9th International Conference on Clifford Algebras
and their Applications in Mathematical Physics
K. Gürlebeck (ed.)
Weimar, Germany, 15–20 July 2011

CONFORMAL GEOMETRIC OBJECTS WITH FOCUS ON ORIENTED POINTS

Dietmar Hildenbrand*, **Patrick Charrier**

*TU Darmstadt
Rundeturmstr. 12, 64283 Darmstadt
E-mail: hildenbrand@cocoon.tu-darmstadt.de

Keywords: Oriented Points, Geometric Algebra, Interpolation

Abstract. *In this paper, we explore the geometric objects of conformal geometric algebra based on their IPNS (inner product null space) representation in some detail. Spheres of dimension 1 , 2 and three are objects of conformal geometric algebra. Usually, points in conformal geometric algebra are represented as ordinary spheres with zero radius, but what about circles with zero radius? We expect many practical applications of these points with additional orientation information.*

1 INTRODUCTION

In recent years, geometric algebra, and especially the 5D conformal geometric algebra, has proved to be a powerful tool for the development of geometrically intuitive algorithms in a lot of engineering areas like robotics, computer vision and computer graphics.

While points and vectors are normally used as basic geometric objects, in the 5D conformal geometric algebra we have a wider variety of basic objects. For example, spheres and circles are simply represented by algebraic objects. To represent a circle you only have to intersect two spheres, which can be done with a basic algebraic operation. For more details please refer for instance to the books [4] and [13], as well as to the tutorials [8] and [6].

The goal of this paper is to describe the conformal geometric objects based on the conventional 3D objects like 3D points, 3D vectors or radius as well as in terms of the conventional products: cross product and scalar product.

It is well known that a sphere

$$S = \mathbf{x} + \frac{1}{2}(\mathbf{x}^2 - r^2)e_\infty + e_0 \tag{1}$$

can be represented based on its 3D center point $\mathbf{x}$, its scalar product $\mathbf{x}^2$ and the radius r. Note that **bold** face letters in equations indicate a standard euclidian 3D vector. Usually a sphere with zero radius is taken as a point in conformal geometric algebra:

$$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0 \tag{2}$$

Alternatively circles can also be described as the intersection of a sphere and a plane (normal $\mathbf{n_c}$) resulting in the following formula.

$$C = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) \tag{3}$$

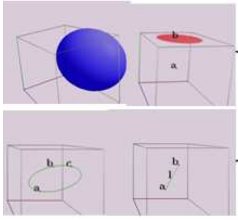
$$+ \left[\frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty.$$

This new representation brings about the direct possibility to extract the normal n_c and the center point c of the circle multivector. The normal vector n_c is a standard euclidian vector with blades e_1,e_2, and e_3. The operation $n_c \wedge e_0$ integrates the normal as linear combination $n_{c1} * e_1 \wedge e_0 + n_{c2} * e_2 \wedge e_0 + n_{c3} * e_3 \wedge e_0$ into the circle multivector. It is therefore trivial to retrieve the normal, by accessing the coefficients of blades $e_1 \wedge e_0$, $e_2 \wedge e_0$, and $e_3 \wedge e_0$. Retrieving the center point c is slightly more complex and is treated in a detailed way in section 4.2.

An oriented point with vanishing radius is defined by

$$O_p = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) + \left[\frac{1}{2}\mathbf{c}^2\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty. \tag{4}$$

As with the alternate circle, it is also very simple to extract the properties of point $\mathbf{c}$ and normal $\mathbf{n_c}$ from an oriented point O_p.



