

Fast complexified quaternion Fourier transform

Salem Said[†]

Nicolas Le Bihan[†]

Stephen J. Sangwine^{‡§}

February 2, 2008

Abstract

A discrete complexified quaternion Fourier transform is introduced. This is a generalization of the discrete quaternion Fourier transform to the case where either or both of the signal/image and the transform kernel are complex quaternion-valued. It is shown how to compute the transform using four standard complex Fourier transforms and the properties of the transform are briefly discussed.

1 Introduction

Discrete quaternion Fourier transforms have been described by several authors [2, 5, 10, 12, 14]. The pioneering work of Ell [3] predated all of these papers and was the inspiration for [12, 14]. Quaternion, or at least vector-valued, signals arise in color image processing and in processing of signals captured by vector sensors, for example, vector geophones [9]. The main motivation for using quaternion algebra to handle such signals is the correspondence between vector samples and the vector parts of quaternions, which permits holistic processing of the vector samples and signals, making use of information about the direction and magnitude of the samples in sample space.

Since the introduction of the quaternion Fourier transform there has been a concern that perhaps a quaternion-valued signal or image requires a transform based on a ‘higher’ algebra, just as the Fourier transformation of real-valued signals or images requires a complex transform. Indeed Bülow and Sommer [2] have shown the advantages of using a quaternion transform even for real (grey-level) images, in order to analyse symmetries in the image. Unlike Bülow and Sommer, we are interested in signals and images with vector samples (that is, three components at least, perhaps even three *complex* components).

However, until now there have been no reported definitions of Fourier transforms for quaternion-valued signals or images based on an algebra of higher dimension than the real quaternions*.

At various times the octonions have been suggested as the basis for a higher transform, but their algebra is non-associative and therefore presents formidable problems compared to the quaternion algebra which is of course non-commutative, but otherwise relatively straightforward to work with. Therefore we have considered the case of a transform based on complexified quaternions, that is quaternions with four complex components. These can be considered equivalently as complex numbers with quaternion real and imaginary parts. The properties of this and other quaternion algebras have been extensively studied in the abstract algebra community, but much less knowledge is available at the practical algebraic level needed for engineering use. The complexified quaternions were studied by Hamilton himself [8] and [7, Lecture VII, § 669 (p.664)],

[†]Laboratoire des Images et des Signaux, CNRS UMR 5083, ENSIEG, 961 Rue de la Houille Blanche, Domaine Universitaire, BP 46, 38402 Saint Martin d’Hères, Cedex, France. nicolas.le-bihan@lis.inpg.fr

[‡]Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom. s.sangwine@ieee.org

[§]The work presented here was carried out at the Laboratoire des Images et des Signaux, Grenoble. Financial support from the Royal Academy of Engineering of the United Kingdom and the Centre National de la Recherche Scientifique (CNRS) in France, is gratefully acknowledged.

*Fourier transforms can be *defined* in many abstract algebras, but actually computing them numerically by a practical fast algorithm is a different matter.

but he called them *biquaternions*. We prefer to avoid this term as it was also used by Clifford with a different meaning.

In this paper we introduce the bare basics of complex quaternions necessary to the paper and we define a complex quaternion Fourier transform analogous to the quaternion Fourier transforms discussed in [14][†]. We then show how the transform may be computed. In common with real quaternion Fourier transforms, a direct (DFT or FFT) implementation is possible using a complex quaternion arithmetic package but it is also possible to factorize the transform into four complex transforms, and thus compute it using standard complex FFT code, thus exploiting the significant effort expended on performance enhancement of FFT code by other authors.

The details of quaternions and of Fourier transforms in general are outside the scope of this paper. Ward's book [15] provides a good introduction to quaternions, and there are many texts in the area of signal processing that discuss discrete and fast Fourier transforms. Generalizing the quaternions to complex quaternions is straightforward – the four components of the quaternion take complex values rather than real values, but there are some important ramifications. Firstly, although the quaternions are a division algebra, the complex quaternions are not. Any real quaternion (other than zero) has a multiplicative inverse and a non-zero norm or modulus. In contrast, a complex quaternion (other than zero) may have a zero norm or modulus and therefore no multiplicative inverse. Obviously this means that care is needed when computing with complex quaternions, and if we are to define and compute Fourier transforms with them, we must establish whether vanishing norms will cause problems. A second issue is that a quaternion Fourier transform necessarily contains as its kernel a quaternion exponential function, and if we are to generalize this to complex quaternions, we need a complex quaternion exponential function. As has been shown in [14], quaternion exponentials can be defined using an arbitrary unit pure quaternion since these are roots of -1 . (It is fundamental to Fourier transforms that a root of -1 is needed, since the transform analyses a signal into sinusoidal components, and the famous formula of de Moivre: $e^{i\theta} = \cos \theta + i \sin \theta$ generalizes to any algebra in which a root of -1 can be defined. This follows from the series expansions of the exponential, cosine and sine functions.) If we are to define a complex quaternion Fourier transform, we require a *complex* quaternion root of -1 . We cannot use a real quaternion root of -1 , or the transform will simply be a quaternion Fourier transform even if it operates on a complex quaternion signal (the transform would be equivalent to two quaternion transforms applied separately to the real and imaginary quaternion parts of the signal). The solutions of the equation $q^2 = -1$ [13] are therefore crucial to defining the transforms presented in this paper.

