Quaternion Fourier Transforms for Signal and Image Processing

FOCUS SERIES

Series Editor Francis Castanié

Quaternion Fourier Transforms for Signal and Image Processing

Todd A. Ell
Nicolas Le Bihan
Stephen J. Sangwine

iSTE

WILEY

First published 2014 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2014934161

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISSN 2051-2481 (Print)
ISSN 2051-249X (Online)
ISBN 978-1-84821-478-1

MIX
Paper from responsible sources
FSC® C013604
www.fsc.org

Printed and bound in Great Britain by CPI Group (UK) Ltd., Croydon, Surrey CR0 4YY

Contents

Nomenclature

Symbol	Meaning	Section	Equation
$\mathbb{R}$	Set of real numbers	section 1.1	
$\mathbb{C}$	Set of complex numbers	section 1.1	
$\mathbb{H}$	Set of quaternions	section 1.1	
$\mathbf{V}(\mathbb{H})$	Set of pure quaternions	section 1.1	
$\mathbb{C}_{\boldsymbol{\mu}}$	Subfield of $\mathbb{H}$	section 1.6	
I	Complex root of -1	(See note below.)	
$\Re(.)$	Real part	section 1.1	
$\Im(.)$	Imaginary part	section 1.1	
$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$	Quaternion basis elements	section 1.1	[1.2]
$\boldsymbol{\mu}$	Pure quaternion	section 1.1	[1.2]
$\Im_{\boldsymbol{i}}$	$\boldsymbol{i}$-imaginary part	section 1.1	[1.4]
$\Im_{\boldsymbol{j}}$	$\boldsymbol{j}$-imaginary part	section 1.1	[1.4]
$\Im_{\boldsymbol{k}}$	$\boldsymbol{k}$-imaginary part	section 1.1	[1.4]
$\mathcal{M}_{\mathbb{R}}(.)$	Real 4×4 matrix representation	section 1.4	[1.69]
$\mathcal{M}_{\mathbb{C}}(.)$	Complex 2×2 matrix representation	section 1.4	[1.70]
$[\![.]\!]$	Alternate real 4×4 matrix representation	section 2.3	[2.20]
$[.]$	Real 4×1 vector representation	section 2.3	[2.20]
$\mathrm{S}(.)$	Scalar part	section 1.1	[1.3]
$\mathbf{V}(.)$	Vector part	section 1.1	[1.3]
(q_1, q_2)	$\mathbb{C}_{\boldsymbol{j}}$-pair notation of q	section 1.4	
$\widehat{\mathcal{A}\mathcal{B}}$	Arc of great circle between $\mathcal{A}$ and $\mathcal{B}$	section 2.2	
$\overline{q}$	Quaternion conjugate	section 1.2	[1.18]
$z^{\star}$	Complex conjugate	section 1.4	[1.54]
$\overline{q}^{\boldsymbol{\mu}}$	Involution	section 1.2	[1.23]
q^{-1}	Inverse	section 1.2	[1.26]
$\langle p, q \rangle$	Inner product	section 1.2	[1.10]
$\boldsymbol{\mu} \times \boldsymbol{\eta}$	Cross product	section 1.2	[1.9]
$\boldsymbol{\mu} \perp \boldsymbol{\eta}$	Orthogonality	section 1.2	

$q \odot p$	Bicomplex product	section 1.4	[1.65]
$\{1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$	Canonical basis in $\mathbb{H}$	section 1.1	
$\|\|q\|\|$	Norm	section 1.2	[1.14]
$\|q\|$	Modulus	section 1.2	[1.16]
$\tilde{q}$	Unit quaternion $\|\tilde{q}\| = 1$	section 1.2	[1.30]
$f * g$	Convolution	section 3	[3.1]
$f \star g$	Correlation	section 3	[3.1]
$\mathcal{F}\{.\}$	Fourier transform	section 3	[3.1]
$\mathcal{P}_{\boldsymbol{\eta}}[.]$	Reflection operator	section 2.1	[2.1]
$\mathcal{R}_q[.]$	Rotation operator	section 2.1	[2.3]
$\mathcal{S}_{\alpha,\beta,\boldsymbol{\mu}}[.]$	Shear operator	section 2.1	[2.9]
$\mathcal{D}_{\boldsymbol{\mu},\alpha}[.]$	Dilation operator	section 2.1	[2.10]
$L^1(\mathbb{G};\mathbb{K})$	Space of absolutely integrable $\mathbb{K}$-valued functions taking arguments in $\mathbb{G}$	section 3.1	
$L^2(\mathbb{G};\mathbb{K})$	Space of square integrable $\mathbb{K}$-valued functions taking arguments in $\mathbb{G}$	section 4.3	
$\mathrm{sgn}(.)$	Signum function	section 4.3	[4.31]
$p.v.(.)$	Principal value of an integral	section 4.3	[4.32]