grade	term	blades	nr.
0	scalar	1	1
1	vector	$e_1, e_2, e_3, e_0, e_\infty$	5
2	bivector	$e_1 \wedge e_2$, $e_1 \wedge e_3$, $e_2 \wedge e_3$, $e_1 \wedge e_\infty$, $e_2 \wedge e_\infty$, $e_3 \wedge e_\infty$, $e_1 \wedge e_0$, $e_2 \wedge e_0$, $e_3 \wedge e_0$, $e_0 \wedge e_\infty$	10
3	trivector	...	10
4	quadvector	$e_1 \wedge e_2 \wedge e_3 \wedge e_\infty$, $e_1 \wedge e_2 \wedge e_3 \wedge e_0$, $e_1 \wedge e_2 \wedge e_0 \wedge e_\infty$, $e_1 \wedge e_3 \wedge e_0 \wedge e_\infty$, $e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	5
5	pseudoscalar	$e_1 \wedge e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	1

Figure 1: The blades of conformal geometric algebra. Spheres and planes, for instance, are vectors. Lines and circles can be represented as bivectors. Other mathematical systems like complex numbers or quaternions can be identified based on their imaginary units i, j, k. This is why also transformations like rotations can be handled within the algebra.

2 CONFORMAL GEOMETRIC ALGEBRA

Conformal geometric algebra is a 5D geometric algebra based on the 3D basis vectors e_1, e_2 and e_3 as well as on the two additional base vectors e_0 representing the origin and e_∞ representing infinity.

Blades are the basic computational elements and the basic geometric objects of geometric algebras. The 5D conformal geometric algebra consists of blades with *grades* (dimension) 0, 1, 2, 3, 4 and 5, whereby a scalar is a *0-blade* (blade of grade 0). The element of grade five is called the pseudoscalar. A linear combination of blades is called a *k-vector*. So a bivector is a linear combination of blades with grade 2. Other k-vectors are vectors (grade 1), trivectors (grade 3) and quadvectors (grade 4). Furthermore, a linear combination of blades of different grades is called a *multivector*. Multivectors are the general elements of a geometric algebra.

Table 1 lists the two representations of conformal geometric objects. The inner product null space (IPNS) and the outer product null space (OPNS) [13] are dual to each other. While we already presented an IPNS representation of circles and spheres, they can also be described with the outer product of 4 points being part of them. In the case of a plane one of these 4 points is the point at infinity e_∞. Circles can be described with the help of the outer product of 3 conformal points lying on the circle or as the intersection of two spheres.

In the OPNS, lines can be described with the help of the outer product of 2 points (alternatively one point pair) and the point at infinity e_∞. In the IPNS, lines can be described with the help of the outer product of 2 planes, which means the intersection of these planes.

3 OBSERVATIONS ON CONFORMAL GEOMETRIC OBJECTS LOCATED AT THE ORIGIN

A point P in Conformal Geometric Algebra (CGA) can be defined as $P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$ (see table 1), where $\mathbf{x}$ is a standard euclidian 3D vector $\mathbf{x} = x_1 e_1 + x_2 e_2 + x_3 e_3$.
In CGA the point at the origin is e_0. The inner product null space (IPNS) of an algebraic expression A is the set of all points P, for which the inner product $A \cdot P$ is equal to zero.

Table 1: The two representations (IPNS and OPNS) of conformal geometric objects. IPNS and OPNS representations are dual to each other, which is indicated by the star symbol. While IPNS is a direct parameterized representation, OPNS describes the geometric objects through an outer product of conformal points defining the object. For instance a line is the outer product of two points and the point at infinity.

object	**IPNS representation**	**OPNS representation**
Point	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$	
Sphere	$S = \mathbf{x} + \frac{1}{2}(\mathbf{x}^2 - r^2)e_\infty + e_0$	$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4$
Plane	$\pi = \mathbf{n} + de_\infty$	$\pi^* = P_1 \wedge P_2 \wedge P_3 \wedge e_\infty$
Circle	$C = S_1 \wedge S_2$	$C^* = P_1 \wedge P_2 \wedge P_3$
Line	$L = \pi_1 \wedge \pi_2$	$L^* = P_1 \wedge P_2 \wedge e_\infty$
Point Pair	$Pp = S_1 \wedge S_2 \wedge S_3$	$Pp^* = P_1 \wedge P_2$

Let us now compute the IPNS of e_0

$$e_0 \cdot P = 0. \tag{5}$$

which gives

$$e_0 \cdot (\mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0) = 0. \tag{6}$$

with the identity $e_0 \cdot e_\infty = -1$ this equals to

$$-\frac{1}{2}\mathbf{x}^2 = 0. \tag{7}$$

or

$$x_1^2 + x_2^2 + x_3^2 = 0 \tag{8}$$

describing only one point at the origin.

