
How to Read This Guide to Geometric Algebra in Practice

This book is called a ‘Guide to Geometric Algebra in Practice’. It is composed of chapters by experts in the field and was conceived during the AGACSE-2010 conference in Amsterdam. As you scan the contents, you will find that all chapters indeed use geometric algebra but that the term ‘practice’ means different things to different authors. As we discuss the various *Parts* below, we guide you through them. We will then see that appearances may deceive: some of the more theoretical looking chapters provide useful and practical techniques.

This book is organized in themes of application fields, corresponding to the division into *Parts*. This is sometimes an arbitrary allocation; one of the powers of geometric algebra is that its unified approach permits techniques and representations from one field to be applied to another. In this guide we move, generally, from the description of physical motion and its measurement to the description of objects of a geometrical nature.

Basic geometric algebra, sometimes known as Clifford algebra, is well understood and arguably has been for many years. It is important to realize that it is not just one algebra, but rather a family of algebras, all with the same essential structure. A successful application for geometric algebra involves identifying, among those in this family, an algebra that considerably facilitates a particular task at hand. The current emphasis on rigid body motion (measurement, interpolation, tracking) has focused the attention on a specific geometric algebra, the *conformal model*. This uses an algebra in which such motions are representable as *rotations* in a carefully chosen model space (for 3D, a 5D space denoted $\mathbb{R}^{4,1}$, with a 32D geometric algebra denoted $\mathbb{R}_{4,1}$). Doing so is an innovation over traditionally used representations such as homogeneous coordinates, since geometric algebra has a particularly powerful representation of rotations (as ‘rotors’—essentially spinors, with quaternions as a very special case). This conformal model (CGA, for Conformal Geometric Algebra) is used in most of this book. We provide a brief tutorial introduction to its essence in the Appendix (Chap. 21), to make this guide more self-contained, but more extensive accessible introductions may be found elsewhere.

Application of geometric algebra started in physics. Using conformal geometric algebra, we can now update its description of motion in *Part I: Rigid Body Motion*. From there, geometric algebra migrated to applications in engineering and computer science, where motion tracking and image processing were the first to appreciate and apply its techniques. In this book these fields are represented in *Part II: Interpolation and Tracking* and *Part III: Image Processing*. More recently, traditionally

combinatorial fields of computer science have begun to employ geometric algebra to great advantage, as *Part IV: Theorem Proving and Combinatorics* demonstrates.

Although prevalent at the moment, the conformal model is not the only kind of geometric algebra we need in applications. It hardly offers more than homogeneous coordinates if your interest is specifically in projective transformations. It takes the geometric algebra of lines (the geometric algebra of a 6D space, for 3D lines), to turn such transformations into rotations (see *Part V: Applications of Line Geometry*), and reap the benefits from their rotor representation. And even if your interest is only rigid body motions, alternative and lower-dimensional algebras may do the job—this is explored in *Part VI: Alternatives to Conformal Geometric Algebra*.

While those parts of this guide show how geometric algebra ‘cleans up’ applications that would classically use linear algebra (notably in its matrix representation), there are other fields of geometrical mathematics that it can affect similarly. Foremost among those is differential geometry, in which the use of the truly coordinate-free methods of geometric algebra have hardly penetrated; *Part VII: Towards Coordinate-Free Differential Geometry* should offer inspiration for several PhD projects in this direction!

To conclude this introduction, some sobering thoughts. Geometric algebra has been with us in applicable form for about 15 to 20 years now, with general application software available for the last 10 years. There have been tutorial books written for increasingly applied audiences, migrating the results from mathematics to physics, to engineering and to computer science. Still, a conference on applications (like the one in Amsterdam) only draws about 50 people, just like it did 10 years ago. This is not compensated by integrated use and acceptance in other fields such as computer vision (which would obviate the need for such a specialized geometric algebra based gathering; after all, few in computer vision would go to a dedicated linear algebra conference even though everybody uses it in their algorithms). So if the field is growing, it is doing so slowly.

Perhaps a major issue is education. Very few, even among the contributors to this guide, have taught geometric algebra, and in none of their universities is it a compulsory part of the curriculum. Although we all have the feeling that we understand linear algebra much better because we know geometric algebra and that it improves our linear-algebra-based software considerably (in its postponement of coordinate choices till the end), we still have not replaced parts of linear algebra courses by the corresponding clarifying geometric algebra.¹ Most established colleagues may be too set in their ways to change their approach to geometry; but if we do not tell the young minds about this novel and compact structural approach, it may never reach its potential.

