Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

RESEARCH ARTICLE

Reconstructing a Rotor from Initial and Final Frames using Characteristic Multivectors: with applications in Orthogonal Transformations

Anthony Lasenby[1] | Joan Lasenby[2] | Charalampos Matsantonis[2]

[1]Kavli Institute of Cosmology, Department of Physics, University of Cambridge, UK
[2]Signal Processing & Communications Lab, Engineering Department, University of Cambridge, UK

Correspondence
Joan Lasenby, Signal Processing & Communications Lab, CUED, Trumpington Street, Cambridge, UK. Email: jl221@cam.ac.uk

Abstract

If an initial frame of vectors $\{e_i\}$ is related to a final frame of vectors $\{f_i\}$ by, in geometric algebra (GA) terms, a *rotor*, or in linear algebra terms, an *orthogonal transformation*, we often want to find this rotor given the initial and final sets of vectors. One very common example is finding a rotor or 4×4 orthogonal matrix representing rotation and translation, given knowledge of initial and transformed points. In this paper we discuss methods in the literature for recovering such rotors and then outline a GA method which generalises to cases of any signature and any dimension, and which is not restricted to orthonormal sets of vectors. The proof of this technique is both concise and elegant and uses the concept of *characteristic multivectors* as discussed in the book by Hestenes & Sobczyk, which contains a treatment of linear algebra using geometric algebra.

Expressing orthogonal transformations as rotors, enables us to create *fractional transformations* and we discuss this for some classic transforms. In real applications, our initial and/or final sets of vectors will be noisy. We show how to use the characteristic multivector method to find a 'best fit' rotor between these sets and compare our results with other methods.

KEYWORDS:
geometric algebra; characteristic multivectors; frame transformations; orthogonal transformations

1 | INTRODUCTION

In a Geometric Algebra approach, orthogonal transformations are carried out via rotors, R, which act two-sidedly on general objects, M, within the algebra via $M \mapsto RM\tilde{R}$. Often we want to be able to determine R given initial and final information about a frame of vectors which has been subject to the transformation. Specifically, given an initial frame of vectors $\{e_i\}$ and a final frame $\{f_i\}$ that we know to be related via $f_i = Re_i\tilde{R}$, we would like to recover the rotor R. Results for this in 3D Euclidean space and 4D spacetime have been known about for some years (Hestenes & Sobcyzk[1],Doran & Lasenby[2]), but recently Shirokov[3] has given a general result which in principle can be used to find R from a knowledge of $\{e_i\}$ and $\{f_i\}$ in any dimension and any signature of metric. This is however, subject to some caveats: firstly, as presented, it assumes the $\{e_i\}$ form a standard orthonormal set. In fact, it should be possible to recover R for any starting set of vectors, as long as they span the complete space. Secondly, the proof given by Shirikov[3] is several pages long and quite detailed, relying on aspects of matrix algebra that are somewhat extraneous to a 'pure GA' approach. In this paper we will present a GA proof of the main result which can be carried out in just a few lines, and also immediately generalises to non-orthonormal frames of vectors. For these reasons, it is useful to re-examine the important result in Shirikov[3] from a pure GA point of view, in order to provide a much shorter proof of a more general result. Furthermore, in the process of doing this, we will find some very interesting connections with some topics in a Geometric Algebra approach to linear algebra which have been pioneered and drawn attention to by David Hestenes[1], but which have gained little visibility as yet, despite being potentially very important. This concerns the subject of *characteristic multivectors* and their associated use in a GA version of the *Cayley-Hamilton theorem*. We shall see that our reformulation of the result in Shirikov[3] brings in both of these elements in interesting ways, pointing towards yet further generalisation.

Given that all Lie Groups can be represented in a rotor formulation Doran *et al*[4], and that the concept of an 'initial state' being mapped to a 'final state' is ubiquitous in both classical and quantum physics, as well as in engineering, the range of application of formulas such as we discuss here is potentially very wide. However, in this paper, we illustrate the results we have found in just two example applications, which are the notion of *orthogonal transforms in signal processing*, where we will look at a rotor formulation of the Haar transform and the 4 × 4 discrete cosine transform (DCT), and the concept of finding the closest orthogonal transform to a given non-orthogonal transform. As well as achieving this aim, of finding an underlying rotor structure for the transform, the approach taken here allows novel concepts such as 'fractional' transforms, or indeed continuous interpolation between classic transform states. Such an approach would obviously generalise to other families of orthogonal wavelet transforms as well, again raising interesting possibilities for future work.

This paper will be structured as follows. Section 2 will describe the concept of *characteristic multivectors* as given in Hestenes & Sobczyk[1] and their relation to the *characteristic polynomial and the Cayley Hamilton theorem.* Section 3 will then give the result and proof of how we construct a rotor relating a frame and its transform from these characteristic multivectors. We will start by assuming orthogonal frames and then generalise to non-orthogonal frames. Section 4 will consider comparisons between this method and other results in the literature. Section 5 will look at applications of our method to classical orthogonal transforms used ubiquitously in image processing and at how it also provides a method for straightforwardly obtaining the 'closest' orthogonal transform to a transform that is assumed to have been formed by noise added to an underlying orthogonal transform.

2 | CHARACTERISTIC MULTIVECTORS

The essential objects here are the *simplicial derivatives*. We start by defining the simplicial derivative as described in Hestenes & Sobczyk[1]: given a set of linearly independent vectors $v_1, v_2, ..., v_p$ we define a *simplicial variable* $v_{(p)} = v_1 \wedge v_2 \wedge, \ldots \wedge v_p$ and a *simplicial derivative* $\partial_{(p)}$ relative to this simplicial variable as:

$$\partial_{(p)} = \frac{1}{p!}(\partial_{v_p} \wedge \ldots \wedge \partial_{v_1}) \tag{1}$$

Now, if f is a vector-valued linear function of a vector a living in an m-dimensional space, V_m, and the output $f(a)$ lives in the same space (the simplest case) then we can define the rth simplicial derivative of f as follows.

We let $\{a_k\},\ k = 1, \ldots, m$ be a *frame* for the space and $\{a^k\}$ its reciprocal frame; we also define $b_k = f(a_k)$ for $= 1, \ldots, m$. We then look at sets of simplicial variables, $a_{(r)} = a_{j_1} \wedge \ldots \wedge a_{j_r}$, where $0 < j_1 < \ldots < j_r \leq m$. The rth *simplicial derivative* of $f_{(r)} = f(a_{j_1}) \wedge f(a_{j_2}) \wedge \ldots \wedge f(a_{j_r})$, is then

$$\partial_{(r)} f_{(r)} = \sum (\partial_{a_{j_r}} \wedge \ldots \wedge \partial_{a_{j_1}})(f(a_{j_1}) \wedge \ldots \wedge f(a_{j_r})) \tag{2}$$

Note we no longer include the $1/r!$ factor. But as $a^k = (a^k \cdot a_k)\partial_{a_k} = \partial_{a_k}$ (no sum over repeated indices), we can write this as:

$$\partial_{(r)} f_{(r)} = \sum (a^{j_r} \wedge \ldots \wedge a^{j_1})(b_{j_1} \wedge \ldots \wedge b_{j_r}) \tag{3}$$

where the sum is over all sets of r indices such that $0 < j_1 < \ldots < j_r \leq m$. The point about these multivector quantities is that they provide invariant information about the function f. The invariance is in the sense that any frame $\{a_k\}$ could be chosen, and we would still get the same objects – they are therefore in some sense 'intrinsic' to the space V_m and the function f.