Let us look on what happens if we subtract some amount of infinity. What does, for instance, $e_0 - \frac{1}{2}r^2 e_\infty$ mean?

Let us compute the IPNS of the expression

$$(e_0 - \frac{1}{2}r^2 e_\infty) \cdot P = 0. \tag{9}$$

which equals to

$$-\frac{1}{2}\mathbf{x}^2 + \frac{1}{2}r^2 = 0. \tag{10}$$

or

$$x_1^2 + x_2^2 + x_3^2 - r^2 = 0 \tag{11}$$

describing all the points with the same distance r to the origin, namely a sphere at the origin.

A circle at the origin can be described by the intersection of a sphere at the origin with an origin plane $\mathbf{n}$.

$$\mathbf{n} \wedge (e_0 - \frac{1}{2}r^2 e_\infty) \tag{12}$$

or

$$\mathbf{n} \wedge e_0 - \frac{1}{2}r^2 \mathbf{n} \wedge e_\infty \tag{13}$$

We get a point pair at the origin with the help of an additional intersection with another origin plane $\mathbf{m}$.

$$\mathbf{n} \wedge \mathbf{m} \wedge (e_0 - \frac{1}{2}r^2 e_\infty) \tag{14}$$

or

$$\mathbf{n} \wedge \mathbf{m} \wedge e_0 - \frac{1}{2}r^2 \mathbf{n} \wedge \mathbf{m} \wedge e_\infty \tag{15}$$

A Point pair at the origin can also be described by the intersection of a line and the sphere as

$$(\mathbf{l}e_{123}) \wedge (e_0 - \frac{1}{2}r^2 e_\infty) \tag{16}$$

Please note, that a translation $T * E * (\tilde{T})$ of a conformal geometric object E by a versor T does not change the observations made above, and therefore allows for arbitrary objects in space.

4 CONFORMAL GEOMETRIC OBJECTS DESCRIBED BY BIVECTORS

In this section we derive alternative descriptions of objects in conformal geometric algebra, shown in table 2. The major advantage of these descriptions is that the characteristics of their underlying objects are indicated in their formula.

Table 2: Alternative descriptions of conformal geometric objects, with the feature of direct indication of object properties.

object	**description**
Line	$L = \mathbf{u}e_{123} + \mathbf{m} \wedge e_\infty$
Circle	$C = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0$ $+(\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) + \left[\frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty$
Oriented point	$O_p = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0$ $+(\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) + \left[\frac{1}{2}\mathbf{c}^2\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty$

4.1 Lines

It is well known that in Geometric Algebra lines can be expressed as

$$L = \mathbf{u}e_{123} + \mathbf{m} \wedge e_\infty, \tag{17}$$

with the 3D pseudoscalar $e_{123} = e_1 \wedge e_2 \wedge e_\infty$, the two 3D points $\mathbf{a}$, $\mathbf{b}$ on the line, $\mathbf{u} = \mathbf{b} - \mathbf{a}$ as 3D direction vector, and $\mathbf{m} = \mathbf{a} \times \mathbf{b}$ as the 3D moment vector (relative to origin). The corresponding six Plücker coordinates (components of $\mathbf{u}$ and $\mathbf{m}$) are (see Figure 2)

$$(\mathbf{u} : \mathbf{m}) = (u_1 : u_2 : u_3 : m_1 : m_2 : m_3). \tag{18}$$

The line L is normalized, if the vector $\mathbf{u}$ is normalized

$$L_{normalized} = \frac{\mathbf{u}e_{123} + \mathbf{m} \wedge e_\infty}{|\mathbf{u}|} \tag{19}$$

