Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090116

Ortogonal Approach for Haptic Rendering Algorithm based in Conformal Geometric Algebra

*Gabriel Sepúlveda-Cervantes** *and Edgar A. Portilla-Flores*

Instituto Politécnico Nacional, CIDETEC, D.F, México

Received: 16 Mar. 2014, Revised: 16 Jun. 2014, Accepted: 17 Jun. 2014
Published online: 1 Jan. 2015

Abstract: This work presents a novel method for haptic rendering contact force and surface properties for virtual objects using the Conformal Geometric Algebra orthogonal decomposition approach. The mathematical representation of geometric primitives along with collision algorithms based on its mathematical properties is presented. The orthogonal decomposition of contact and interaction forces is achieved using the same framework and dynamic properties in both subspaces are rendered simultaneously. Comparing with vector calculus, the Conformal Geometric Algebra (CGA) approach provides an easier and more intuitive way to deal with haptic rendering problems due to its inner properties and a simpler representation of geometric objects and linear transformation. The results of the evaluation of the method using a 3 DOF haptic device are presented.

Keywords: Haptics; Geometric Algebra; Haptic Rendering

1 Introduction

Haptic rendering is a computational process by which desired kinesthetic or tactile stimuli are imposed on the user within a human-machine interaction system [2]. In other words, haptic rendering techniques provide different ways to represent object's physical attributes like shape, elasticity, texture, and so on [16]. This representation is just as the visual rendering of virtual objects, there exist many different visual rendering techniques in order to generate by computational calculus an image so the user eyes could perceive and then interpret as a 3D sphere, a building, or a nice character. In the same way the haptic rendering process generates by computational calculus a force-response model so a haptic device provides the user with a kinesthetic or tactile stimuli that the user interprets as touching or grabbing a virtual or remote object.

The main steps in haptic rendering algorithms are: Collision detection, Force-response and Control algorithms. The collision detection algorithms provide information about contacts occurring between the user and the virtual object. Force-response algorithms provides a way to establish a force interaction between the user and the virtual object. The control algorithms are the tool by which the force calculated in the previous algorithms is exerted on the user by the real electromechanical haptic device.

Many different approaches have been proposed, from Ideal Haptic Interface Point (IHIP) where the algorithm checks to see if the end point of the haptic device is inside a virtual object, then a virtual point is calculated at the collision point and the penetration depth is calculated by the difference between this point and the current position of the haptic device [16]. Another approach presented in [17] is based in computing collision and contact force over virtual objects defined by implicit functions. Using orthogonal decomposition for decomposing the haptic device dynamics is an approach presented in [18] for suturing task.

The algorithms mentioned above along others use different kind of mathematical framework for collision detection, force-response calculation and control algorithms, e.g. for collision detection and implicit function is evaluated then the force-response model is calculated using vector algebra and control algorithm takes use of linear algebra. In this work a new approach based in Geometric Algebra is presented that applies the same mathematical framework to collision detection and force-response modelling while keeping the algorithm intuitive straight forward and enabling the haptic

* Corresponding author e-mail: gsepulvedac@ipn.mx

rendering simultaneously normal and tangent properties for virtual objects.

In [1] the basic concept of a hapting rendering algorithm based on Geometric Algebra framework is presented, but it lacks a methodology for implementation using Geometric Algebra concepts, also a practical implementation and experimental results are not presented.

2 Geometric Algebra

Clifford and Grassman algebras was the base for developing Geometric Algebra in 1984 by David Hestenes [4]. Geometric Algebra (GA) unify, simplify and generalize many areas of the mathematics involving geometric concepts [5]. A brief history of GA is presented in [6] and its applications in Special Relativity and Quantum Mechanics. The applications of GA concepts are wide, from electromagnetism in antennas [7], visual computing [8], parallel graphic computing [9], features extraction [10] and robotics [11].

2.1 Geometric Algebra Basics

The basic concepts of GA are found in [5], [4], [6], [12], among others. In this section just the necessary concepts needed for haptic rendering and collision algorithms development are presented.

