Constructing conformal motions

LEO DORST is assistant professor at the Informatics Institute (UvA), specialised in geometric algebra. Currently he is using mathematical morphology for the reassembly of broken archaeological artefacts (H2020 project GRAVITATE).

→ Reference

1: L. Dorst. The construction of 3D conformal motions, *Mathematics in Computer Science* 10, 97–113 (2016). https://doi.org/10.1007/s11786-016-0250-8
2: L. Dorst, D. Fontijne, S. Mann. *Geometric Algebra for Computer Science*, Morgan Kauffman, 2007. ISBN13: 9780123749420. http://www.geometricalgebra.net

"A snail shell traces out a conformal motion"

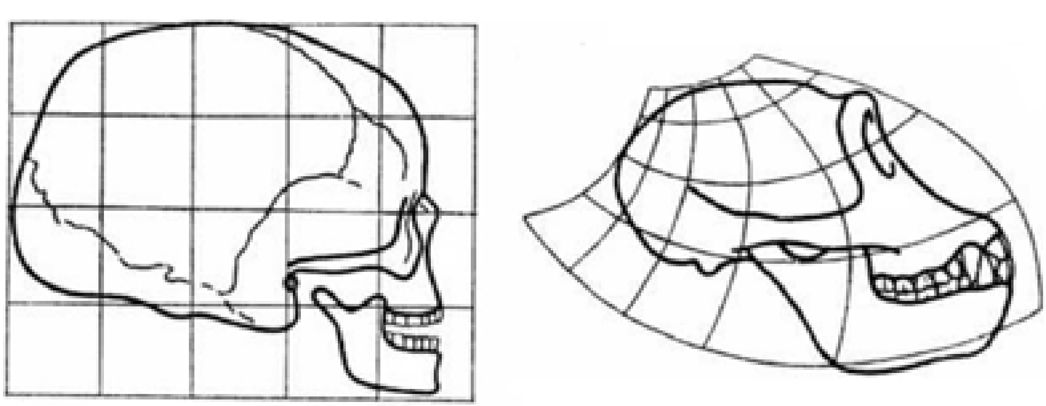
→ Imagine a very fine square grid on a piece of paper. It is possible to smoothly transform (or 'warp') this grid in such a way that the grid-lines are still locally perpendicular to each other at their crossing points, but the lines themselves are no longer straight. This is called a *conformal transformation*. If you draw a picture on the original square grid, and then let it undergo such a transformation, the image looks like a warped but recognisable version of the original, see Figure 1. Such 'shape-preserving warps' can be easily extended to 3D space. 3D conformal transformations are used in computer graphics to make acceptably different avatars from prototype shapes or to blend surfaces smoothly, and have also been applied to produce organic architectural surfaces. In some applications, we want to gradually deform an object, generating a *conformal motion*. This requires parameterising the transformation, which is often done by 'factorising it' into a standardised sequence of easily parameterisable basic transformations. As an example, any rigid body motion (which is a special case of conformal motion) can be parameterised as a rotation around an axis through the origin (giving an object the desired orientation), followed by a translation (to put the object in its place). This seems natural but is actually awkward; although the starting and ending point of the motion might be correct, the motion itself is not generally realistic or useful. (When moving a camera, one does not first point it correctly and then slide it without turning; that would make for an awkward pan.) Moreover, twice doing 'half' the motion (as characterised by halving the parameter values) leads to a different result. A more natural factorisation is achieved by performing the rotation through a general axis (not through the origin), done simultaneously with a translation along that same axis. Now the motion can be easily interpolated: $1/n$-th of the motion (characterised by $1/n$-th of the parameter values) done n times gives the same result. This generates a step-wise screw motion from initial to final position.

For general conformal motions, only factorisations of the first type were known (involving translations, rotations, scalings and transversions). They could not be converted into 'simultaneously executable simple motions' because of the complicated commutation rules between those four basic operations: when you swap their order, the result is totally different.

A new factorisation

In our work [1], we have found a way to factorise 3D conformal motions using only two basic operations. Moreover, these two operations have a straightforward geometry: they are both circular motions governed by a pair of points. One can picture these point pairs as being the two poles of a sphere (Figure 2). We can classify the point pairs into three basic kinds, determined by the square of the diameter of the sphere, as indicated in the figure. Each point pair determines the motion for any point x: it moves on a circular orbit that either passes through the sphere's poles (real point pair); lies tangential to the point pair (tangent vector); or crosses the surface of the sphere at right angles (imaginary point pair). When we choose two point pairs (from these three fundamental choices) with the right mutual location, perpendicular orientation and specific relative size, their two motions commute and therefore can be done simultaneously, 'a bit of one, then a bit of the other'. The composite motion then moves any object (such as a point x) on a spatial net of orthogonal circles, determined fully by the two point pairs. If you would move a little triangle with your point x, its angles would be preserved: that is what it means to move conformally. It is very satisfying, and simplifying, that any conformal motion in 3D can be represented simply by defining two orthogonal point pairs.