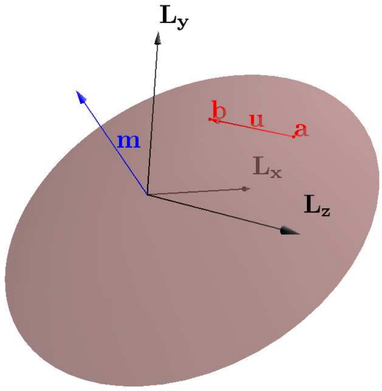


Figure 2: The line L through the 3D points $\mathbf{a}$, $\mathbf{b}$ and the visualization of its 6D Plücker parameters based on the two 3D vectors $\mathbf{u}$ and $\mathbf{m}$ of equation (18).

If only the algebraic expression for L is known, the normalized line can be computed according to the following procedure:
Since $L \cdot e_0$ results in the negative of $\mathbf{m}$

$$L \cdot e_0 = -\mathbf{m}, \tag{20}$$

$\mathbf{u}e_{123} = L - \mathbf{m} \wedge e_\infty$ can be computed as

$$\mathbf{u}e_{123} = L + (L \cdot e_0) \wedge e_\infty \tag{21}$$

or multiplied by e_{123} (Note that $e_{123}^2 = -1$)

$$\mathbf{u} = -(L + (L \cdot e_0) \wedge e_\infty)e_{123} \tag{22}$$

and the scaling factor for the normalization can be computed as follows:

$$|\mathbf{u}| = |(L + (L \cdot e_0) \wedge e_\infty)e_{123}| \tag{23}$$

Another alternative expression for lines is

$$L = \mathbf{d}e_{123} + (\mathbf{d} \times \mathbf{t}) \wedge e_\infty, \tag{24}$$

with the 3D pseudoscalar $e_{123} = e_1 \wedge e_2 \wedge e_3$, the normalized 3D direction vector $\mathbf{d}$ and one 3D point $\mathbf{t}$ of the line.

4.2 Circles and Oriented Points

In this subsection we introduce new formulas for circles and oriented points in CGA, from which their geometric meaning can be easily derived.

A circle can be described as the intersection of a plane and a sphere (normal $\mathbf{n_c}$) resulting in the following formula

$$C = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) \tag{25}$$

$$+ \left[\frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty.$$

Proof We start with the intersection of a plane and a sphere.
$C = \pi \wedge S = (\mathbf{n_c} + de_\infty) \wedge (\mathbf{c} + \frac{1}{2}(\mathbf{c}^2 - r^2)e_\infty + e_0)$
$C = \mathbf{n_c} \wedge \mathbf{c} + \frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} \wedge e_\infty + \mathbf{n}_c \wedge e_0 + de_\infty \wedge \mathbf{c} + \frac{1}{2}d(\mathbf{c}^2 - r^2)e_\infty \wedge e_\infty + d(e_\infty \wedge e_0)$

Now since $e_\infty \wedge e_\infty = 0$
$C = \mathbf{n_c} \wedge \mathbf{c} + \frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} \wedge e_\infty + \mathbf{n}_c \wedge e_0 + de_\infty \wedge \mathbf{c} + d(e_\infty \wedge e_0)$,

the plane equation gives $d = \mathbf{c} \cdot \mathbf{n_c}$
$C = \mathbf{n_c} \wedge \mathbf{c} + \frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} \wedge e_\infty + \mathbf{n}_c \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})e_\infty \wedge \mathbf{c} + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0)$,

Reordering and factoring leads to
$C = \mathbf{n_c} \wedge \mathbf{c} + \mathbf{n}_c \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) + \left[\frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty$.