2 Definitions

Before defining the transform we must define some complexified quaternion concepts. We have not given proofs of these: most can be found in [15] although with different notation. A complex quaternion has four complex components, for example: $q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, where w, x, y, z are complex and \mathbf{i}, \mathbf{j} and \mathbf{k} are the three quaternion operators. We denote the complex square root of -1 by a capital \mathbf{I} to distinguish it from the quaternion \mathbf{i} with which it must not be confused (this is a fundamental aspect of complexified quaternions). Note that, although the multiplication of (complex) quaternions is not commutative, \mathbf{I} , and therefore all complex numbers (complex scalars), commute with \mathbf{i}, \mathbf{j} and \mathbf{k} , and therefore with all quaternions. In what follows, $\Re(q)$ denotes the real part of q , that is: $\Re(q) = \Re(w) + \Re(x)\mathbf{i} + \Re(y)\mathbf{j} + \Re(z)\mathbf{k}$, and similarly $\Im(q)$ denotes the imaginary part of q .

2.1 Inner product, norm and modulus

The inner product of two complex quaternions $q = w_q + x_q\mathbf{i} + y_q\mathbf{j} + z_q\mathbf{k}$ and $p = w_p + x_p\mathbf{i} + y_p\mathbf{j} + z_p\mathbf{k}$ is $\langle q, p \rangle = w_q w_p + x_q x_p + y_q y_p + z_q z_p$. The semi-norm[‡] of a quaternion $\|q\| = \langle q, q \rangle = w^2 + x^2 + y^2 + z^2$, and the

[†]A more recent and more detailed paper by Ell and Sangwine is under review.

[‡]A semi-norm is a generalization of the concept of a norm, with no requirement that the norm be zero only at the origin [1]. Here, it is possible for $\|q\| = 0$, even though $q \neq 0$ as discussed in subsection 2.3.

modulus is the square root of the norm. All of these results are complex, in general. In the special case where $\Im(q) = 0$ the semi-norm reduces to the usual quaternion norm. If the quaternion has zero real part ($\Re(q) = 0$), the semi-norm reduces to a real, but negative, norm (because $\mathbf{I}^2 = -1$). The inner product provides us with the concept of orthogonality, which is essential to the construction of a (complex) orthonormal basis. Two complex quaternions are orthogonal if their inner product is zero, that is $q \perp p \iff \langle q, p \rangle = \langle p, q \rangle = 0$.

2.2 Complex quaternion roots of -1

Let μ be a complexified quaternion root of -1 . It was shown in [13] that μ must be a pure complex quaternion satisfying the following conditions:

$$\Re(\mu) \perp \Im(\mu), \quad \|\Re(\mu)\| - \|\Im(\mu)\| = 1$$

For example, $\mu = \mathbf{i} + \mathbf{j} + \mathbf{k} + (\mathbf{j} - \mathbf{k})\mathbf{I}$ is a root of -1 and has a unit (real) norm, as can easily be verified.

2.3 Complex quaternions with null modulus

It is possible for a non-zero complex quaternion to have zero semi-norm. The conditions for the norm to vanish were discovered by Hamilton [7, Lecture VII, § 672 (p.669)]. Informally, the result can be shown as follows. Starting with the semi-norm of the quaternion in the form $\|q\| = w^2 + x^2 + y^2 + z^2$, expand one of the components in terms of its real and imaginary parts, for example w :

$$\begin{aligned} w^2 &= (\Re(w) + \Im(w)\mathbf{I})^2 \\ &= \Re(w)^2 - \Im(w)^2 + 2\Re(w)\Im(w)\mathbf{I} \end{aligned}$$

We see that the real part of the semi-norm consists of the difference between the norms of the real and imaginary parts of q , and the imaginary part of the semi-norm is twice the inner product of the real and imaginary parts of q . For the semi-norm to vanish, its real and imaginary parts must separately be zero. This requires the real and imaginary parts of q to be orthogonal real quaternions of equal norm, that is: $\langle \Re(q), \Im(q) \rangle = 0$ and $\|\Re(q)\| = \|\Im(q)\|$.