NOTE.– The complex root of -1 which is usually denoted i, or, in engineering texts j, is denoted throughout this book by a capital letter I, in order to avoid any confusion with the first of the three quaternion roots of -1, all three of which are denoted throughout in bold font like this: $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$.

Preface

This book aims to present the state of the art, together with the most recent research results in the use of quaternion Fourier transforms (QFTs) for the processing of color images and complex-valued signals. It is based on the work of the authors in this area since the 1990s and presents the mathematical concepts, computational issues and some applications to signals and images. The book, together with the MATLAB® toolbox developed by the authors, [SAN 13b] allows the readers to make use of the presented concepts and experiment with them in practice through the examples provided.

Todd A. ELL
Minneapolis, MN, USA

Nicolas le BIHAN
Melbourne, Australia

Stephen J. SANGWINE
Colchester, UK

April 2014

Introduction

This book covers a topic that combines two branches of mathematical theory to provide practical tools for the analysis and processing of signals (or images) with three- or four-dimensional samples (or pixels). The two branches of mathematics are not recent developments, but their combination has occurred only within the last 25–30 years, and mostly since just before the millennium.

I.1. Fourier analysis

Fourier analysis was, in 1822, with Joseph Fourier's development of techniques, the first to analyze mathematical functions into sinusoidal components. In signal and image processing, Fourier's ideas underpin the two fundamental representations of a signal: one in the *time (or image) domain* where the signal (or image) is represented by samples (or pixels) with amplitudes and the other in the *frequency domain* where the signal (or image) is represented by sinusoidal frequency components, each with an amplitude and a phase. Mathematically, these concepts are not limited to time and frequency: one can use Fourier analysis on a function of any variable, resulting in a representation in terms of sinusoidal functions of that variable. However, this book is concerned with signal and image processing, and we will therefore use the terms *time* and *frequency* rather than more general concepts. It should be understood throughout that when we talk of images, the concept of time is replaced by the two spatial coordinates that define pixel position within an image.

Today, Fourier analysis is classically taught to mathematicians, scientists and engineers in several related ways, each applicable to a specific subset of mathematical functions or signals:

1) Fourier *series* analysis [SNE 61] in which continuous *periodic* functions of time, with infinite duration, are represented as sums of cosine and sine functions, each with infinite duration;

– Fourier *integrals* or transforms [BRA 00, ROB 68] in which continuous (but aperiodic) functions of time are represented as continuous functions of frequency (or *vice versa*);

2) Discrete Fourier transforms in which signals defined at discrete intervals in time are represented in the frequency domain by cosine and sine functions. This topic is broken down into:

– discrete-time Fourier transforms, in which discrete-time signals of limited duration are represented as continuous frequency-domain distributions;

– discrete Fourier transforms, in which discrete-time, discretized (that is *digital*) signals of finite duration are represented by a finite-length array of digital frequency coefficients. (These are usually computed numerically using the *fast Fourier transform* (FFT)).

The key to all of the above ideas is the representation of a signal using complex exponentials, often known as *harmonic analysis*, although this term has a somewhat wider meaning in mathematics than its usage in signal and image processing. The complex exponential with angular frequency ω and phase ϕ: $f(t) = A\exp(\omega t + \phi) = A\left(\cos(\omega t + \phi) + I\sin(\omega t + \phi)\right)$ has cosine and sine components in its real and imaginary parts, respectively. Since, in this book, we are concerned with signals that have three- or four-dimensional samples, it is helpful to consider classical Fourier analysis in terms of complex exponentials rather than in terms of separate cosines and sines.