The identity $\mathbf{a} \times \mathbf{b} = -(\mathbf{a} \wedge \mathbf{b})e_{123} = (\mathbf{b} \wedge \mathbf{a})e_{123}$ gives us
$C = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n}_c \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})(e_\infty \wedge e_0) + \left[\frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty$.

An oriented point is a circle with vanishing radius $r = \lim_{r_n \to 0} r_n$ and is defined by

$$O_p = (\mathbf{c} \times \mathbf{n_c})e_{123} + \mathbf{n_c} \wedge e_0 + (\mathbf{c} \cdot \mathbf{n_c})E + \left[\frac{1}{2}\mathbf{c}^2\mathbf{n_c} - (\mathbf{c} \cdot \mathbf{n_c})\mathbf{c}\right] \wedge e_\infty. \tag{26}$$

Most interestingly, the oriented point at the origin is $\mathbf{n} \wedge e_0$, which is the intersection of an origin plane with normal $\mathbf{n}$ and the point at the origin e_0.

Extracting point and normal from an oriented point. Another advantage of this formulation is the possibility to extract the normal $\mathbf{n_c}$ and the euclidian 3D point $\mathbf{c}$. Whilst this is trivial for the normal $\mathbf{n_c}$, it is slightly more complex for the point $\mathbf{c}$.

The normal $\mathbf{n_c}$ is contained in the statement $\mathbf{n_c} \wedge e_0$, and may be extracted by accessing the coefficients of blades $e_1 \wedge e_0$, $e_2 \wedge e_0$, and $e_3 \wedge e_0$ of the oriented point multivector O_p.

Once we have the normal $\mathbf{n_c}$, we may calculate the point $\mathbf{c}$ by the following formula. Note that the expressions $(\mathbf{c} \times \mathbf{n_c})$ and $(\mathbf{n_c} \cdot \mathbf{c}) = (\mathbf{c} \cdot \mathbf{n_c})$ are also contained as blade coefficients of the multivector O_p, so that the formula can directly be evaluated.

$$\mathbf{c} = \mathbf{n_c} \times (\mathbf{c} \times \mathbf{n_c}) + \mathbf{n_c}(\mathbf{n_c} \cdot \mathbf{c}) \tag{27}$$

Proof of equation 27. From linear algebra we know that the following equation holds for three arbitrary vectors **f**,**g**,**h**. It is a regularly used formula in physics and is called the *triple vector product*.

$$\mathbf{f} \times (\mathbf{g} \times \mathbf{h}) = \mathbf{g}(\mathbf{f} \cdot \mathbf{h}) - \mathbf{h}(\mathbf{f} \cdot \mathbf{g}) \tag{28}$$

We now insert $\mathbf{f} = \mathbf{n_c}$, $\mathbf{g} = \mathbf{c}$ and $\mathbf{h} = \mathbf{n_c}$ into equation 28.

$$\mathbf{n_c} \times (\mathbf{c} \times \mathbf{n_c}) = \mathbf{c}(\mathbf{n_c} \cdot \mathbf{n_c}) - \mathbf{n_c}(\mathbf{n_c} \cdot \mathbf{c}) \tag{29}$$

By analytically solving with respect to $\mathbf{c}$, we get

$$\mathbf{c} = \frac{\mathbf{n_c} \times (\mathbf{c} \times \mathbf{n_c}) + \mathbf{n_c}(\mathbf{n_c} \cdot \mathbf{c})}{\mathbf{n_c} \cdot \mathbf{n_c}}. \tag{30}$$

We further simplify this by using our knowledge that the normal $\mathbf{n_c}$ is normalized. It follows that $\mathbf{n_c} \cdot \mathbf{n_c} = 1$, which leads to the final result in equation 27.