2.4 Transform pair

Given the definition of the transform ‘axis’[§] as any complexified quaternion root of -1 , μ , we can now define the transform itself, which is almost trivial. In this paper we restrict our account to the one-dimensional case, but all our definitions and results generalize to the two-dimensional case without difficulty.

$$\begin{aligned} F[u] &= \sum_{n=0}^{N-1} \exp\left(-2\pi\mu\frac{nu}{N}\right) f[n] \\ f[n] &= \frac{1}{N} \sum_{u=0}^{N-1} \exp\left(2\pi\mu\frac{nu}{N}\right) F[u] \end{aligned} \tag{1}$$

where the signal $f[n]$ and its ‘spectrum’ $F[u]$ have N samples. The placement of the minus sign in the forward transform is, of course, arbitrary. The transform pair defined in equation 1 must, of course, be evaluated in complex quaternion arithmetic, and the exponential is a complex quaternion exponential. All the necessary operations, and the transform itself, have been implemented in the *Quaternion Toolbox for Matlab*[®] library [11]. It is conventional in Matlab[®] for the scale factor $\frac{1}{N}$ to be applied to the inverse transform and we have followed this convention in implementing the complex quaternion transforms[¶].

[§]The axis of the transform is a direction in 3-space (in this paper *complex* 3-space) which defines the orientation of the sine component of the transform in the space of the vector part of the signal samples.

[¶]An alternative and widely used convention is to distribute the scale factor between the forward and inverse transforms. If the two scale factors are to be equal, each must be $\frac{1}{\sqrt{N}}$.

An alternative transform is possible, by interchanging the positions of the exponential and the function $f[n]$. This transform is closely related to the first and easily implemented, although we omit the details here.

Now we consider whether vanishing norms can occur in the transform. The norm of the exponential cannot vanish for any n or u because $\boldsymbol{\mu}$ has unit modulus, and the cosine and sine parameters are real. However, the signal $f[n]$ could contain samples with vanishing norms, unless there is some restriction on the sample values which prevents the conditions for a vanishing norm occurring. For example, if the sample values are real or imaginary quaternions they cannot have vanishing norms. In the more general case where the samples have complex quaternion values, the possibility of samples with vanishing norm cannot be eliminated. Such samples would not contribute to the ‘spectrum’ and therefore could not be reconstructed by the inverse transform. It is not known whether samples of the spectrum could have vanishing norms if computed from real (or imaginary) quaternion signals. If this did occur, it would again mean that the transform would not correctly invert. Clearly this point requires further study.

3 Factorization into four complex transforms

The definition of the transform in (1) leads directly to a discrete Fourier transform implementation, which is useful as a reference or test implementation, but useless for practical computation except for small values of N . A fast implementation requires either a custom coding of a classic FFT algorithm in complexified quaternion arithmetic, or a decomposition into complex Fourier transforms that can be implemented using existing complex functions or code. It is the latter approach that we describe here and it follows from a similar approach developed for the real quaternion case in 2000 [4].

The signal function $f[n]$ in (1) may be expressed in terms of an orthonormal basis defined by $\boldsymbol{\mu}$, the transform axis. Apart from being a complex orthonormal basis, this is exactly as in the real quaternion case. The complex orthonormal basis is defined by $\boldsymbol{\mu}$ and two other unit pure complex quaternions $\boldsymbol{\nu}$ and $\boldsymbol{\xi}$ such that $\boldsymbol{\mu} \perp \boldsymbol{\nu} \perp \boldsymbol{\xi}$ and $\boldsymbol{\mu}\boldsymbol{\nu} = \boldsymbol{\xi}$ (and $\boldsymbol{\mu}\boldsymbol{\nu}\boldsymbol{\xi} = -1$)^{||}. The basis may also be represented by a 3×3 *complex* orthogonal (*not Hermitian*) matrix:

$$\begin{pmatrix} \mu_x & \mu_y & \mu_z \\ \nu_x & \nu_y & \nu_z \\ \xi_x & \xi_y & \xi_z \end{pmatrix}$$