Figure I.1 shows a real-valued signal (on the left-hand side of the plot, with time increasing away from the viewer). The signal is a sawtooth waveform reconstructed from its first five non-zero harmonics, which are plotted in the center of the figure as helices. (The horizontal spacing between the helices is introduced simply to make them clearer: there is no mathematical significance to it). The five helices on the left are the positive frequency complex exponentials and the five helices on the right are the negative frequencies. Note that the positive and negative frequency exponentials have opposite directions of rotation. The real parts of the harmonics are projected onto the right-hand side of the figure (these sum to give the reconstructed waveform on the left) and the imaginary parts of the harmonics are projected onto the base of the figure (these cancel out because the exponentials occur in complex conjugate pairs at positive and negative frequencies, a symmetry due to the original signal being real-valued).

In general, with a complex signal analyzed into complex exponentials in the same way, there would be no symmetry between the positive and negative frequency exponentials. This case is a useful model for what follows in this book, where we consider signals and images with three- and four-dimensional samples. Figure I.2 shows a complex signal constructed by bandlimiting a random complex signal.

Time is plotted on the right, increasing to the right, and at each time instant the signal has a complex value. The signal evolution over time traces out a path in the complex plane, and the figure renders this path as a three-dimensional view by plotting the signal values, in effect, on a stack of 2,000 transparent complex planes perpendicular to the time axis. The real and imaginary parts of the signal are also plotted on the base of the axes, and on the rear plane of the axes. Analysis of a complex signal into positive and negative frequency complex exponentials is not conceptually different from the real case depicted in Figure I.1: each complex exponential will have an amplitude and phase, and their sum will reconstruct the original signal.