2.1 | The Characteristic Polynomial and the Cayley-Hamilton Theorem

We can now employ the results of the previous section to look at the *characteristic polynomial* and *Cayley- Hamilton theorem*. These use just the scalar parts of the various simplicial derivatives. As shown in Hestenes & Sobczyk[1] Section 3-2, the characteristic polynomial $C_f(\lambda)$ of a linear function f is given by:

$$C_f(\lambda) = \sum_{s=0}^{m} (-\lambda)^{m-s} \partial_{(s)} * f_{(s)} \tag{4}$$

where $*$ means '*take the scalar part of the geometric product*' and $\partial_{(0)} * f_{(0)}$ is taken as 1. λ is the scalar argument of the polynomial function. If λ is an eigenvalue of f, i.e. $f(a) = \lambda a$, then it is a root of the characteristic polynomial, i.e. we will have $C_f(\lambda) = 0$.

The Cayley-Hamilton theorem states that a linear function satisfies its own characteristic equation, which then tells us that

$$\sum_{s=0}^{m} (-\lambda)^{m-s} \partial_{(s)} * f_{(s)} f^{m-s}(a) = 0 \tag{5}$$

for any input vector a. $f^{(r)}$ is the r-fold application of f and $f^{(0)}(a)$ is interpreted as a.

2.2 | The basic result for reconstructing a rotor

We now let f be a rotor, i.e. $f(a) = Ra\tilde{R}$ where $R\tilde{R} = 1$. Let us suppose that the set $\{a_i\}$, $i = 1, \ldots n$, form an n-dimensional frame, and $b_i = f(a_i) = Ra_i\tilde{R}$ are what the a_i's are mapped to under f.

Then our basic claim is that $\tilde{R}$ is a scalar multiple of the sum of the characteristic multivectors of f, i.e.

$$\tilde{R} = \alpha \sum_{r=0}^{n} \partial_{(r)} f_{(r)} \tag{6}$$

where α is a real scalar, and the $\partial_{(r)} f_{(r)}$ are given in terms of the 'input' and 'output' frames by equation (3). This holds in *any dimension and signature* provided there are no null basis vectors.

2.3 | Proof

We start with the case where the frames are orthonormal, and write $a_i = e_i$, $i = 1, \ldots, n$. The sum on the r.h.s. of (6) is then (again with the sum over the repeated indices restricted by $0 < j_1 < \ldots < j_r \leq m$)

$$\begin{aligned}
\sum_{r=0}^{n} \sum \left(a^{j_r} \wedge \ldots \wedge a^{j_1}\right)\left(b_{j_1} \wedge \ldots \wedge b_{j_r}\right) &= \sum_{r=0}^{n} \sum \left(e^{j_r} \ldots e^{j_2} e^{j_1}\right)\left(Re_{j_1}\tilde{R}Re_{j_2}\tilde{R} \ldots Re_{j_r}\tilde{R}\right) \\
&= \sum_{r=0}^{n} \sum e^{j_r} \ldots e^{j_2} e^{j_1} Re_{j_1} e_{j_2} \ldots e_{j_r} \tilde{R} \\
&= \left(\sum_{r=0}^{n} \sum e^{j_r} \ldots e^{j_2} e^{j_1} Re_{j_1} e_{j_2} \ldots e_{j_r} \right) \tilde{R}
\end{aligned} \tag{7}$$

We will thus have proved the result in (6) if we can show that the double sum within the brackets in the last line of (7) is just a scalar. If we divide the overall 2^n dimensional GA space into vectors, a, bivectors B, trivectors T, quadrivectors Q, and so on up to Q_n say, where n is the dimension of the space, then it is easy to see this quantity is

$$R + \partial_a Ra + \partial_B RB + \partial_T RT + \partial_Q RQ + \ldots + \partial_{Q_n} RQ_n \tag{8}$$

We see that equation 8 is true from the definition of the differentiation wrt to an r-vector, $X = X^K e_K$, where the $\{e_K\}$ are the basis r-vectors:

$$\partial_X RX = e^J \frac{\partial}{\partial X^J} RX^K e_K = e^J Re_J$$

which is precisely what the inner sum in (7) gives us. At this stage it is clear that the rotor nature of R is irrelevant, and we in fact seek to show that

$$M + \partial_a M a + \partial_B M B + \partial_T M T + \partial_Q M Q + \ldots + \partial_{Q_n} M Q_n \tag{9}$$

is a scalar for a general even element M, which will then make the LHS of equation 7 proportional to $\tilde{R}$. We can make this easier by recognising that what we are now doing is forming a derivative with respect to all 2^n independent elements in the space. In particular, if we write this set of 2^n elements symbolically as E_j, $j = 1, \ldots, N$, where $N = 2^n$ is the dimension of the entire space, then what we want to show is that, in the usual multivector derivative notation,

$$\partial_E M E = \text{scalar} \tag{10}$$

for any even element M, and E is the set of all $\{E_j\}$. This means that we want $\partial_E \langle M \rangle_r E = 0$ for each even grade $r > 0$ in the space (as this is a reflection which preserves grade). This is easy to establish as follows. Rewriting

$$\partial_E M E = \sum_{j=1}^{2^n} E^j M E_j \tag{11}$$

we can see that the result we want would follow if each even basis element of the space, apart from the scalar identity, commutes with $N/2$ elements and anticommutes with $N/2$. In this case the sandwiching above would result in everything cancelling except scalar quantites. This result is not difficult to show for M even, but it is in fact more general and holds for any element of the space, not just even elements. As this result could be useful in other contexts, the proof for any element of the space (not just even elements) is given in Appendix A.

The overall result is therefore:

$$\partial_E M E = \sum_{j=1}^{2^n} E^j M E_j = 2^n \langle M \rangle \tag{12}$$

2.4 | An important special case

We can see that an important special case will arise, and the above method will fail to return a rotor, if it happens that the rotor we are trying to recover has zero scalar part, since then the sum of characteristic bivectors will just return 0. There is nothing wrong with the above mathematics in this case, it is just that the factor in front on $\tilde{R}$ in the last line of equation (7), which we now know to equal $2^n \langle R \rangle$, will be 0, and hence we can't invert it to recover $\tilde{R}$.

As an example of a rotor with zero scalar part, we can consider 180° spatial rotations. E.g., in Euclidean space, the rotor $R = e_1 e_2$ rotates both the e_1 and e_2 axes by 180°, whilst leaving the other axes alone, and clearly has no scalar part.

It is also possible to have rotors with neither a scalar nor bivector part. E.g., in Euclidean 4d space the pseudoscalar $I = e_1 e_2 e_3 e_4$ is a rotor, since it satisfies $I\tilde{I} = 1$. This quantity rotates all four axes by 180°.

If we have a rotor R with no scalar part, but possessing a non-zero bivector part, then by multiplying it with each basis bivector B_i in turn, then we must certainly at some point reach a (combined) rotor RB_i with a non-zero scalar part. Of course, if we only have available information about the initial and final frames, and not the rotor itself, then we cannot explicitly form RB_i. However, we can simulate the effect that the B_i would have on the final frame by noting that if $B_i = e_j e_k$ say, has negative square, then it will flip the j^{th} and k^{th} axes by 180°, and if it has positive square if will flip all the axes *except* the j^{th} and k^{th} ones by 180°.