where $\boldsymbol{\mu} = \mu_x \mathbf{i} + \mu_y \mathbf{j} + \mu_z \mathbf{k}$ etc. We show here only the decomposition of the forward transform, as the inverse is similar. We first write $f[n]$ in terms of its four complex components:

$$f[n] = w[n] + x[n]\mathbf{i} + y[n]\mathbf{j} + z[n]\mathbf{k}$$

Now applying a change of basis, we express $f[n]$ in terms of the new basis. The change of basis is very simply implemented by resolving the vector part of each sample of $f[n]$ into the three complex directions defined by the new basis, using the inner product defined in subsection 2.1. Thus we have:

$$\begin{aligned} x'[n] &= \langle \boldsymbol{\mu}, x[n]\mathbf{i} + y[n]\mathbf{j} + z[n]\mathbf{k} \rangle \\ y'[n] &= \langle \boldsymbol{\nu}, x[n]\mathbf{i} + y[n]\mathbf{j} + z[n]\mathbf{k} \rangle \\ z'[n] &= \langle \boldsymbol{\xi}, x[n]\mathbf{i} + y[n]\mathbf{j} + z[n]\mathbf{k} \rangle \end{aligned}$$

In the new basis, we have:

$$\begin{aligned} f[n] &= w[n] + x'[n]\boldsymbol{\mu} + y'[n]\boldsymbol{\nu} + z'[n]\boldsymbol{\xi} \\ &= [w[n] + x'[n]\boldsymbol{\mu}] + [y'[n] + z'[n]\boldsymbol{\mu}]\boldsymbol{\nu} \end{aligned}$$

Using this change of basis, we are able to separate the transform into the sum of two transforms:

$$F[u] = \sum_{n=0}^{N-1} \begin{pmatrix} e^{-2\pi\mu\frac{nu}{N}} [w[n] + x'[n]\boldsymbol{\mu}] \\ + e^{-2\pi\mu\frac{nu}{N}} [y'[n] + z'[n]\boldsymbol{\mu}]\boldsymbol{\nu} \end{pmatrix}$$

^{||} $\boldsymbol{\mu} \perp \boldsymbol{\nu}$ means $\langle \boldsymbol{\mu}, \boldsymbol{\nu} \rangle = 0$

We now separate the terms on the right into real and imaginary parts and group the real components together and the imaginary components together to make four complex terms.

$$F[u] = \sum_{n=0}^{N-1} \begin{pmatrix} e^{-2\pi\mu\frac{nu}{N}} [\Re(w[n]) + \Re(x'[n])\mu] \\ + e^{-2\pi\mu\frac{nu}{N}} [\Im(w[n]) + \Im(x'[n])\mu] \\ + e^{-2\pi\mu\frac{nu}{N}} [\Re(y'[n]) + \Re(z'[n])\mu] \nu \\ + e^{-2\pi\mu\frac{nu}{N}} [\Im(y'[n]) + \Im(z'[n])\mu] \nu \end{pmatrix}$$

All four of the transforms within this expression are now isomorphic to a complex Fourier transform. That is, we may replace μ with \mathbf{I} (the complex root of -1 , not the quaternion root of -1 , \mathbf{i}) in order to compute the transform, and we will obtain the same numeric results. After computing the four complex transforms, all that remains is to re-assemble the parts of $F[u]$ and invert the change of basis. The latter step is equivalent to multiplying out the factors of μ and ν appearing above, but it is more easily performed by a change of basis using the transpose of the original basis matrix used to change from the standard basis to the (μ, ν, ξ) basis.

Note that the factorization into four complex transforms reduces to a factorization into two complex transforms in the special cases where the function to be transformed has either zero imaginary part or zero real part. If, additionally, μ is restricted to be real, the transform reduces to a previously published [4] quaternion Fourier transform. However, if the function to be transformed is either real or imaginary, the transform defined in this paper is not equivalent to a real quaternion transform, because the kernel is a *complex* quaternion exponential. Finally we note that if the function to be transformed has no vector part (that is, it is real or complex), and we choose for μ the degenerate value \mathbf{I} (the complex root of -1) the transform reduces to the usual complex Fourier transform.

The factorization described above has been used in the implementation of the transforms described in this paper in both one and two dimensions as part of the *Quaternion Toolbox for Matlab*® library [11, v0.8 onwards]. A direct DFT implementation of equation 1 is provided for verification of the fast transforms. The complex fast transforms are the usual Matlab® functions, which are implemented internally by the FFTW library [6].