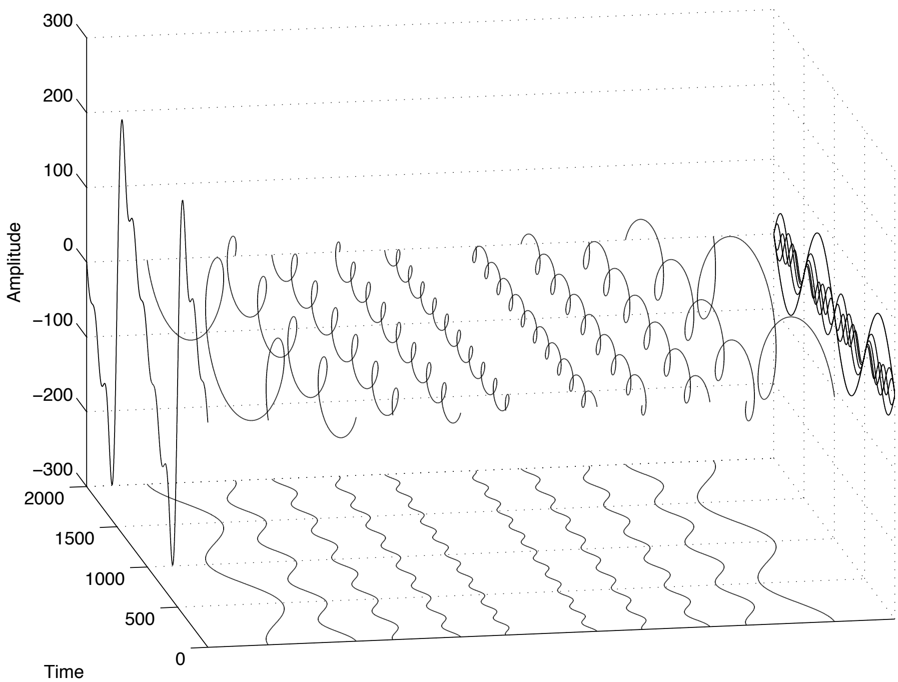

Figure I.1. *Analysis of a real signal into complex exponential harmonics*

The time and frequency domain representations of a signal are not mutually exclusive: the field of time-frequency analysis [FLA 98] is concerned with intermediate representations that combine aspects of time and frequency. The need for intermediate representations arises due to the variation of frequency content in a signal over time. This is not an easy concept to understand, but it follows from the uncertainty principle or *Gabor limit*: a signal cannot be bandlimited (i.e. with frequency content limited to a finite range of frequencies) and simultaneously be of limited time duration. A pure sinusoidal signal with unlimited duration (infinite extent) can be represented in the frequency domain as an impulse (that is a function with zero value everywhere except at one frequency point). Conversely, an impulse in the time domain has infinite bandwidth. However, a signal that contains a specific frequency for a limited time requires a time-frequency representation. Examples of such signals occur widely in the real world: speech and music contain frequencies that are present for a short time (one note played on a musical instrument, for example, which lasts for the duration of the note, plus some reverberation time afterward). An in-depth discussion of these ideas is outside the scope of this book, but is assumed to be understood; although much of the contents of the book relates to Fourier transforms, the quaternion approach can easily be applied to time-frequency concepts, such as fractional and short-time Fourier transforms, by combining quaternion transform formulations with existing knowledge from classical signal processing.

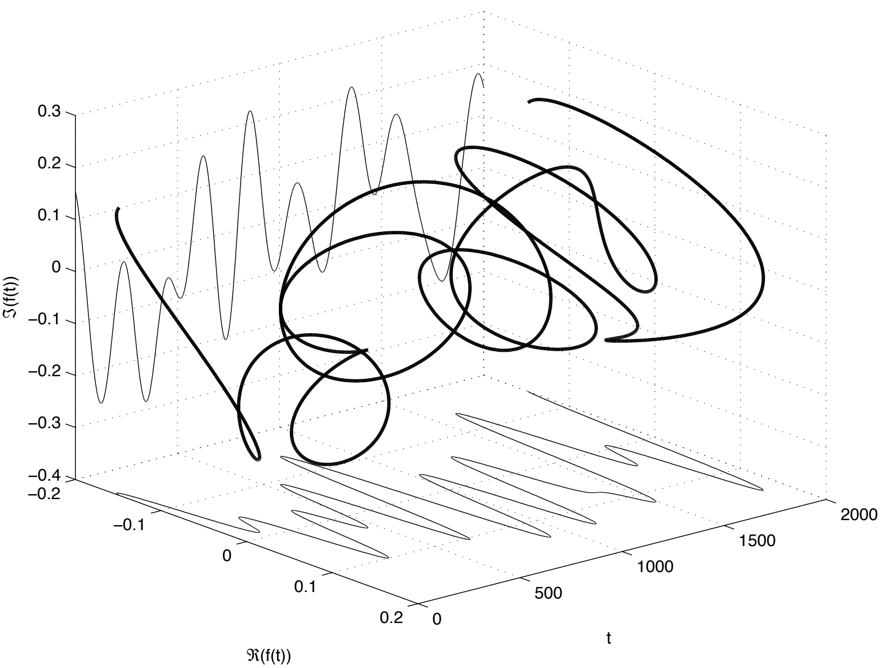

Figure I.2. *A bandlimited complex signal showing real and imaginary parts projected onto the base and rear of the grid box*

I.2. Quaternions

In this book, we are concerned with signals and images that have vector-valued samples (that is samples with three or more components), and their processing using

Fourier transforms based on four-dimensional hypercomplex numbers (quaternions). In Chapter 4 (section 4.3), we show that quaternion Fourier transforms also have applications for the processing of complex signals, exploiting the symmetry properties of a quaternion Fourier transform that are missing from a complex Fourier transform.

A vector-valued signal (in three dimensions, for example) evolves over time and traces out a path in three-dimensional space. To render a plot of such a signal requires four dimensions, and therefore we cannot produce a graphical representation like the one in Figure I.2. Decomposition of a vector-valued signal into harmonic components requires a Fourier transform in an algebra with dimension higher than 2, and this is the motivation for the use of quaternions, which, as we will see, are the next available higher-dimensional algebra after the complex numbers.

Quaternions followed the work of Fourier just over 20 years later, Sir William Rowan Hamilton in 1843 to generalize the complex numbers to three dimensions, was forced to resort to four dimensions in order to obtain what we now call a *normed division algebra*, that is, an algebra where the norm of a product equals the product of the norms, and where every element of the algebra (except zero) has a multiplicative inverse [WAR 97]. Hamilton opened a door in mathematics to hypercomplex algebras in general [STI 10, Chapter 20], [KAN 89], leading to the octonions [CON 03, BAE 02] in less than a year, and the Clifford algebras about 30 years later [LOU 01, POR 95].