In Figures 3 and 4, we give some examples of the motions we can generate. First off, in 2D, a combination of a real and an imaginary point pair reproduces the path of a charged particle moving between two charges in the presence of a perpendicular magnetic



a b

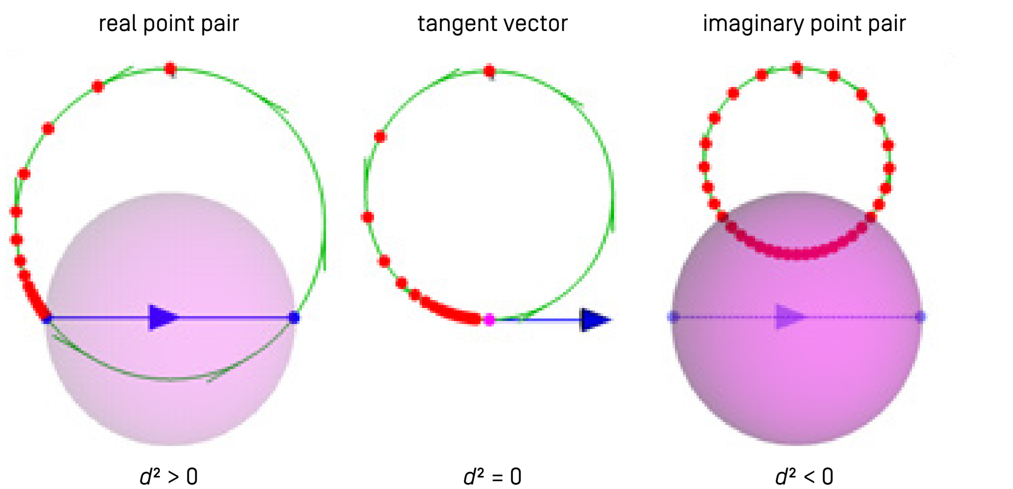
← Figure 1
The suggested correspondence in geometry between a human skull (a) and a chimpanzee skull (b), from the D'Arcy Thompson's classic work *On Growth and Form* (Cambridge University Press 1917), is a conformal transformation.

field (Figure 3). Finally, as a snail grows, it maintains its local shape (as defined by local angles), so a snail shell traces out a conformal motion (Figure 4); the combination of a real point pair that scales and an imaginary point pair that rotates can generate a motion in which a point and a circle naturally form a snail shell shape.

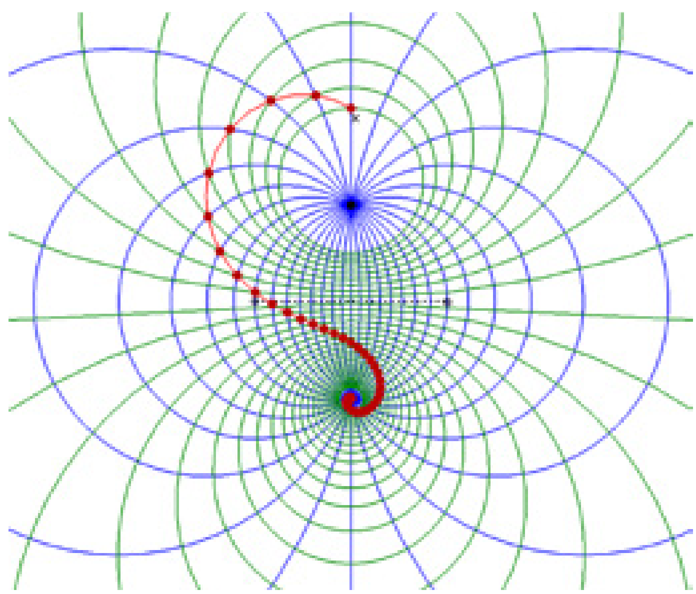
Geometric algebra

These results were obtained with geometric algebra [2], which is essentially a method to compute directly with geometric primitives. In the particular geometric algebra used, the primitive objects are 3D spheres and their intersections. A certain product of two spheres is a new element representing their intersection, and so is a circle; multiplying by a sphere again gives a point pair (the intersection of 3 spheres). This 3D conformal geometric algebra lives in a space of 4 positive dimensions and 1 negative dimension (this is the Minkowski space $\mathbb{R}^{4,1}$). In this space, the geometric perpendicularity we need (as mentioned earlier) is equivalent to algebraic orthogonality (zero dot product). Arithmetic operations that one is used to performing on numbers can be interpreted meaningfully for the geometric primitives. For instance, the square root of the ratio of two circles C1 and C2 is a conformal transformation that moves C1 to C2, and the logarithm of that transformation gives the point pairs of that motion. Because some point pairs square to a negative number, they can generate rotations. Moreover, the same motion can be applied to any object in the algebra.