Let G^n denote the algebra of n-dimensions which is a graded-linear space. This algebra is an extension of the inner product space $\Re^n$. First it is an associative algebra, that is, a vector space with a defined product, called geometric product, that satisfies the following properties P1-P6, for all scalars a and $A,B,C \in G^n$:

P1: A(B+C) = AB + AC, (B+C)A = BA + CA.
P2: (aA)B = A(aB) = a(AB).
P3: (AB)C = A(BC).
P4: 1A = A1 = A

The members of G^n are called *multivectors*, this members are a new mathematical entity capable of representing geometric objects. This could be interpreted by the following sentences: in simple algebra, a literal a could represent any amount or a given or unknown quantity, in the same way different members $A,B,C \in G^n$ could represent a geometric objects like points, vector, lines, planes, circles and spheres. Two more properties are needed to construct the Geometric Algebra [5], [13]:

P5: The geometric product is linked to the algebraic structure of $\Re^n$ by

$$uu = u \cdot u = \|u\|^2, \forall u \in \Re^n \tag{1}$$

P6: Every orthonormal basis for $\Re^n$ determines a *standard basis* for the vector space G^n

If u and v are orthogonal, then the non-commutative property arise as follows:

$$vu = -uv, u \perp v \tag{2}$$

This geometric product of orthogonal basis are so called n-vectors, and spans the standard basis for a given GA. Most geometric CGA-elements consist of the independent base elements that have the same number of base vectors. A *blade B* for a k-dimensional subspaces of $\Re^n$ is a product of an orthogonal basis for the subspace: $B = b_1 b_2 \cdots b_k$, and is called a blade of *grade k*.

One of the most important features of this mathematical approach to the field of haptics derives in the following capabilities:

*It is possible to represent geometric objects in $\Re^n$ with members of G^n.
*Geometric Algebra represents geometric operations on these objects with algebraic operations.
*Coordinates are not used in these representation

In the approach presented in this work it is specify the geometric algebra G^n of the n dimensional space by $G^{p,q,r}$, where p, q and r stand for the number of basis vectors which squares to 1, -1, 0 respectively and fulfils $n = p + q + r$.

Denoting e_i as the basis vector i in GA $G^{p,q,r}$, the geometric product of two basis vectors is defined as:

$$e_i e_j = \begin{cases} 1 & for\ i = j \in 1,\ldots,p \\ -1 & for\ i = j \in p+1,\ldots,p+q \\ 0 & for\ i = j \in p+q+1,\ldots,p+q+r \\ e_i \wedge e_j & for\ i \neq j \end{cases} \tag{3}$$

The standard basis for the entire algebra is denoted by:

$$\{1\},\{e_i\},\{e_i \wedge e_j\},\{e_i \wedge e_j \wedge e_k\},\ldots,\{e_i \wedge e_j \wedge e_k \wedge \ldots \wedge e_n\} \tag{4}$$

with $i \neq j \neq k$.

The inner and outer product of vector algebra are related to the geometric product as follows:

$$uv = u \cdot v + u \wedge v \tag{5}$$

The geometric product of two vectors is the sum of a scalar and a bivector. There is no formulation in vector algebra that the inner and outer product are part of a whole: the geometric product is an associative product in which nonzero vector have an inverse.

3 Conformal Geometric Algebra

Geometric Algebra $G^{n,1,0} = G^{n,1}$ is a powerful GA for Euclidean geometry. This model also unifies Euclidean, hyperbolic and elliptic geometries. The Conformal Geometric Algebra (CGA) extends $\Re^d$ (the common cartesian space) with vector e_+ and e_-, orthogonal to $\Re^d$

and satisfying $e_+^2 = 1, e_-^2 = -1$. The basis e_+, e_- are used to generate the more useful basis:

$$e_0 = \frac{1}{\sqrt{2}}(e_- - e_+) \tag{6}$$

$$e_\infty = \frac{1}{\sqrt{2}}(e_- + e_+) \tag{7}$$

Then based on (3) it leads to:

$$e_0^2 = e_\infty^2 = 0 \tag{8}$$

$$e_0 \cdot e_\infty = -1 \tag{9}$$

The basic geometric elements are classified in 4 types:

1. Rounds: includes points, point pairs, circles, spheres and n-spheres.
2. Flats: a flat is a round containing the point at infinity e_∞ and represents, lines, planes and hyper-planes.
3. Free blades: this are elements without position. This elements have no e_0 in its formula and can be considered as "directions".
4. Tangent blades: this elements have no e_∞ component and can be considered as "tangent subspaces".