This then provides us with an algorithm to deal with the case where we take the sum of characteristic bivectors, as in the l.h.s. of equation (7), but find this gives 0. We now form alternative sums, where the signs of the b_j's are flipped in accordance with the object we are conceptually multiplying the desired rotor R on the right by. As described, there will be two flips if the object is a negative square bivector, and $n-2$ flips for a positive square one. For each such object tried we can ask if we then get a rotor S for which there is a scalar part. If we do, the process terminates, and if the bivector basis element concerned was B_i say, we can form the desired R from S by multiplying on the right by B_i^{-1}, since $S = B_i R$.

If we have gone through the entire bivector basis and still not found an S with non-zero scalar part, then we go to the grade 4 basis elements that could be rotors, and work through these, flipping the signs of the b_j's according to the effects that sandwiching in this grade 4 element and its reverse would have on the initial $\{e_i\}$ set of basis vectors. If *this* fails, then we go to the grade 6 basis elements which could be rotors, and so on.

Unless R is actually 0, then this process must terminate at some point, and we will have succeeded in recovering R.

2.5 | Extension to non-orthogonal frames

In Section 2.2 we made the statement that equations 6 and 7 (evaluating the simplicial derivatives from the 'input' and 'output' frames as given in equation 3) held for any start and end set of frames and in any signature and dimension – however, in Section 2.3 we gave a proof using orthogonal frames. We now note that there is nothing in the proof that specified the signature of the space (provided no basis elements square to zero), and also that it was for any dimension. Additionally, the important point about the characteristic multivectors of a transformation is that provided the frame $\{a_k\}$ spans the space, the characteristic multivectors themselves are independent of the choice of frame, and depend just on the transformation f and the space. This means the proof goes through unchanged for non-orthogonal frames.

3 | COMPARISON WITH EXISTING RESULTS

This section will look at some existing results for recovering rotors from initial and final frames, in particular, the results in Shirikov[3].

3.1 | Rotors in Euclidean 3D

If we have a frame (not necessarily orthogonal) $\{e_i\}$ in 3D Euclidean space which is rotated by a rotor R to a frame $\{f_i\}$ (so that $f_i = Re_i\tilde{R}$), it is well known that we recover the rotor via the following simple expression:

$$R = \beta(1 + f_i e^i) \tag{13}$$

where the constant β ensures that $R\tilde{R} = 1$. This is undoubtedly the simplest form for recovering the required rotation. This result first appeared in Hestenes & Sobczyk[1] and can also be found in Doran & Lasenby[2]. If we were now to use our characteristic multivector formula for recovering the rotor, we have:

$$\tilde{R} \propto 1 + e^i f_i + \left(e^k \wedge e^j\right)\left(f_j \wedge f_k\right) + \left(e^3 \wedge e^2 \wedge e^1\right)\left(f_1 \wedge f_2 \wedge f_3\right) \tag{14}$$

where $i, j, k = 1, 2, 3 \quad j \neq k$. Initially this does not look like equation 13, but in fact, it can be shown that the *bivector-bivector* term is a multiple of the *vector-vector* term and the *trivector-trivector* term is a constant (obvious as both factors are proportional to the pseudoscalar), and the multiples/constants are such that we obtain $\tilde{R} \propto 1 + e^i f_i$, which gives the standard result on taking the reverse.

3.2 | Rotors in 4D Spacetime

Now, again consider a spacetime rotor R such that $f_i = Re_i\tilde{R}$, $i = 1, .., 4$ and the signature of the space is $(+, -, -, -)$. It is again well known (see Hestenes & Sobczyk[1],Doran & Lasenby[2]) that the rotor can be recovered by the remarkably simply formula:

$$R \propto f_k e^k \tag{15}$$

For our characteristic multivector formula to work we need the *bivector-bivector*, *trivector-trivector* and *4-vector - 4-vector* terms to jointly cancel out the scalar term and give a multiple of the *vector-vector* term. This is indeed exactly what happens – depending on the nature of the rotation, the *bivector-bivector* and *trivector-trivector* parts are multiples of the *vector-vector* part or multiples of the bivector part of the *vector-vector* part, while the *4-vector - 4-vector* part gives a scalar.

It is interesting to note that the above relations between terms in our characteristic multivector expression can often be related via the *characteristic polynomial*. This will be discussed further elsewhere.

3.3 | Formula in Hestenes & Sobcyzk[1] for orthogonal transformations

Chapter 3 in Hestenes & Sobczyk[1] describes how *Linear Algebra* is dealt with in GA. Here we see that if two *orthonormal* frames $\{e_k\}$ and $\{f_k\}$ are related by a rotor R, where $k = 1, 2, ..., n$ ($f_k = Re_k\tilde{R}$) and the space is Euclidean, we can write:

$$R = R_n \dots R_2 R_1 \tag{16}$$

Here each R_k is a rotation in an elemental plane – so the overall rotor is made up of *rotations in orthogonal planes*. These planes can be shown to be the planes formed by the eigen-bivectors (which can also be formed from the complex eigenvectors in a standard matrix decomposition). In a later section we will look specifically at orthogonal transformations and show how the characteristic multivector approach compares to this plane-wise decomposition. We also note here that the recent paper by Roelfs & de Keninck[5] shows how we can decompose either a rotor R or a bivector B into mutually commuting simple rotors or simple bivectors respectively.

3.4 | Shirikov's formulation

Sirikov[3] gives the main result in his equation (6.1). This is that if two orthonormal frames, $\{e_a\}$ and $\{\beta_a\}$, are related by a rotor S, i.e.

$$Se_a\tilde{S} = \beta_a \tag{17}$$

where $a = 1, \dots, n$, then if

$$M = 1 + \beta_a e^a + \dots + \beta_{1\dots n} e^{1\dots n} \tag{18}$$

is non-zero, then

$$S = \pm\frac{M}{\sqrt{\tilde{M}M}} \tag{19}$$

Here the notation $\beta_{1\dots k}$, $k \leq n$, is explained in his equation (2.4), which has the definition

$$\beta_{a_1\dots a_k} = \beta_{a_1}\cdots\beta_{a_k}, \quad 1 \leq a_1 < \dots < a_k \leq n \tag{20}$$

We should note carefully that in[3] the notation for upper multi-indices is such that it is assumed that $e^A = (e_A)^{-1}$, which therefore gives $e^{a_1\cdots a_k} = (e_{a_1\dots a_k})^{-1} = e^{a_k}\cdots e^{a_1}$ and not $e^{a_1}\dots e^{a_k}$, (which might be assumed from the way the lumped downstairs indices work), because $e^{a_i} = (e_{a_i})^{-1}$. We thank Dmitri Shirokov for this clarification concerning the notation.

This formula is therefore the same as our formula in equation 6 for the orthogonal case. As the result is written entirely in terms of orthogonal frames, it is difficult to extend it to non-orthogonal frames, but does hold for arbitrary signature, provided no basis vectors square to 0, and arbitrary dimension. Note that we have also given the extension to cases where $M = 0$.

4 | APPLICATIONS TO ORTHOGONAL TRANSFORMS

In this section we will look at how expressing some conventionally very important orthogonal transforms as rotors; this will enable us to create some novel linear mappings, such as *fractional transformations*.

As examples, we will look at the 2 × 2 Haar Transform and the 4 × 4 Discrete Cosine Transform [DCT]. Both have been extremely important over the years in *image processing and image coding* - one reason for their importance is because they are *energy preserving* transforms.