Our message to you and them is: ‘Go forth and multiply—but use the geometric product!’

¹The textbook *Linear and Geometric Algebra* (by Alan Macdonald, 2010) enables this, and we should all consider using it!

Part I: Rigid Body Motion

The treatment of rigid body motion is the first algebraically advanced topic that the geometry of Nature forces upon us. Since it was the first to be treated, it shaped the field of geometric representation; but now we can repay our debt by using modern geometric representations to provide more effective computation methods for motions. All chapters in this part use conformal geometric algebra to great advantage in compactness and efficiency.

Chapter 1: *Rigid Body Dynamics and Conformal Geometric Algebra* uses conformal geometric algebra to reformulate the Lagrangian expression of the classical physics of combined rotational and translational motion, due to the dynamics of forces and torques. It uses the conformal rotors ('spinors') to produce a covariant formulation and in the process extends some classical concepts such as inertia and Lagrange multipliers to their more encompassing geometric algebra counterparts. In its use of conformal geometric algebra, this chapter updates the use of geometric algebra to classical mechanics that has been explored in textbooks of the past decade. A prototype implementation shows that this approach to dynamics really works, with stability and computational advantages relative to more common methods.

As we process uncertain data using conformal geometric algebra, our ultimate aim is to estimate optimal solutions to noisy problems. Currently, we do not yet have an agreed way to model geometrical noise; but we *can* determine a form of optimal processing for conflicting data. This is done in quite general form for rigid body motions in Chap. 2: *Estimating Motors from a Variety of Geometric Data in 3D Conformal Geometric Algebra*. Polar decomposition is incorporated into conformal geometric algebra to study how motors are embedded in the even subalgebra and what is the best projection to the motor manifold. A general, dot-product-like similarity criterion is designed for a variety of geometrical primitives. Instances of this can be added to give a total similarity criterion to be maximized. Lagrangian optimization of the total similarity criterion then reduces motor estimation to a straightforward eigenrotor problem. The chapter provides a very general means of estimation and, despite its theoretical appearance, may be one of the more influential applied chapters in this book.

In robotics, the inverse kinematics problem (of figuring out what angles to give the joints to reach a given position and orientation) is notoriously hard. Chapter 3: *Inverse Kinematics Solutions Using Conformal Geometric Algebra* demonstrates that having spheres and lines as primitives in conformal geometric algebra really helps to design straightforward numerical algorithms for inverse kinematics. Since the geometric primitives are more directly related to the type of geometry one encounters, they lead to realtime solvers, even for a 3D hand with its 25 joints.

Another example of the power of conformal geometric algebra to translate a straightforward geometrical idea directly into an algorithm is given in Chap. 4: *Reconstructing Rotations and Rigid Body Motions from Exact Point Correspondences Through Reflections*. There, rigid body motions are reconstructed from corresponding point pairs by consecutive midplanes of remaining differences. Applying the algorithm to the special case of pure rotation produces a quaternion determination

formula that is twice as fast as existing methods. That clearly demonstrates that understanding the natural geometrical embedding of quaternions into geometric algebra pays off.

Part II: Interpolation and Tracking

Conformal geometric algebra can only reach its full potential in applications when middle-level computational operations are provided. This part provides those for recurring aspects of motion interpolation and motion tracking.

Chapter 5: *Square Root and Logarithm of Rotors in 3D Conformal Geometric Algebra Using Polar Decomposition*, gives explicit expressions for the square root and logarithm of rotors in conformal geometric algebra. Not only are these useful for interpolation of motions, but the form of the bivector split reveals the orthogonal orbit structure of conformal rotors. In the course of the chapter, a polar decomposition is developed that may be used to project elements of the algebra to the rotor manifold.

Geometric algebra offers a characterization of rotations through bivectors. Since these form a linear space, they permit more stable numerical techniques than the nonlinear and locally singular classical representations by means of, for instance, Euler angles or direction cosine matrices. Chapter 6: *Attitude and Position Tracking* demonstrates this for attitude estimation in the presence of the notoriously annoying ‘coning motion’. It then extends the technique to include position estimation, employing the bivectors of the conformal model.

An important step in the usage of any flexible camera system is calibration relative to targets of unknown location. Chapter 7: *Calibration of Target Positions Using Conformal Geometric Algebra* shows how this problem can be cast and solved fully in conformal geometric algebra, with compact simultaneous treatment of orientational and positional aspects. In the process, some useful conformal geometric algebra nuggets are produced, such as a closed-form formula for the point closest to a set of lines, the inverse of a linear mapping constrained within a subspace, and the derivative with respect to a motor of a scalar measure between an element and a transformed element. It also shows how to convert the coordinate-free conformal geometric algebra expressions into coordinate-based formulations that can be processed by conventional software.