The geometric properties of $\Re^3$ could be treated with CGA $G^{4,1}$, in this work this approach is used without losing generality. The vectors e_1, e_2, e_3 are orthonormal vectors in Euclidian space $\Re^3$.

3.1 Point

In CGA a point can be represented by a round or a flat, or it can be seen as a sphere with radius 0. The conformal model representation of a point at the end of vector $p \in \Re^3$ is:

$$p_c := p + e_0 + \frac{1}{2}p^2 e_\infty \in G^{4,1} \tag{10}$$

The flat point F corresponding to the same point is:

$$F = (p + e_0) \wedge e_\infty \tag{11}$$

Note that the vector is normalized and the coefficient of e_0 is 1, the multivector p is null, this is: $p^2 = p \cdot p = 0$ and more generally

$$p \cdot q = -\frac{1}{2}(p - q)^2 \tag{12}$$

Round points and flat points could be multiply by an arbitrary scalar non-zero factor and it represents the same point.

3.2 Point pair, circle and sphere

Point pair, circles, spheres and hyperspheres are rounds with grade 2, 3, 4 and above respectively. A point pair is a sphere in 1D, which means the set of points in a line having the same distance to a point called center. Point pair also represents a line segment from P to Q, see fig. 1. The CGA representation of a point pair is:

$$B_2 := P \wedge Q \tag{13}$$

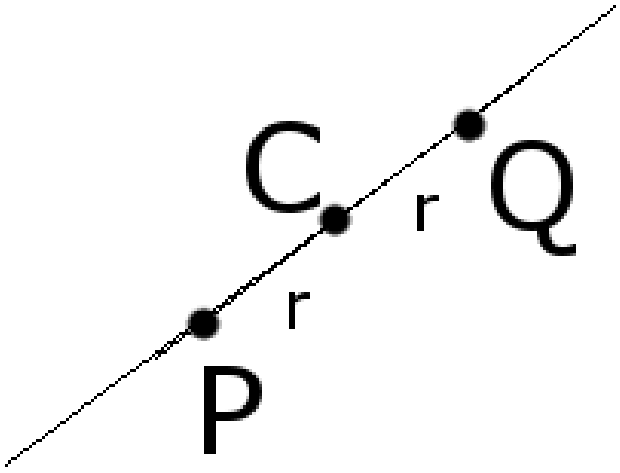


Fig. 1: CGA representation of a Point Pair, also the representation of 1D sphere. P and Q are at the same distance r from the center C

The CGA representation of a circle, sphere and hypersphere are presented below:

$$C := P \wedge Q \wedge R \tag{14}$$

$$B_3 := P \wedge Q \wedge R \wedge S \tag{15}$$

$$B_n := P \wedge Q \wedge R \wedge S \wedge \cdots \wedge N \tag{16}$$

3.3 Line plane and hyperplane

A straight line in CGA is represented by a 3 dimensional blade:

$$L := P \wedge Q \wedge e_\infty, P, Q \in G^{4,1} \tag{17}$$

where P, Q are round points along the line.

The flats in CGA represents planes and hyperplanes. A 3D-plane is a blade with grade 4, and generally a dD-hyperplane has a grade $d+1$.

$$\Pi := P \wedge Q \wedge S \wedge e_\infty, P, Q, S \in G^{4,1} \tag{18}$$

where P, Q, S are round points.

An example of a representation of geometric objects in $\Re^3$ using CGA is presented in Figure 2, e_1, e_2, e_3 stands for the orthonormal basis, the points are $p_1 = e_1 + e_0 + \frac{1}{2}e_\infty$, $p_2 = e_2 + e_0 + \frac{1}{2}e_\infty$, $p_3 = e_3 + e_0 + \frac{1}{2}e_\infty$, $p_4 = -e_1 + e_0 + \frac{1}{2}e_\infty$. The line is defined $L = p_2 \wedge p_4 \wedge e_\infty$, the circle is defined $B2 = p_1 \wedge p_2 \wedge p_3$ and the sphere $B3 = p_1 \wedge p_2 \wedge p_3 \wedge p_4$.