Recall, a **real** $n \times n$ matrix Q is orthogonal/orthonormal if its rows/columns are orthonormal vectors, so that

$$QQ^T = Q^TQ = \mathrm{I}, \quad \Rightarrow \quad Q^T = Q^{-1}$$

where I is the $n \times n$ identity matrix and we are working in an nD Euclidean space. Let us look at orthogonal transforms in two ways

First: Let an orthogonal basis in this space be $\{e_i\},\ i = 1, .., n$. The matrix Q will take this set of n basis vectors to a set of n orthogonal vectors $\{f_i\}$, where the $\{f_i\}$ are the columns of Q.

$$Qe_i = f_i\ , \qquad f_i \cdot f_j = \delta_{ij}$$

If we are given $\{e_i\}$ and $\{f_i\}$, we would like to recover the rotor, R which takes $\{e_i\}$ to $\{f_i\}$ [assuming one exists].

Second: For an $n \times n$ matrix, consider each pixel as a dimension. We then act two-sidely (can think of this as acting on columns and rows sequentially) on an $n \times n$ array of pixels, call this X, to produce a transformed block of pixels Y via:

$$Y = QXQ^T$$

(recall, we noted that this is an energy preserving transform: $y^T y = x^T x$). We can standardly write X as a linear combination of basis arrays:

$$X = \sum Y_{ij} M_{ij}$$

where $M_{ij} = q_i^T q_j$ is an $n \times n$ array, and where q_i is the ith row of Q.

- Now take M_{ij} and unwrap row by row to form a vector – call this m_k, with $(i, j) = (1, 1) \Rightarrow k = 1$; $(i, j) = (1, 2) \Rightarrow k = 2$; ....., $(i, j) = (n, n) \Rightarrow k = n^2$.

  It is not hard to show that the set $\{m_k\}$ is an orthogonal set if the matrix Q is orthogonal.

- We then look for the rotor which takes the set of pixels, each one being a basis vector, $\{e_k\}$, $k = 1, ..., n^2$, to the orthogonal set of vectors $\{m_k\}$, $k = 1, ..., n^2$.

- In most image processing/coding applications, we learn a lot about the effects of our transforms by decomposing into basis functions, so this approach may well yield something interesting.

We will now look at some explicit examples.

4.1 | Example 1: The 2x2 Haar Transform/Haar Wavelet

The *Haar transform* has, over the years, been an important tool in image processing. It is also the simplest form of *wavelet* that displays desirable characteristics. The form of the 2×2 Haar transform, T, most commonly used is:

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{21}$$

Since

$$Tx = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} x_1 + x_2 \\ x_1 - x_2 \end{bmatrix}$$

we can see that the Haar transform has *lowpass* and *highpass* components.

Recall we form the familiar *Haar basis functions* by taking $t_i^T t_j$, so that

$$M_{11} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad M_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$
$$M_{21} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad M_{22} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

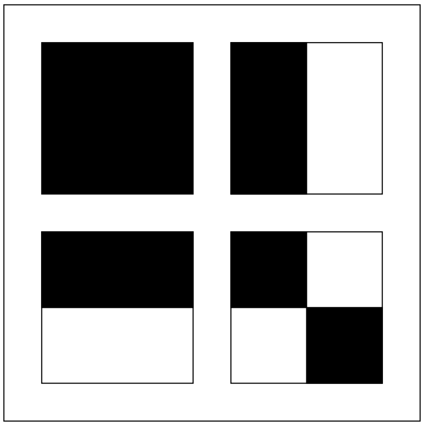

FIGURE 1 The high- and low-pass basis functions of the 2×2 Haar transform

These are illlustrated in figure 1 below:

We now look for the 4D rotor R which performs the following mappings (unwrapping row by row) :

$$\begin{aligned} e_1 &\longrightarrow \frac{1}{2}[e_1+e_2+e_3+e_4] & e_2 &\longrightarrow \frac{1}{2}[e_1-e_2+e_3-e_4] \\ e_3 &\longrightarrow \frac{1}{2}[e_1+e_2-e_3-e_4] & e_4 &\longrightarrow \frac{1}{2}[e_1-e_2-e_3+e_4] \end{aligned}$$

[note, we could choose to do this unwrapping other ways, for example, column by column].

Using the characteristic multivector formula in 4D (noting that orthogonality implies $e^i = e_i$, we have the following expression for the reverse rotor:

$$\tilde{R} \propto 1 + [e_1 f_1 + e_2 f_2 ....] + [(e_2 \wedge e_1)(f_1 \wedge f_2) + ...] +$$
$$[(e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3) + ...] + [(e_4 \wedge e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3 \wedge e_4)]$$

$$\Rightarrow \quad R = \frac{1}{\sqrt{2}}\left[e_{12} + e_{24} - e_{13} - e_{34} + e_{23}\right]$$

Note 1: if you apply the formula in equation 7, you find $\langle R \rangle_0 = 0$: to get around this we can apply a simple rotor to the f_is, work out the new rotor and unwrap the simple rotor, as described earlier.

Note 2: if you unwrap the basis function column by column, you get a different rotor, which is related to the above by a rotor that simply permutes (and in some cases negates) the f_is.

4.2 | Fractional Transforms

Having expressed the Haar transform as a rotor, we can now investigate what *fractional transforms* might look like. The prescription of generating the fractional rotor is simply to extract the *bivector*, B, from the rotor, R, and then to create a rotor R_α from αB, where $\alpha \in [0, 1]$. In order to extract the bivector we first choose how we wish to express our rotor as a function of the bivector, e.g. exponential, Cayley, outer exponential. Below we give the prescription and examples using the Cayley form as the inversion is easy to illustrate, but it is likely that the exponential form may produce more interesting fractional transforms.

- Write the extracted rotor R as a function of B using the Cayley transform and invert to give B in terms of R:

$$R = \frac{1-B}{1+B} \qquad B = \frac{1-R}{1+R}$$

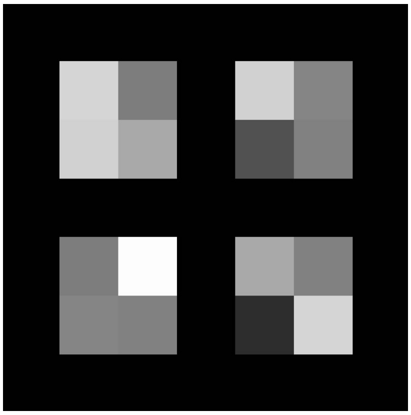

FIGURE 2 The basis functions for the $\alpha = 0.5$ fractional Haar transform

- Create a new rotor using αB, $\alpha \in [0, 1]$:

$$R_\alpha = \frac{1 - \alpha B}{1 + \alpha B}$$

We can now look at the properties of these new transforms generated by the R_α.

