 4(a) and (b), respectively. In this case, the top left and bottom left regions have same color but different grayscale structures while the bottom left and bottom right regions have the same grayscale structures but different colors. The power map picked all the different textural regions. As the bottom left and bottom right regions are only different in color, both appeared texturally similar on the power map.

In order to evaluate the performance of the power map of the QS transform in color images with more complex patterns, two more examples are presented in Figs. 5(a) and 6(a) with the corresponding power maps in Figs. 5(b) and 6(b), respectively. The original color images are freely available at http://www.freephotoshop.com/html/free_textures.html. The color patterns of these two color images are less clear compared to the once presented
above. In particular, it is very difficult to recognize the patterns in Fig. 6(a) unless we look closely. In contrast the texture map made the patterns clearly visible even from a distance.

We have to give a remark that the QS transform can also be applied on gray scale images. In [41] it has been shown the advantages of using the QFT to gray level (real valued) images in order to analyze symmetries in the image. A grayscale image is represented in the quaternion space as a pure quaternion with all three components equal to the gray value (normalized by $1/\sqrt{3}$). The QS transform of a grayscale image would be a quaternion giving the same amplitude and (eigen) phase information like the complex S transform. In this case the eigen axis would be the vector $(1, 1, 1)/\sqrt{3}$ or the gray line. However, the QS transform and the S transform of a real image will have different symmetries in the $u, v$ plane. An example is given in Fig. 7 for demonstration.

5. Additional issues

Only the modulus of the QS transform is used in our texture algorithm. Two other attributes of a quaternion have been defined in Section 2. The phase is real and the eigen axis is generally a quaternion. Some authors define the quaternion phase as a vector with three components each of which has a different meaning and this concept has been used in classifying local image structures [22]. Further development of the QS method is needed to determine use of the eigen angle (phase) and the eigen axis. In particular, the QS transform, which is an extension of the complex S transform to the quaternion space, keeps the absolute phase information which is useful in many applications.

There are many more potential applications of the transform. Most applications such as image filtering, edge detection and segmentations done using other techniques can more conveniently be done using the QS transform. One application can be in extraction of second order statistical texture features. We have seen that the power map is able to extract dominant textures within a color image. We can combine this concept with other statistical methods to extract second order texture features. One way is to extract texture features from the gray level co-occurrence matrix (GLCM) [62] of the power map which is usually done by applying the GLCM on the raw image [63,64]. The resulting features can be used in texture based classification and auto-segmentation. Research is underway in order to implement this approach and other applications of the QS transform.
There are still some unresolved issues with the implementation of the power map of the QS transform in extracting textures. The power map is computed as the resultant local power of the QS transform involving all wavenumbers. We have not, however, said much about which wavenumbers should be included in the power map computation. In this study all the wavenumbers are included to compute the power map. However, the low wavenumbers are usually affected by edge effects and the high ones are noise. Hence the medium wavenumbers have better efficiencies in extracting textural features. That means we need to threshold the wavenumbers, although, finding an optimal value is not a trivial task.

6. Conclusion

In summary, this study presented a space localized quaternion based integral (QS) transform that shows great promises in color image analysis. The QS transform allows local analysis of all color components simultaneously, as compared to separate analysis of color components. The QS transform is invertible and is rotation invariant. It is also shown how the QS transform can easily be computed in the Hartley space using convolutions. The power map of the QS transform is found to be useful in extraction of dominant color textures. More research is needed to understand the usefulness of the two attributes of the QS transform other than the amplitude, i.e. the local eigen angle (phase) and local eigen axis.

Acknowledgements

The idea of using quaternions in local Fourier transforms arose in past discussions between one of the authors L. Mansinha and Dr. R. Pinneger, who was a doctoral student at the time.

We have implemented the discrete form of the QS transform in parallel. Special thanks to SHARCNET, the High Performance Computing Consortium, located at the University of Western Ontario, for allowing us to use the facilities.