I.3. Quaternion Fourier transforms

Quaternion Fourier transforms, the subject of this book, are a generalization of the classical Fourier transform to process signals or images with three- or four-dimensional samples. Such signals arise very naturally in the physical world from the three dimensions of physical space. Quite independently, for very different (physiological) reasons connected with the trichromatic nature of human color vision [MCI 98], color images have three components per pixel. The fourth dimension of the quaternions plays a role in at least two ways: the frequency-domain representation of a signal with three-dimensional samples requires four dimensions (just as in the complex case, two dimensions are required in the frequency domain, even if the original signal has one-dimensional samples). But more generally, the four dimensions of the quaternions can be used to represent a most general set of geometric operations in three dimensions using homogeneous coordinates, which are explored in a later chapter (see section 2.3) and in [SAN 13a]. Of course, generalizations to higher dimensions are possible, and there is a wide range of work on Clifford Fourier transforms, which is outside the scope of this book (we refer the readers to a recent volume for further details [HIT 13], and in particular the historical introduction contained within [BRA 13]).

I.4. Signal and image processing

Fourier transforms are a fundamental tool in signal and image processing. They convert a signal or image from a representation based on sample or pixel amplitudes into a representation based on the amplitudes and phases of sinusoids. The latter representation is said to be in the *frequency domain*, and the original signal is said to be in the *time domain* for a signal which is a time series, or in the *image domain* for an image captured with a camera or scanner. Of course, signals may be encountered that are not time series, for example, measurements of some physical quantity made at (regular) intervals in space; in this book, we will use the terminology of time series for simplicity, since the processing of other signals is mathematically no different.

The Fourier transformation is *invertible*, which means that the original signal or image may be recovered from the frequency domain representation. More interestingly, the frequency domain representation may be modified before inversion of the transform, so that the recovered signal or image is a modified version of the original, for example, with some frequencies or bands of frequencies suppressed, attenuated or amplified. In some applications, inversion of the transform is not needed: the processing performed in the frequency domain directly yields information that can be immediately utilized. An example is computer vision, where a decision based on analysis of an image may result in an action without any need to construct an image from the processed frequency domain representation. At a more detailed level, another example includes correlation, where the signal or image is processed in the frequency domain to yield information about the location of a known object within an image (the same applies in signal processing to find a known signal occurring within a longer, noisy signal).

The classical Fourier transform is inherently based on complex numbers. This is obvious from the fact that the frequency domain representation must represent both the amplitude and the phase of each frequency present in the signal or image. The symmetry of the transform means that the signal may be complex without any modification of the transform. (There are some specialized variants of the Fourier transform that handle only real signals, for example the Hartley transform [BRA 86]). Given a signal with three components (representing, for example, acceleration in three mutually perpendicular directions), how can a frequency domain representation be calculated? The question is very similar if one considers a color image: is it possible to construct a *holistic* frequency-domain representation of the entire image? Obviously one can compute separate classical (i.e. complex) Fourier transforms of the three components in both of these cases, but one then has three separate frequency-domain representations, each representing one aspect of the original image (the frequency content of one of the color or luminance/chrominance channels). Processing of separate representations is sometimes known as *marginal* processing, for reasons connected with techniques in the hand computation of *marginal distributions* in statistics [TRU 53, section 1.22]. It is axiomatic in this book that marginal processing is not the best way to handle signals and images with more than two components per sample, but we will attempt to justify this belief throughout the book, by showing how holistic approaches with quaternions yield better results.