Example 1: As a first example, look at $\alpha = 0.5$ in the 2×2 Haar transform. Taking $f_i = R_\alpha e_i \tilde{R}_\alpha$ and forming the orthogonal matrix, F from the f_is, gives (where we are unwrapping the pixels to form vectors, i.e. *Method 2*):

$$F = \begin{bmatrix} a & b+c & b-c & b \\ b-c & d & b+2c & -b+c \\ b+c & b-2c & d & -b-c \\ b & -b-c & -b+c & a \end{bmatrix} \tag{22}$$

with $a = 0.68,\ b = 0.32,\ \ c = 0.24\sqrt{2},\ \ d = 0.04$. If this F now acts on a 2×2 block of pixels that we unwrap to give a vector $x = [x_1, x_2, x_3, x_4]$, and we rewrite the resulting vector $y = Fx$ as a 2×2 block of pixels, we have the following result:

$$Fx = x_1 \begin{bmatrix} \alpha & \beta - \gamma \\ \beta + \gamma & \beta \end{bmatrix} + x_2 \begin{bmatrix} \beta + \gamma & \delta \\ \beta - 2\gamma & -(\beta + \gamma) \end{bmatrix} + x_3 \begin{bmatrix} \beta - \gamma & \beta + 2\gamma \\ \delta & -\beta + \gamma \end{bmatrix} + x_4 \begin{bmatrix} \beta & -\beta + \gamma \\ -(\beta + \gamma) & \alpha \end{bmatrix} \tag{23}$$

These new basis elements are shown in figure 2 .

Example 2: As our next example we look at the 4x4 Discrete Cosine Transform (DCT)

The 4×4 DCT is the basis of the compression standard JPEG XR. The transform matrix, T_{DCT}, is standardly given by:

$$T_{DCT} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{\sqrt{(2)}}{2}\cos(\frac{\pi}{8}) & \frac{\sqrt{(2)}}{2}\cos(\frac{3\pi}{8}) & -\frac{\sqrt{(2)}}{2}\cos(\frac{3\pi}{8}) & -\frac{\sqrt{(2)}}{2}\cos(\frac{\pi}{8}) \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{\sqrt{(2)}}{2}\cos(\frac{3\pi}{8}) & -\frac{\sqrt{(2)}}{2}\cos(\frac{\pi}{8}) & \frac{\sqrt{(2)}}{2}\cos(\frac{\pi}{8}) & -\frac{\sqrt{(2)}}{2}\cos(\frac{3\pi}{8}) \end{bmatrix} \tag{24}$$

The basis functions/arrays of T_{DCT} are shown in the left image of figure 3 – we have 16 4×4 basis arrays, which show increasing horizontal and vertical frequencies as we move from the top left to the bottom right. The application of the 4×4 DCT to the *Lenna* image is shown in the right image of figure 3 (reordered to view low-low frequency components at the top left and high-high frequency components at the bottom right).

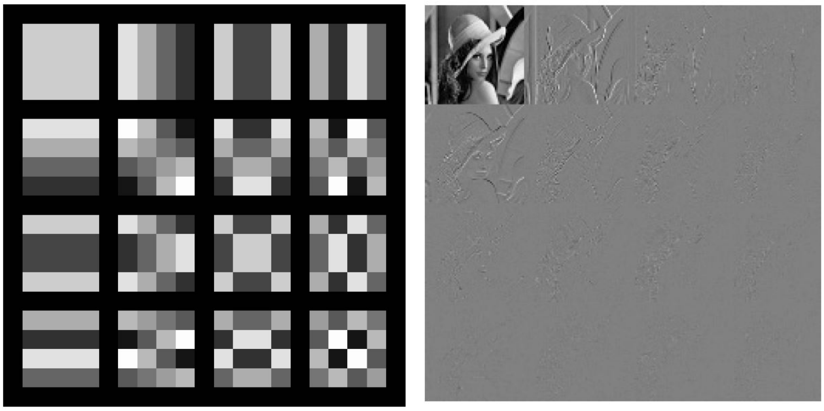

FIGURE 3 **Left**: The 16 basis functions of the 4 × 4 DCT. **Right**: The 4 × 4 DCT applied to the *Lenna* image.

Here we will use *Method 1* to look for the rotor which takes an orthogonal 4D Euclidean frame $\{e_i\}$ to $\{f_i\}$, where the $\{f_i\}$ are the columns of the DCT. To do this we will use the method of *characteristic multivectors* to give:

$$\begin{gathered}\tilde{R} \propto 1 + [e_1 f_1 + e_2 f_2 ....] + [(e_2 \wedge e_1)(f_1 \wedge f_2) + ...] + \\ [(e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3) + ...] + [(e_4 \wedge e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3 \wedge e_4)] \\ \equiv 1 + Inv1 + Inv2 + Inv3 + Inv4\end{gathered}$$

The numerical values are given below – they work out nicely since $\langle R \rangle_0 \neq 0$ so no small extra rotor applied to the $\{f_i\}$ is needed:

$$\begin{gathered}\tilde{R} \propto 1 + [-0.153(e_1 e_2) + 0.229(e_1 e_4) + 0.229(e_2 e_3) - 0.153(e_3 e_4)] + \\ [0.0761(e_1 e_2 e_3 e_4) - 0.0761(e_1 e_4) - 0.383(e_3 e_4) - 0.383(e_2 e_4) + \\ 0.383(e_1 e_2) - 0.383(e_1 e_3) + 0.076(e_2 e_3) - 1.924] + \\ [-0.153(e_1 e_2) + 0.229(e_1 e_4) + 0.229(e_2 e_3) - 0.153(e_3 e_4)] + 1\end{gathered}$$

which simplifies (when normalised) to:

$$\begin{gathered}\tilde{R} = 0.069 - 0.069(e_1 e_2) - 0.347(e_1 e_4) - 0.485(e_2 e_3) + \\ 0.625(e_3 e_4) + 0.347(e_2 e_4) + 0.347(e_1 e_3) + 0.069(e_1 e_2 e_3 e_4)\end{gathered}$$

It is then easy to show (which must follow as the *characteristic multivectors* are invariants) that

$$R(Invk)\tilde{R} = Invk, \quad k = 1, 2, 3, 4$$

The interesting invariant is $Inv1$ ($\equiv Inv3$):

$$-0.153[(e_1 e_2) + (e_3 e_4)] + 0.229[(e_1 e_4) + (e_2 e_3)]$$

which is clearly an eigen-bivector but not an eigenblade. As noted in Section 3.3, orthogonal transformations can be written as a composition of rotors which represent rotations in orthogonal planes, Hestenes & Sobczyk[1] also showed how these relate to the *complex eigenvalues/vectors* of the matrix.

For the 4 × 4 DCT we have 4 complex eigenvalues and 4 complex eigenvectors. Take the two which are not just congjugates:

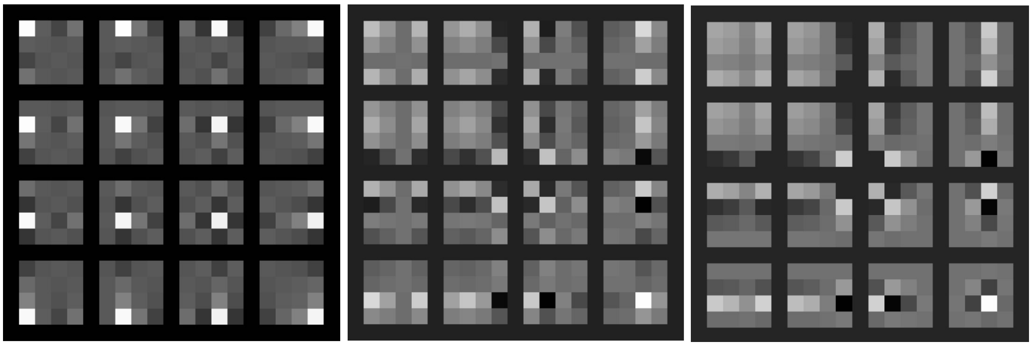

FIGURE 4 The basis functions for (from left to right) the fractional 4×4 DCT with $\alpha = 0.1, 0.5, 0.7$

$$v_1 = p_1 + iq_1 \equiv [0.613, 0.237, 0.206, 0.159] + i[0, -0.512, 0.285, 0.396]$$

$$v_2 = p_2 + iq_2 \equiv [-0.206, 0.159, 0.613, -0.237] + i[0.285, -0.396, 0, -0.512]$$

These result in the following *eigen-bivectors*:

$$B1 = p_1 \wedge q_1 \quad \text{and} \quad B2 = p_2 \wedge q_2$$

where $B1$ and $B2$ commute. It is then not hard to show that

$$Inv1 \propto (B1 + B2)$$

Therefore the sum of the *commuting blades* which are the *eigenblades* gives the *eigen-bivector* formed via the *characteristic multivectors*. An alternative approach would be to take the B formed from the R and split it into commuting blades via, for example, techniques outlined in Roelfs & de Keninck[5].