This work has been supported by a NSERC fellowship to D. Assefa, and through a NSERC discovery grant to H. Rasmussen and K. Abdella. The work by K.F. Tiampo was funded by a NSERC discovery grant and the NSERC and Aon Benfield/ICLR Industrial Research Chair.

Appendix A. Convolution using the quaternion Fourier transform

The convolution of two functions $f(x,y)$ and $g(x,y)$ in the quaternion space is given by

$$f(x,y) * g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-s,y-t)g(s,t) \, ds \, dt$$

where $*_{q}$ denotes the quaternion convolution operator. Clearly quaternion convolution is not commutative. The convolution theorems for the two types of the QFT are presented below. We skipped the proofs as it can easily be shown to hold. The quaternion convolutions of both type I and type II QFTs are presented in [39] which require the decomposition of the QFT into complex Fourier transforms. The type II convolution is shown in a different form in [14]. That also requires the decomposition of one of the original convolved functions into its symplectic form.

Theorem 3 (Quaternion convolution: type I). Let $f_1(x,y) = f_1(x,y) + f_2(x,y)i + f_3(x,y)j + f_4(x,y)k$ and $g(x,y) = g_1(x,y) + g_2(x,y)i + g_3(x,y)j + g_4(x,y)k$ be two quaternion valued functions and $c(x,y) = f(x,y) * g(x,y)$ be the convolution between them. Let also $G_0 = G_1 + iG_2 + jG_3 + kG_4$, $F_1$, $F_2$, $F_3$, and $F_4$ be the QFTs of $f_1$, $f_2$, $f_3$, and $f_4$, respectively. Then the QFT, $C$, of the convolution, $c$, is given by

$$C(u,v) = (F_1(u,v) + jF_2(u,v))G_0(u,v) + (F_1(u,-v) + jF_2(u,-v))G_0(u,v)$$

$$+ kF_3(u,v) + jG_4(u,v) + (jF_3(u,v) + kG_4(u,v))$$

$$+ kG_4(u,v) + jF_3(u,v))G_0(u,v) + (jF_3(u,v) + kG_4(u,v))$$

Corollary 1. When one of the convolved functions $f(x,y)$ is real valued then the above theorem reduces to

$$C(u,v) = F(u,v)G_0(u,v) + G_4(u,v)F(u,v)$$

where $F$ is the QFT of $f$.

Proof. The proof directly follows from the above theorem.

Theorem 4 (Quaternion convolution: type II). Let $f_1(x,y) = f_1(x,y) + f_2(x,y)i + f_3(x,y)j + f_4(x,y)k$ and $g(x,y) = g_1(x,y) + g_2(x,y)i + g_3(x,y)j + g_4(x,y)k$ be two quaternion valued functions and $c(x,y) = f(x,y) * g(x,y)$ be the convolution between them. Let also $F_1$, $G_1$, $G_2$, $G_3$, and $G_4$ be the QFTs of $f_1$, $g_1$, $g_2$, $g_3$, and $g_4$, respectively. Then the QFT, $C$, of the convolution, $c$, is given by

$$C(u,v) = G_1(u,v)F(u,v) + G_2(u,v)F(u,v)i + G_3(u,v)F(u,v)j$$

$$+ G_4(u,v)F(u,v)k$$

Corollary 2. When one of the convolved functions $f(x,y)$ is real valued then the above theorem reduces to

$$C(u,v) = F(u,v)G_0(u,v)$$

Proof. For the special case where $f(x,y)$ is real valued the following is true:

$$C(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi umx + 2\pi vnt} f(x-s,y-t)$$

$$\times g(s,t) \, dx \, dy \, ds \, dt$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi umx + 2\pi vnt}$$

$$\times f(m,n)e^{-i2\pi umx + 2\pi vnt} g(s,t) \, dm \, dn \, ds \, dt$$

$$= F(u,v)G_0(u,v)$$

where $F$ and $G$ are the QFTs of $f$ and $g$, respectively.

Appendix B. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.sigpro.2009.11.031.
References