4 Discussion

The complex quaternion transform applied to a real quaternion signal exhibits quaternion conjugate symmetry in the ‘spectrum’ in the same way as a complex transform applied to a real signal. The shift and similar properties follow those of the complex transform. It remains to be seen in what ways the complex quaternion transform may be useful, but one possible area is the development and fast implementation of linear vector filters based on convolution with complex quaternion coefficients. We leave a more detailed discussion of these issues for a later paper.

5 Conclusions

This paper has introduced for the first time a complex quaternion Fourier transform which may be used to compute the transform of a quaternion or complex quaternion signal or image. It may be easily computed using decomposition into four complex Fourier transforms. It appears that the transform exhibits symmetries that are absent in quaternion Fourier transforms, just as the complex Fourier transform shows symmetries for real signals that are not apparent with a real-valued transform such as the Hartley transform or the DCT.

References

- [1] E. J. Borowski and J. M. Borwein, editors. *Collins Dictionary of Mathematics*. HarperCollins, Glasgow, 2nd edition, 2002.
- [2] T. Bülow and G. Sommer. Hypercomplex signals – a novel extension of the analytic signal to the multidimensional case. *IEEE Trans. Signal Processing*, 49(11):2844–2852, November 2001.
- [3] T. A. Ell. *Hypercomplex Spectral Transformations*. PhD thesis, University of Minnesota, 1992.
- [4] T. A. Ell and S. J. Sangwine. Decomposition of 2D hypercomplex Fourier transforms into pairs of complex Fourier transforms. In Moncef Gabbouj and Pauli Kuosmanen, editors, *Proceedings of EUSIPCO 2000, Tenth European Signal Processing Conference*, volume II, pages 1061–1064, Tampere, Finland, 5–8 September 2000. European Association for Signal Processing.
- [5] M Felsberg and G Sommer. Optimized fast algorithms for the quaternionic Fourier transform. In F. Solina and A. Leonardis, editors, *Computer Analysis of Images and Patterns*, volume 1689 of *Lecture Notes in Computer Science*, pages 209–216. Springer, 1999. 8th International Conference, CAIP’99 Ljubljana, Slovenia, September 1–3, 1999 Proceedings.
- [6] FFTW. [Online], 1997–9. Software library, written by Matteo Frigo and Steven G. Johnson at Massachusetts Institute of Technology. Available: <http://www.fftw.org>.
- [7] W. R. Hamilton. *Lectures on Quaternions*. Hodges and Smith, Dublin, 1853. Available online at Cornell University Library: <http://historical.library.cornell.edu/math/>.
- [8] W. R. Hamilton. On the geometrical interpretation of some results obtained by calculation with bi-quaternions. In H. Halberstam and R. E. Ingram, editors, *The Mathematical Papers of Sir William Rowan Hamilton*, volume III Algebra, chapter 35, pages 424–5. Cambridge University Press, Cambridge, 1967. First published in *Proceedings of the Royal Irish Academy*, 1853.
- [9] Nicolas Le Bihan and Jérôme Mars. Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Processing*, 84(7):1177–1199, July 2004.
- [10] Soo-Chang Pei, Jian-Jiun Ding, and Ja-Han Chang. Efficient implementation of quaternion Fourier transform, convolution, and correlation by 2-D complex FFT. *IEEE Trans. Signal Processing*, 49(11):2783–2797, November 2001.
- [11] Quaternion Toolbox for Matlab®. [Online]. Software library, written by Stephen Sangwine and Nicolas Le Bihan. Available: <http://qtfm.sourceforge.net/>.
- [12] S. J. Sangwine. Fourier transforms of colour images using quaternion, or hypercomplex, numbers. *Electronics Letters*, 32(21):1979–1980, 10 October 1996.
- [13] S. J. Sangwine. Biquaternion (complexified quaternion) roots of -1, June 2005, arXiv:math.RA/0506190.
- [14] S. J. Sangwine and T. A. Ell. The discrete Fourier transform of a colour image. In J. M. Blackledge and M. J. Turner, editors, *Image Processing II Mathematical Methods, Algorithms and Applications*, pages 430–441, Chichester, 2000. Horwood Publishing for Institute of Mathematics and its Applications. Proceedings Second IMA Conference on Image Processing, De Montfort University, Leicester, UK, September 1998.
- [15] J. P. Ward. *Quaternions and Cayley Numbers: Algebra and Applications*, volume 403 of *Mathematics and Its Applications*. Kluwer, Dordrecht, 1997.