Having expressed the DCT as a rotor, we can again investigate what *fractional transforms* look like by following the procedure below (again, while we use the Cayley transform here, other functions of the bivector can be used):

- Use the Cayley transform to find B given R:

$$R = \frac{1 - B}{1 + B} \qquad B = \frac{1 - R}{1 + R}$$

- Create a new rotor using αB, $\alpha \in [0, 1]$:

$$R_\alpha = \frac{1 - \alpha B}{1 + \alpha B}$$

Figure 4 shows the basis functions for three fractional DCTs: $\alpha = 0.1, 0.5, 0.7$. As expected, for $\alpha = 0.1$ each basis function is approx a delta function at a given pixel, while the other basis functions are more difficult to interpret, but are moving more towards the case of $\alpha = 1$ shown in Figure 3 .

What uses might we put such fractional transforms to? One possibility is, for a given image, to compress using a transform optimised for that particular image – the decoder would only need to know one extra scalar, α. However, finding an optimal α would perhaps be difficult (it would perhaps need to be learned!).

5 | FORMING THE ‘CLOSEST’ ORTHOGONAL TRANSFORM

One common task which occurs in a number of fields (eg computer vision, molecular dynamics, robotics etc.) is to find, given noisy sets of vectors (which span the space), the rotor which takes one to another. Another version of this (which can be viewed as having noise only on the target set of vectors) is to find the ‘closest’ orthogonal matrix to a given non-orthogonal matrix.

This problem has been addressed multiple times in the literature Kabsch[6],Horn *et al*[7], Lasenby *et al*[8], Wu *et al*[9], and in this volume the problem has also been approached from a particular decomposition viewpoint by Sarabandi & Thomas[10] (also to appear in this volume – citation when available).

Given two sets of noisy vectors, $\{e_i\}$ and $\{f_i\}$, (or one set of vectors mapping to another set of vectors with added noise) where the true (no-noise) vectors are related via an underlying rotor, can we use the characteristic multivector formula directly (with the noisy vectors) to get the closest [in some sense] orthogonal transform/rotor relating the sets? Using the characterisic multivector formula will produce a rotor for Euclidean signatures (this will be shown elsewhere, but it is easy to verify symbolically for a given dimension), but how does this rotor relate to, for example, the rotor produced by the standard SVD formula, which we know should be optimal under gausssian noise (as it satisfies the least squares criterion)?

Here we present a preliminary investigation of this (a more in-depth study will be presented elsewhere) by asking the following question: given a 4×4 matrix, $\boldsymbol{M}$, find the rotor relating $\{e_i\}, i = 1, .., 4$ to $\{f_i\}, i = 1, .., 4$, which are the (non-orthogonal) columns of $\boldsymbol{M}$ via

$$\begin{gathered}\tilde{R} \propto 1 + [e_1 f_1 + e_2 f_2 ....] + [(e_2 \wedge e_1)(f_1 \wedge f_2) + ...] + \\ [(e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3) + ...] + [(e_4 \wedge e_3 \wedge e_2 \wedge e_1)(f_1 \wedge f_2 \wedge f_3 \wedge e_4)]\end{gathered}$$

(always give a valid rotor) and investigate the nature of this rotor. For a given rotor $\boldsymbol{R}$, our simulations will add both uniform and gaussian noise to the vectors $\{\bar{f}_i\}$ where $\bar{f}_i = \boldsymbol{R} e_i \tilde{\boldsymbol{R}}$ (adding noise to each of the components of the vector) and will then compare the recovered rotor to $\boldsymbol{R}$.

5.1 | Some preliminary results

In this section we will compare the standard *Singular Value Decomposition* (SVD) method for finding the ‘optimal’ rotor between two sets of vectors, with the *Characteristic Multivector* (CM) described in earlier sections. This will be done for the 4D case and for the case of an orthogonal set $\{e_i\}$ mapping to a non-orthogonal set, $\{f_i\}$ – which is equivalent to finding the closest orthogonal matrix to a given non-orthogonal matrix (so that its columns are the $\{f_i\}$). Following the presentation of results by Sarabandi & Thomas[10], and in order to compare with their findings at some later stage, the generation of the 4×4 orthogonal matrices to which noise is added follows the same procedure of doing this via sampling points on a 4-sphere (see below).

Using double precision arithmetic, we repeat the procedure outlined below for both uniform and gaussian noise. For the uniform case, additive noise is taken from a uniform distribution in the interval $[-a, a]$ with a in the interval $[0, 0.1]$ with 100 intermediate noise levels. For the gaussian case, additive noise is taken from a normal distribution with zero mean and standard deviation σ, where again σ takes 100 evenly spaced values between 0 and 0.1. The simulation then proceeds as follows:

1. Generate two sets of 100 quaternions, q_r and q_l using the *‘Method of choosing a point on the 4-sphere’* as presented in Marsaglia[11]. This method chooses a point according to a uniform distribution on the surface of the unit 4-sphere.
2. Any 4D rotation (orthogonal) matrix can be written as a commutative product of two orthogonal matrices (see Cayley[12], Kim & Rote[13]). In geometric algebra terms, this involves splitting the bivector, $\boldsymbol{B}$ (such that, for example, the rotor $\boldsymbol{R}$ corresponding to the rotation, can be written as $\exp \boldsymbol{B}$) into commuting blades, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ ($\boldsymbol{B} = \boldsymbol{B}_1 + \boldsymbol{B}_2$), so that our two commuting rotors are $\exp \boldsymbol{B}_1$ and $\exp \boldsymbol{B}_2$. In matrix terms this can be done by decomposing into *right- and left-isoclinic* matrices as in Kim & Rote[13]. Since each of these commuting rotors or *right- and left-isoclinic* matrices can be written as quaternions, it can also be said that any 4D rotation can be represented by a *pair of quaternions* or a *double quaternion*.

   These double quaternions are then converted to 4D rotation matrices: create the 4D rotation matrix by writing it as a Van Elfrinkhof matrix (Elfrinkhof[14], Sarabandi *et al*[15]).

3. Add uniform/gaussian distributed noise to each element of the 4D rotation matrices generated.

4. Compute the 4D 'nearest' rotation matrices using both the MATLAB[16] in-built function for SVD and the CM method. This is done using double precision arithmetic for both methods. For the rotor obtained from the CM method, also convert to a rotation matrix.

5. Compute the maximum and the average squared Frobenius norm of the difference between the rotation matrices without noise, and the obtained nearest rotation matrices from the SVD and CM respectively. The Frobenius norm is:

$$||A||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}^2} \tag{25}$$

   where a_{ij} is the ijth element of the matrix, and N is the matrix size, 4 in this case.