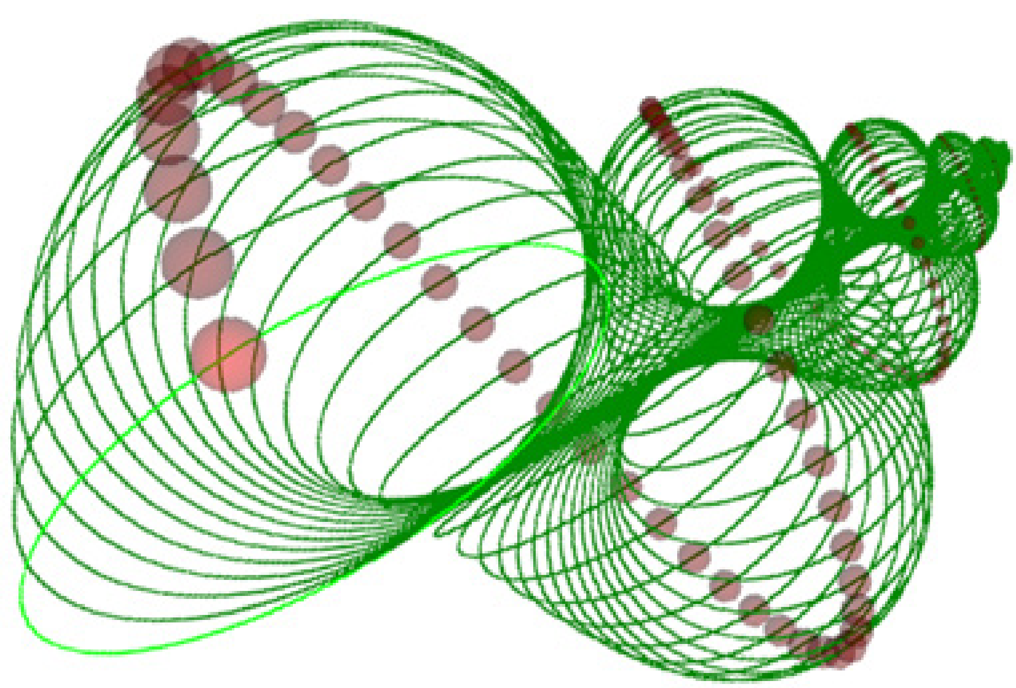
By this new approach, we have a tool that allows direct control over a class of conformal shapes that is intriguingly organic in their motions, yet very simple to generate. These can form new primitives for computer graphics, architecture and manufacturing. All this is a particular spin-off of our general work in geometric algebra. The tight relationship between algebra and geometry enables graphical computer interfaces that turn interactive manipulation of objects into the corresponding computations, and vice versa. The algebraic structure then allows the compiler, rather than the programmer, to generate efficient and error-free software. Ω



↑ Figure 2
The three types of point pairs (denote by the black arrows), defined as the poles of a sphere (magenta), form our alphabet to describe general motions. The diameter *d* of the sphere (real-valued, zero, or imaginary) determines how the point pair determines the motion of a point *x*. Steps on the orbit denote the varying speed of movement.



← Figure 3
A 2D conformal motion in a plane generated by a real and imaginary point pair.



↓ Figure 4
Snail shells grow conformally, as illustrated by the conformal motion of a circle and a small sphere.

Omission of 3D figure from ‘Constructing Conformal Motions” in Amsterdam Science 8, November 2018, pg. 16-17, committed during the publication process. This omission is really unfortunate since it shows that the Geometric Algebra approach naturally extends to 3D (whereas 2D things like Figure 3 have been done by complex numbers before), and that there is a fascinating and tantalizing connection with knots. Insert this at the top of page 17.

…field (Figure 3). In 3D, a well-chosen combination of two imaginary point pairs generates the motion of a point to form a knot (on a Dupin cycloid, which is a conformally transformed torus, Figure *below*). We could apply the same spinor to a circle and generate a knot of moving circles. Finally…