Part III: Image Processing

Apart from the obviously geometrical applications in tracking and 3D reconstruction, geometric algebra finds inroads in image processing at the signal description level. It can provide more symmetrical ways to encode the geometrical properties of the 2D or 3D domain of such signals.

Chapter 8: *Quaternion Atomic Function for Image Processing* deals with 2D and 3D signals and shows us one way of incorporating the rotational structure into a quaternion encoding of the signal, leading to monogenic rather than Hermitian

signals. Kernel processing techniques are developed for these signals by means of atomic functions.

The facts that real 2-blades square to -1 and their direct correspondence to complex numbers and quaternions have led people to extend classical Fourier transforms by means of Clifford algebras. The geometry of such an algebraic analogy is not always clear. In the field of color processing, the 3D color space does possess a perceptual geometry that suggests encoding hues as rotations around the axis of grays. For such a domain, this gives a direction to the exploration of Clifford algebra extensions to the complex 1D Fourier transform. Chapter 9: *Color Object Recognition Based on a Clifford Fourier Transform* explores this and evaluates the effectiveness of the resulting encoding of color images in an image retrieval task.

Part IV: Theorem Proving and Combinatorics

A recurrent theme in this book is how the right representation can improve encoding and solving geometrical problems. This also affects traditionally combinatorial fields like theorem proving, constraint satisfaction and even cycle enumeration. The null elements of algebras turn out to be essential!

Chapter 10: *On Geometric Theorem Proving with Null Geometric Algebra* gives a good introduction to the field of automated theorem proving, and a tutorial on the authors' latest results for the use of the null vectors of conformal geometric algebra to make computations much more compact and geometrically interpretable. Especially elegant is the technique of dropping hypotheses from existing theorems to obtain new theorems of extended and quantitative validity.

As a full description of geometric relationships, geometric algebra is potentially useful and unifying for the data structures and computations in Computer Aided Design systems. It is beginning to be noticed, and in Chap. 11: *On the Use of Geometric Algebra in Geometrical Constraint Solving* the structural cleanup conformal geometric algebra could bring is explored in some elementary modeling computations.

Part of the role geometric algebra plays as an embedding of Euclidean geometry is a consistent bookkeeping of composite constructions, of an almost Boolean nature. The Grassmann algebra of the outer product, in particular, eliminates many terms 'internally'. In Chap. 12: *On the Complexity of Cycle Enumeration for Simple Graphs*, that property is used to count cycles in graphs with n nodes, by cleverly representing the edges as 2-blades in a $2n$ -dimensional space and their concatenations as outer products. Filling the usual adjacency matrix with such elements and multiplying them in this manner algebraically eliminates repeated visits. It produces compact algorithms to count cycle-based graph properties.

Part V: Applications of Line Geometry

Geometric algebra provides a natural setting for encoding the geometry of 3D lines, unifying and extending earlier representations such as Plücker coordinates. This is immediately applicable to fields in which lines play the role of basic elements of

expression, such as projective geometry, inverse kinematics of robots with translational joints, and visibility analysis.

Chapter 13: *Line Geometry in Terms of the Null Geometric Algebra over $\mathbb{R}^{3,3}$, and Application to the Inverse Singularity Analysis of Generalized Stewart Platforms* provides a tutorial introduction on how to use the vectors of the 6D space $\mathbb{R}^{3,3}$ to encode lines and then applies this representation effectively to the analysis of singularities of certain parallel manipulators in robotics. Almost incidentally, this chapter also indicates how in the line space $\mathbb{R}^{3,3}$, projective transformations become representable as rotations. Since this enables projective transformations to be encoded as rotors, this is a potentially very important development to the applicability of geometric algebra to computer vision and computer graphics.

In Chap. 14: *A Framework for n -Dimensional Visibility Computations*, the authors solve the long-standing problem of computing exact mutual visibility between shapes, as required in soft shading rendering for computer graphics. It had been known that a Plücker-coordinate-based approach in the manifold of lines offered some representational clarity but did not lead to efficient solutions. However, the authors show that when the full bivector space $\bigwedge^2(\mathbb{R}^{n+1})$ is employed, visibility computations reduce to a convex hull determination, even in n -D. They can then be implemented using standard software for CSG (Computational Solid Geometry).