There is another reason for using a quaternion Fourier transform in some applications, and it provided the motivation for the earliest published work on quaternion Fourier transforms (in the field of nuclear magnetic resonance (NMR)). When a two-dimensional signal is captured (that is samples are measured over a two-dimensional grid, like an image), it is sometimes necessary to regard the two dimensions of the sampling grid as independent time-like axes. Processing such a signal with a classical two-dimensional complex transform mixes the two dimensions, whereas a suitably formulated quaternion transform does not. This is because it is possible to associate each of the time-like dimensions with a different dimension of the four-dimensional quaternion space, thus keeping the frequency-domain representations of the first and second time-like axes apart. There were two independent (as far as we are aware) early formulations of quaternion Fourier transforms, by Ernst [ERN 87, section 6.4.2] and Delsuc [DEL 88, equation 20], which are almost equivalent (they differ in the relative placement of the exponentials and the signal, and in the signs, inside the exponentials):

$$F(\omega_1, \omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t_1, t_2) e^{\boldsymbol{i}\omega_1 t_1} e^{\boldsymbol{j}\omega_2 t_2} \mathrm{d}t_1 \mathrm{d}t_2 \qquad [I.1]$$

Note that the two-dimensional signal $f(t_1, t_2)$, is scalar-valued (i.e. not quaternion-valued as in most of the cases discussed in this book). The two time-like axes t_1 and t_2 are treated using separate quaternion roots of -1 ($\boldsymbol{i}$ and $\boldsymbol{j}$), and therefore they are not mixed. This may also be regarded as being due to the orthogonality of the imaginary parts of the two exponentials.

Similar considerations motivated Thomas Bülow [BÜL 99, BÜL 01] when he processed grayscale images using a quaternion Fourier transform. By using a transform with samples of dimension greater than 2, Bülow was able to study symmetries present in certain images in a way that is not possible with the two-dimensional complex Fourier transform.

I.5. Other hypercomplex algebras

Of course, there are alternatives to quaternions for the construction and computation of Fourier transforms for the type of applications suggested so far in this introductory chapter, and it is worth briefly reviewing them here in order to provide a full picture. As already noted, the quaternion algebra is not the only hypercomplex algebra. In fact, the quaternion algebra is a specific example of a more general class of hypercomplex algebras discovered by William Kingdon Clifford in 1876 (about 30 years after Hamilton's discovery of quaternions) and known as *Clifford algebras*. The Clifford algebras include the complex numbers and quaternions, but not the octonions, curiously. However, as already mentioned, the quaternions share with the real numbers and the complex numbers a very specific property that sets them aside from all other Clifford algebras: every non-zero quaternion, $q \neq 0$, has a multiplicative inverse such that $q\,q^{-1} = q^{-1}q = 1$. Furthermore, the quaternion algebra is *normed*. This means that it is possible to define a norm (representing the squared length of the quaternion in four dimensions) such that the norm of a product of two quaternions equals the product of the norms of the two quaternions taken separately: $\|pq\| = \|p\|\,\|q\|$. This is discussed in section 1.2 (see [1.14]). Hypercomplex algebras in general have other troublesome properties. Many contain values that are *idempotent* or *nilpotent*. An idempotent value q squares to give itself: $q^2 = q$; and a nilpotent value squares to give zero: $q^2 = 0$. Such values obviously have the potential to cause problems in numerical algorithms [ALF 07], and the choice of the quaternion algebra avoids these problems entirely, because there are no nilpotent or idempotent quaternions other than 0 and 1, respectively.

The one property of the quaternions that cannot be avoided is that multiplication of quaternions is not commutative. This means that pq gives a different result in general from qp for two arbitrary quaternions p and q. The reason for this can be stated quite simply – the vector (or cross) product in three dimensions is not commutative, and the product of two quaternions contains a vector product. It is important to understand that this is an inherent property of three-dimensional geometry and is not specific to the quaternions. This is again discussed in Chapter 1. Non-commutative multiplication can be avoided by choosing a different hypercomplex algebra, but since any hypercomplex algebra which is commutative contains divisors of zero (a consequence of the Frobenius [DIC 14, section 11]), any attempt to avoid non-commutative multiplication will inevitably lead to other problems, which may well be more troublesome than non-commutativity. Non-commutative multiplication also occurs in linear algebra, of course, where the product of matrices is dependent on ordering; so it should not cause undue concern to anyone contemplating using quaternions.

I.6. Practical application

The ideas and concepts in this book are realisable in practice in several different ways, particularly using software.

I.6.1. *Software libraries*

The library [SAN 13b] permits experimentation with transforms and other algorithms operating on three- and four-dimensional data in MATLAB®. Since

MATLAB® can generate C and C++ code (with some restrictions on supported language features), quaternion code can, in principle, be used to generate code for stand-alone applications, subject to licensing[1].