5 APPLICATIONS

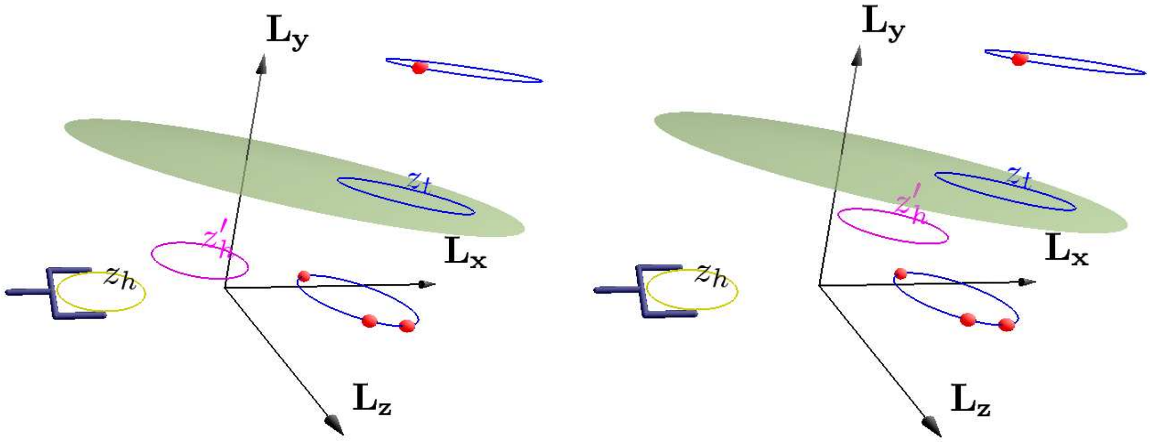
5.1 Inverse Kinematics in Robotics

The paper [7] showed many advantages of a formulation of an inverse kinematics algorithm in conformal geometric algebra. The algorithm controls a robot, performing a grasping operation on a target object. Once the object's position is located, an important part of the algorithm (see figure 5.1) is responsible for the movement of the robot gripper from its starting position to the target object. This movement is implemented as a versor-based transformation. A simpler formulation by an interpolation $\mathbf{z'_h}$ of oriented points $\mathbf{z_h}$ and $\mathbf{z_t}$

$$\mathbf{z'_h} = (1-t)\mathbf{z_h} + t\mathbf{z_t} \tag{31}$$

produces similar promising results, and is therefore subject to further research on this topic.

Figure 3: Movement of a robot gripper from the starting position z_h to the target object's position z_t. The current position is marked by z'_h



5.2 Virtual Camera Tracking Shots

In the field of cinematography (film) the term camera tracking shot means the movement of the recording camera along a trajectory with an alternating orientation. Several other properties like zoom or focal length may also vary along the trajectory, but we will focus on the translational and orientational camera properties.

There are several approaches [1] for this problem. The most popular one is to model the camera path by a parametric curve and to define the orientation seperately. A parametric curve is defined by a discrete ordered set of points in space on which it performs an interpolation with arbitrary differentiability. Orientation has to be treated differently in this approach, which introduces additional complexity.

An oriented point approach in comparison may simply perform a linear, quadratic or cubic interpolation on a discrete ordered set of oriented points, and may thereby simplify the formulation.

5.3 Modified Coons Patches

Coons Patches [3] are parametric surfaces in space defined between four given curves. A simplified approach is to define four points with corresponding normals in three-dimensional space and to construct a surface between them by interpolating. Since oriented points can be interpolated directly, they provide a simple framework for this simplified approach.

5.4 Molecular Dynamics

Molecular Dynamics (MD) is the simulation of a system of molecules according to known physical interaction rules, for a limited period of time. It is often refered to as a virtual microscope, which gives us a detailed virtual picture of the processes inside a large system of molecules. As molecules approximately behave like very small rigid bodies, the well known laws of rigid body motions can be applied to them. More detailed, each molecule has the properties of its position of its center of mass, its orientation, its linear and angular velocity, as well as its linear an angular acceleration.