6. Compute the maximum and the average *orthogonality error* obtained using the nearest rotation matrices from the SVD and CM methods; if $\boldsymbol{R}$ or R is the original rotor/rotation matrix, and $\boldsymbol{R}^*$ or R* is the estimated rotor/rotation matrix, the *orthogonality error* is given by:

$$||R\tilde{R}^* - 1||_F \quad \text{or} \quad ||\mathrm{RR}^{*T} - \mathrm{I}||_F \tag{26}$$

Figure 5 shows the results for uniform noise – both the maximum and mean Frobenius norm (over 100 trials) are shown. We see that the mean results are very similar, but note that the CM method produces a more variable maximum error for higher noise levels. However, we see in figure 6 that the CM method produces a marginally 'more orthogonal' rotor.

Figures 7 and 8 show the equivalent results for gaussian noise where, again, the SVD produces a marginally better average Frobenius error while the CM method is marginally more orthogonal.

We note that if single precision arithmetic is used the errors are significantly higher for the SVD method, but that the CM method results are affected less.

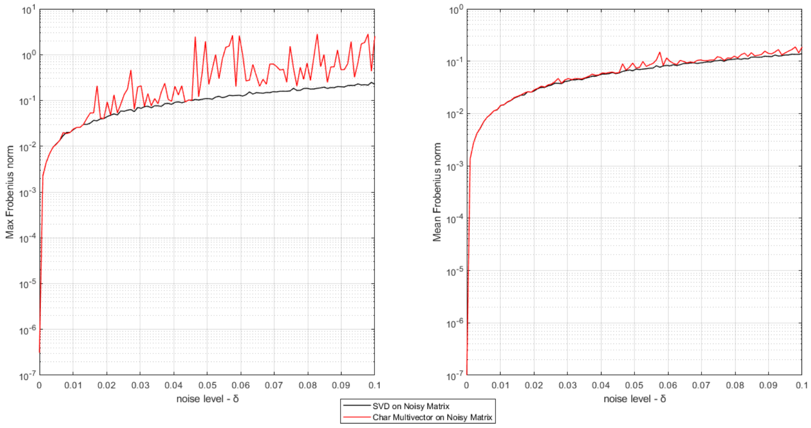

FIGURE 5 The maximimum (left) and mean (right) Frobenius norm between the original (R) and estimated (R*) rotation matrices with the SVD and CM methods (over 100 trials), against noise level. x-coordinate indicates noise is taken from a uniform distribution between $[-x, x]$.

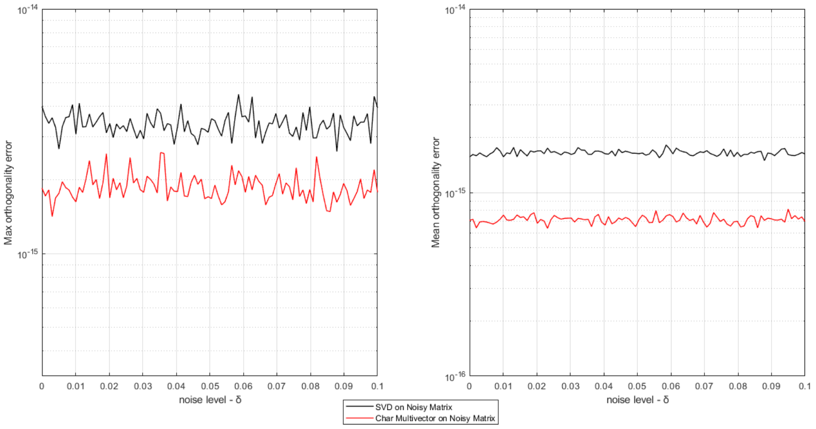

FIGURE 6 The maximum (left) and mean (right) orthogonality error between the original (R) and estimated (R^*) rotation matrices with the SVD and CM methods (over 100 trials), against noise level. x-coordinate indicates noise is taken from a uniform distribution between $[-x, x]$.

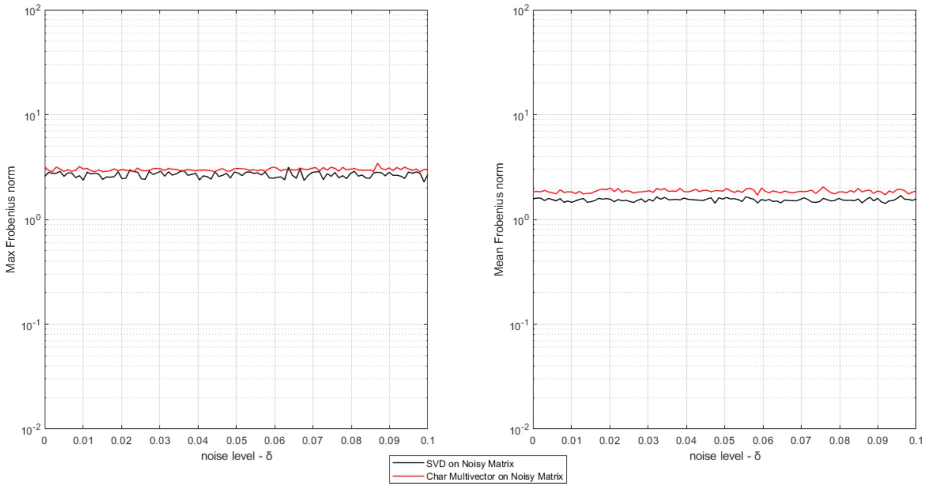

FIGURE 7 The maximum (left) and mean (right) Frobenius norm between the original (R) and estimated (R^*) rotation matrices with the SVD and CM methods (over 100 trials), against noise level. x-coordinate indicates noise is taken from a normal distribution with zero mean and standard deviation x.

6 | CONCLUSIONS

As stated in the introduction, this paper was inspired by the work of Shirikov[3] who essentially gave the rotor formula in equation 6 for restricted cases – our aim was to give a more general formulation, and a proof that required only a knowledge of basic geometric algebra. We also took inspiration from the work of Sarabandi *et al*[10] for an investigation into what the nature of the characteristic multivector sum is when the start and end vectors are not related by a rotor.

Given any set of frame vectors $\{e_i\}$ (working in any dimension and any signature, provided no basis vector is null) and a set of frame vectors $\{f_i\}$ which are related to the original frame via a rotor R, ie $f_i = Re_i\tilde{R}$, we have shown how to recover R via a closed form expression (equation 6). Expressing orthogonal transformations as rotors provides us with a variety of possibilities, such as defining fractional transforms by interpolating the rotor – some illustrative examples are given for the Haar and DCT