Alternatively, code can be custom-written, using a quaternion library such as the Boost library for C++, which contains some quaternion functions in the Math toolkit [BRI 13]. This is discussed again in section 3.3, particularly with respect to the use of decompositions into complex transforms to avoid the need to code elaborate algorithms directly in quaternion code.

Both the QFTM and Boost libraries adopt the approach of directly coding quaternion operations, that is they represent quaternions as quadruplets of real (or complex) values, and provide elementary functions to add and multiply quaternions, implementing the famous rules for $\boldsymbol{ijk}$ given in section 1.1, directly in code.

I.6.2. *Matrix representations*

An alternative to the use of quaternion libraries is possible, using matrix representations, which we discuss here, and will return to with a very practical application (to verification) in section 3.3.3.1.

Hypercomplex algebras (with the exception of the octonions [CON 03, BAE 02], which are not associative) have matrix representations. What this means is that for a given algebra, there exists a matrix algebra with real or complex elements that is equivalent to the given hypercomplex algebra, in the sense that multiplication (and addition of course) of the matrix representations is equivalent to multiplication in the hypercomplex algebra. There are also other equivalences, for example the norm of a hypercomplex value may be equivalent to the determinant of the matrix representation. The matrix algebra using the given representation is said to be *isomorphic* to the hypercomplex algebra. Matrix representations for the quaternions are discussed in section 1.4.3, but we discuss the ramifications here.

Given the existence of a matrix representation of quaternions, it is theoretically possible to substitute matrix representations for quaternions, both in algebraic manipulation and in computer coding (the same is true for other hypercomplex algebras except the octonions). Doing so can be a useful technique in theoretical development because it can reduce a hypercomplex problem to a problem involving real (or complex) matrices, and thus provide a deep insight into the relation between the hypercomplex case and the well understood real and complex cases. However, there are disadvantages of using a matrix representation compared to a direct quaternion approach, *in practice*:

1 QTFM is licensed under the GNU General Public License.

1) Computation with matrices is numerically inefficient, and it requires four times as much memory as a direct quaternion representation storing only four values. A quaternion product requires 16 multiplications and 12 additions, whereas the equivalent 4×4 matrix product requires 64 multiplications and 48 additions, an increase by a factor of 4. This disadvantage effectively rules out the use of matrix representations for implementation, even for computer simulations (as run-time is four times less using a quaternion library that directly calculates quaternion products than using a general matrix package and a matrix representation for each quaternion).

2) The result of a sequence of arithmetic operations *in matrix form* may not be an accurate representation of a quaternion matrix. This disadvantage again rules out the use of matrix representations for implementation (a quaternion library will yield more accurate results).

3) The matrix representation provides little geometric insight. As will be shown in Chapter 2, the quaternion algebra provides a remarkably intuitive link with the geometry of three or four dimensions. It is possible to manipulate quaternion symbols algebraically in order to derive expressions for geometric operations. This is the central idea in *geometric algebras* [SOM 01]. Geometric algebras, as might be expected from the preceding text, are not division algebras; so there is a price to be paid for their additional geometric utility in the form of divisors of zero, which make them less attractive for applications in digital signal and image processing.

The matrix representation is certainly useful, and it is helpful in any quaternion library to have the ability to convert between a direct quaternion representation and the matrix representation. In the QTFM library [SAN 13b], for example, functions called *adjoint*[2] and *unadjoint* are provided to perform the conversion, even for matrices of quaternions (the adjoint in the latter case is a block matrix with each block representing one quaternion).

I.7. Overview of the remaining chapters

The rest of the book is divided into four chapters. Chapter 1 covers the quaternion algebra, and provides the mathematical definitions and concepts necessary for the later chapters. Chapter 2 presents the geometric applications of quaternions, and provides the ideas necessary to understand how quaternions can be used to represent both three- and four-dimensional values and geometric operations applied to them. Chapter 3 gives a detailed and comprehensive account of quaternion Fourier transforms, including their definitions, operator formulas and how they may be computed. Chapter 4 shows how quaternion Fourier transforms can be applied in signal and image processing.

2 The “adjoint” terminology was taken from Zhang’s 1997 paper [ZHA 97].