Geometric Algebra has been applied to a Molecular Dynamics Simulation before in a different formulation. The existing approach expresses the molecule motions in terms of transformation versors (combined position and orientation), velocity screws (lin. and ang. velocity) and derivatives of velocity screws (lin. and ang. acceleration) [14].

A versor is a transformation, a screw is a differentiation of a transformation. Oriented points and their derivatives on the contrary, are simple geometric concepts. Both have complicated mathematical formulations, but an oriented point and its derivatives are quite simple to understand from a geometrical perspective in comparison to a versor and its derivatives.

In terms of runtime performance both yield a very similar performance, because both lead to the same arithmetic instructions when broken down by an optimizing compiler like Gaalop [11].

6 IMPLEMENTATIONAL ASPECTS

The paper [9] showed for the first time that implementations of geometric algebra algorithms can be faster than conventional ones. This is due to optimization approaches like Gaalop [10] and Gaigen [5], which are powerful tools making it easy to implement performant algorithms in CGA.

In the meantime, there is even additional potential for runtime improvements in the advancing field of General Purpose GPU computing, and with new programming languages like OpenCL [12]. Algebraically, lines, circles, and oriented points, are 10-dimensional bivectors, that can be handled in parallel using the 16-dimensional vector data types OpenCL is providing.

Gaalop Compiler Driver (GCD [2]) applies the compiler driver concept to C++/CUDA and OpenCL programs, simplifying the optimized integration of GA-based algorithms into programming toolchains. It invocates Gaalop in the background and thus also profits from all its advantages.

REFERENCES

[1] Jules Bloomenthal. Graphics gems. chapter Calculation of reference frames along a space curve, pages 567–571. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[2] Patrick Charrier and Dietmar Hildenbrand. Gaalop compiler driver. In *proceedings of the GraVisMa workshop, Brno*, 2010.

[3] S. A. Coons. Surfaces for computer-aided design of space forms. Technical report, Cambridge, MA, USA, 1967.

[4] Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric Algebra for Computer Science, An Object-Oriented Approach to Geometry*. Morgan Kaufman, 2007.

[5] Daniel Fontijne, Tim Bouma, and Leo Dorst. Gaigen 2: A geometric algebra implementation generator. Available at `http://staff.science.uva.nl/~fontijne/gaigen2.html`, 2007.

[6] Dietmar Hildenbrand. Geometric computing in computer graphics using conformal geometric algebra. *Computers & Graphics*, 29(5):802–810, 2005.

[7] Dietmar Hildenbrand, Eduardo Bayro-Corrochano, and Julio Zamora. Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra. In *Advances in Applied Clifford Algebras*. Springer, 2008.

[8] Dietmar Hildenbrand, Daniel Fontijne, Christian Perwass, and Leo Dorst. Tutorial geometric algebra and its application to computer graphics. In *Eurographics conference Grenoble*, 2004.

[9] Dietmar Hildenbrand, Daniel Fontijne, Yusheng Wang, Marc Alexa, and Leo Dorst. Competitive runtime performance for inverse kinematics algorithms using conformal geometric algebra. In *Eurographics conference Vienna*, 2006.

[10] Dietmar Hildenbrand and Joachim Pitt. The Gaalop home page. Available at `http://www.gaalop.de`, 2010.

[11] Dietmar Hildenbrand, Joachim Pitt, and Andreas Koch. *Gaalop - High Performance Parallel Computing based on Conformal Geometric Algebra*, volume 1, pages 350–358. Springer, May 2010.

[12] Khronos-Group. The OpenCL home page. Available at `http://www.khronos.org/opencl/`, 2009.

[13] Christian Perwass. *Geometric Algebra with Applications in Engineering*. Springer, 2009.

[14] Florian Seybold, Patrick Charrier, Dietmar Hildenbrand, M. Bernreuther, and D. Jenz. Runtime performance of a molecular dynamics model using conformal geometric algebra. Slides available at `http://www.science.uva.nl/~leo/agacse2010/talks_world/Seybold.pdf`, 2010.