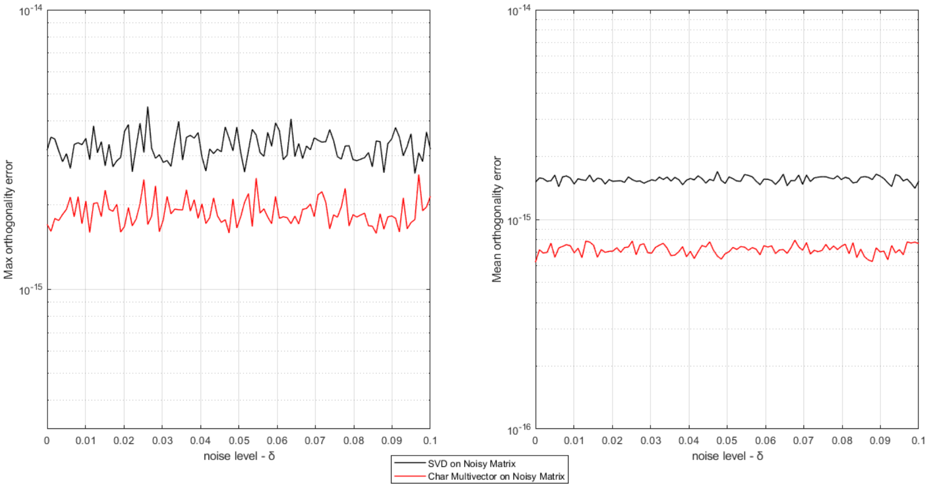

FIGURE 8 The maximimum (left) and mean (right) orthogonality error between the original (R) and estimated (R^*) rotation matrices with the SVD and CM methods (over 100 trials), against noise level. x-coordinate indicates noise is taken from a normal distribution with zero mean and standard deviation x.

transforms. Finally, we note that if the $\{e_i\}$ and $\{f_i\}$ are *not* related by a rotor, the characteristic multivector (CM) formula still gives a rotor in the Euclidean case. This rotor appears to be closely related to the UV^T (or VU^T) construction obtained from the SVD – in fact we believe that it is the optimal rotor obtained when minimising over the difference of *all geometric objects* not just rotors. We also show that if we use the CM method to obtain the 'closest' orthogonal matrix to a noisy orthogonal matrix, the CM performs marginally better (in terms of Frobenius norm and orthogonality measure) than the SVD under double precision arithmetic. This is likely to be due to the difference between an iterative (SVD) algorithm and a closed form solution (CM). The hope is that now this method has been shown to work in practice, more diverse applications will be found.

□

APPENDIX

A: Proof of equal numbers of commuting and anticommuting instances

We aim to show that for *each* basis element P in the $N = 2^n$ space, i.e. not just for even grade elements, then P commutes with $N/2$ elements and anticommutes with $N/2$. The exceptions are for P being the scalar element, where it commutes with all others, or for P being the pseudoscalar in a space with odd n, where again it commutes with all others.

Considering pseudoscalars first, then from considering passing an e_j through $I = e_1 \dots e_n$, then in a space with n even, even grade blades commute with I and odd grade blades anticommute with I. Since there are the same number of even and odd grade blades, this establishes what we want for the pseudoscalar in such a space.

For a space with n odd, it is easy to see that the pseudoscalar commutes with all elements, as stated above.

So now consider a non-pseudoscalar blade P in the $N = 2^n$ space. Since it is not a pseudoscalar, there must be at least one vector element, e_j say, which is not part of the blade. Now consider the reduction of the overall space to a space in which none of the elements contain e_j, together with its complement where all the elements contain e_j. Call these spaces A and B; both A and B have $2^{n-1} = N/2$ blades and together they make up the whole 2^n space. By construction $P \in A$. The only thing we have to guard against is whether it has now become the pseudoscalar of an odd n space. This could happen if the original space had an even n and P was of highest odd grade. For the moment, let us suppose this is not the case. Now suppose that our theorem (that P commutes and anticommutes with an equal number of blades) is true for $n-1$; P must therefore commute with $N/4$ elements of A and anticommute with $N/4$ elements of A. Call these subsets $A^{(1)}$ and $A^{(2)}$ respectively. Since P does not contain e_j, then elements of the form $e_j A^{(1)}$ will commute with P if P is even, and anticommute with P if P is odd. Similarly, elements of the form $e_j A^{(2)}$ will anticommute with P if P is even, and commute with P if P is odd. Thus, for P either even or odd, we see that the subset B is divided equally into elements that commute and anticommute with P. If the theorem is true for $n-1$ it is therefore true for n as well.

The only thing to check is what happens if when reducing to the subspace not containing e_j, then P becomes the pseudoscalar of that space. As we have said, this can only happen if P is odd. In this case P commutes with all of 'A', which looks very different to above. However, now all elements of the form $e_j A$, i.e. all elements of B, will anticommute with P, since all elements of A commute with with P and e_j anticommutes, since P is odd. Thus the same conclusion is reached, ie P commutes with $N/2$ and anticommutes with $N/2$ elements in the 2^n space, although it happens in a different way in this case.

It is trivially true for $r=2$ since vectors (e_1, e_2) commute with the scalar and pseudoscalar/bivector $(e_1 e_2)$ elements and anticommutes with the vectors, while the bivector commutes with the bivector element and anticommutes with the vectors – thus we have things commuting with 2 elements and anticommuting with 2 elements of the space. If it is true for $r=2$ it is then true for all r.

References

1. Hestenes D., Sobczyk G.. *Clifford Algebra to Geometric Calculus. A Unified Language for Mathematics and Physics*. Kluwer Academic Publishers, Dordrecht; 1984.
2. Doran C. J. L. D., Lasenby A. N.. *Geometric Algebra for Physicists*. Cambridge University Press; 2003.
3. Shirokov D. Calculation of elements of spin groups using method of averaging in Clifford's geometric algebra. *Advances in Applied Clifford Algebras.* 2019;29(3):50.
4. Doran C. J. L., Hestenes D., Sommen F., Acker N.. Lie Groups as Spin Groups. *J. Math. Phys.*. 1993;34(8):3642-3669.
5. Roelfs M., deKeninck S.. Graded Symmetry Groups, Plane and Simple. *arXiv: math-ph; 2107.03771v1.* 2021;.
6. Kabsch W.. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica.* 1976;A34(5):827-828.
7. Horn B. K. P., Hilden H. M., Negahdaripour S.. Closed form solution of abso- lute orientation using orthonormal matrices. *Journal of the Optical Society of America A.* 1988;5(7):1127-1135.
8. Lasenby J., Fitzgerald W. J., Lasenby A. N., Doran C. J. L.. New Geometric Methods for Computer Vision: An Application to Structure and Motion Estimation. *International Journal of Computer Vision.* 1998;26:191-213.
9. Wu J., Sun Y., Wang M., Liu M.. Hand-eye calibration: 4D Procrustes analysis approach. *IEEE Transactions on Instrumentation and Measurement.* 2020;69:2966-2981.
10. Sarabandi S., Thomas F.. On closed-form solutions to the 4D nearest rotation matrix problem. *YouTube: https://www.youtube.com/watch?v=gs-GBWYTUi8.* 2021;.
11. Marsaglia G.. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics.* 1972;43(2):645-646.
12. Cayley A.. On certain results relating to quaternions. *Philosophical Magazine.* 1845;26.
13. Kim H., Rote G.. Congruence Testing of Point Sets in 4-Space. In: Fekete Sándor, Lubiw Anna, eds. *32nd International Symposium on Computational Geometry (SoCG 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 51: :48:1–48:16Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik; 2016; Dagstuhl, Germany.
14. Elfrinkhof L.. Eene eigenschap van de orthogonale substitutie van de vierde orde. *andelingen van het zesde Nederlandsch Natuurkundig en Geneeskundig Congres (Acts of the sixth Dutch physical and medical congress).* 1897;:237-240.

15. Sarabandi S., Perez-Gracia A., Thomas F.. On Cayley's Factorization with an Application to the Orthonormalization of Noisy Rotation Matrices. *Advances in Applied Clifford Algebras.* 2019;29(49).

16. MATLAB . *version R2021b.* Natick, Massachusetts: The MathWorks Inc.; 2021.