Part VI: Alternatives to Conformal Geometric Algebra

The 3D conformal geometric algebra $\mathbb{R}_{4,1}$ is five-dimensional and often feels like a slight overkill for the description of rigid body motion and other limited geometries. This part presents several four-dimensional alternatives for the applications we saw in Part I.

Embedding the common homogeneous coordinates into geometric algebra begs the question of what metric properties to assign to the extra representational dimension. Naive use of a nondegenerate metric prevents encoding rigid body motions as orthogonal transformations in a 4D space. Chapter 15: *On the Homogeneous Model of Euclidean Geometry* updates results from classical 19th century work to modern notation and shows that by endowing the dual homogeneous space with a specific degenerate metric (to produce the algebra $\mathbb{R}_{3,0,1}^*$) one can in fact achieve this. The chapter reads like a tutorial introduction to this framework, presented as a complete and compact representation of Euclidean geometry, kinematics and rigid body dynamics.

To some in computer graphics, the 32-dimensional conformal geometric algebra $\mathbb{R}_{4,1}$ is just too forbidding, and they have been looking for simpler geometric algebras to encode their needs. Chapter 16: *A Homogeneous Model for Three-Dimensional Computer Graphics Based on the Clifford Algebra for \mathbb{R}^3* shows that a representation of some operations required for computer graphics (rotations and perspective projections) can be achieved by rather ingenious use of \mathbb{R}_3 (the Euclidean geometric algebra of 3-D space) by using its trivector to model mass points.

In Chap. 17: *Rigid-Body Transforms Using Symbolic Infinitesimals*, an alternative 4D geometric algebra is proposed to capture the structure of rigid body motions. It is nonstandard in the sense that one of the basis vectors squares to infinity. The authors show how this models Euclidean isometries. They then apply their algebra to Bezier and B-spline interpolation of rigid body motions, through methods that can be transferred to more standard algebraic models such as conformal geometric algebra.

Chapter 18: *Rigid Body Dynamics in a Constant Curvature Space and the ‘1D-up’ Approach to Conformal Geometric Algebra* proposes yet another way to representing 3D rigid body motion in the geometric algebra of a 4D space. It takes the unusual approach of viewing Euclidean geometry as a somewhat awkward limit case of a constant curvature space and analyzes such spaces first. The Lagrangian dynamics equations take on an elegant form and lead to the surprising view of translational motion in real 3D space as a fast precession in the 4D representational space. The author then compares this approach to that of Chap. 15, after first embedding that into conformal geometric algebra; and the flat-space limit to the Euclidean case is shown to be related to Chap. 17. Thus all those 1D up alternative representations of rigid body motions are shown to be closely related.

Part VII: Towards Coordinate-Free Differential Geometry

Differential geometry is an obvious target for geometric algebra. In its classical description by means of coordinate charts, its structure easily gets hidden in notation, and that limits its applications to specialized fields. Geometric algebra should be able to do better, especially if combined with modern insights in the system of geometrical invariants.

Chapter 19: *The Shape of Differential Geometry in Geometric Calculus* shows how geometric algebra can offer a direct notation in terms of clear concepts such as the tangent volume element (‘pseudoscalar’), attached at all locations of a vector manifold, and in terms of its derivative as the codification of ‘shape’ in all its aspects. The coordinate-free formulation can always be made specific for any chosen coordinates and is hence computational. The chapter ends with open questions, intended as suggestions for research projects. The editors are grateful to have this thought-provoking contribution by David Hestenes, the grandfather of geometric algebra.

The field of “moving frames” has developed rapidly in the past decade, and structured algorithmic methods are emerging to produce invariants and their syzygy relationships for Lie groups. We have invited expert Elisabeth Mansfield, in Chap. 20: *On the Modern Notion of Moving Frames*, to write an introduction to this new field, since we believe that its concretely abstract description should be a quite natural entry to formulate invariants for the Lie groups occurring in geometric algebra. Besides an introductory overview with illustrative examples and detailed pointers to current literature, the chapter contains a first attempt to compute moving frames for $SE(3)$ in conformal geometric algebra.

Part VIII: Tutorial Appendix

In Chap. 21: *Tutorial Appendix: Structure Preserving Representation of Euclidean Motions through Conformal Geometric Algebra*, we provide a self-contained tutorial to the basics of geometric algebra in the conformal model.

Leo Dorst
Joan Lasenby

<http://www.springer.com/978-0-85729-810-2>

Guide to Geometric Algebra in Practice

Dorst, L.; Lasenby, J. (Eds.)

2011, XVII, 458 p., Hardcover

ISBN: 978-0-85729-810-2