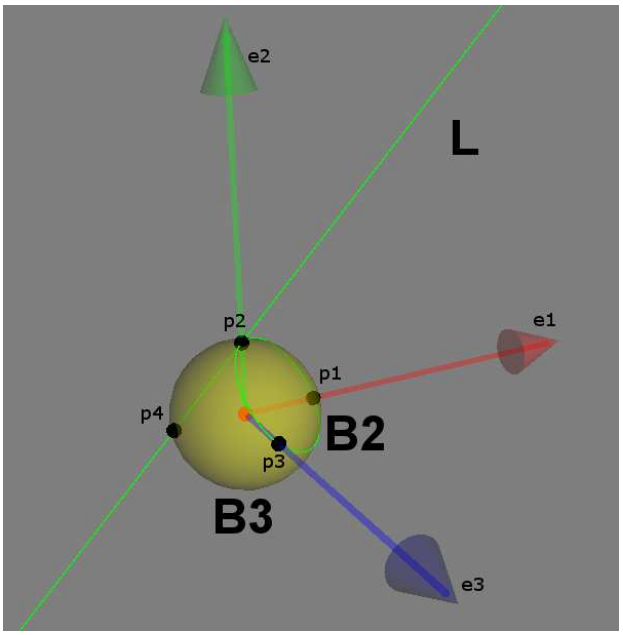


Fig. 2: CGA visual representation of a $\Re^3$ space. The line L passes through points $p2$ and $p4$, the circle $B2$ passes through points p_1, p_2, p_3, and the sphere $B3$ passes through points p_1, p_2, p_3, p_4.

An excellent tool for introducing and visual understanding CGA is *GAViewer* [15]. It is an application specialized in visualizing 3D objects represented by CGA. Figure 2 was made using this tool.

3.4 CGA Pseudoscalar and Dual

An important member of CGA is the *pseudoscalar* I and is defined by:

$$I = e_1 e_2 \cdots e_n \tag{19}$$

Note that $I^{-1} = e_n \cdots e_2 e_1$ and in CGA model the product $II^{-1} = 1$. This pseudo scalar is used to calculate the dual of a blade in CGA. Given a blade B its dual B^* is given by:

$$B^* = BI^{-1} \tag{20}$$

The dual B^* in CGA represents the same geometric object, if B has grade r in a $d-d$*imensional* CGA, then the dual B^* has grade $d-r$. An example of this is the duality between a plane in $\Re^3$ and its normal vector, the space is 3-dimensional, the plane is 2-dimensional and the normal vector is 1-dimensional.

3.5 Parametric Representation of Sphere and Plane

The particular case of representing a 3D sphere and plane in Conformal Geometric Algebra could be understand as an equivalent to the conventional algebra formulation using the dual B_3^* and Π^*. It is presented in [5] the dual representation of the sphere and plane as:

$$B_3^* := C_B - \frac{1}{2}\rho^2 e_\infty \tag{21}$$

$$\Pi^* := n + (p \cdot n)e_\infty \tag{22}$$

where C_B, ρ stands for the sphere center and radius respectively, and n, p stand for the normal of the plane and a point on the plane respectively.

3.6 Orthogonal Decomposition

Due that the CGA model represents geometric operations in $\Re^3$ with algebraic operations in $G^{4,1}$, the orthogonal decomposition of a vector a with respect to a blade B is given by the following operations:

$$a = a_\parallel + a_\perp \tag{23}$$

$$a_\parallel = (a \cdot B)B^{-1} \tag{24}$$

$$a_\perp = (a \wedge B)B^{-1} \tag{25}$$

where $a_\parallel, a_\perp$ represents the normal and tangent component of vector a respect to the blade B, note that B could represent a vector, plane, sphere, or any blade in CGA, see Figure 6. This simple procedure for orthogonal decomposition has no analogy in the normal vector algebra, except when B is another vector in $\Re^3$.

4 Haptic Rendering

Haptic devices provides a tool in virtual reality systems to interact with 3D virtual environments using user's touch, tactile and kinesthetic capabilities.

The interaction with virtual objects is achieved using haptic rendering algorithms. Typically, a haptic rendering algorithm is made of two parts: collision detection and collision response [16]. The user manipulates the haptic device in order to interact with a virtual environment next by acquiring the position and orientation of the device a collision within the virtual environment is calculate. After

the collision detection the interaction forces must be computed in order to provide the user with a congruent response of the interaction with the 3D objects and their surface details. Hence, a haptic loop with frequency above 1 KHz must be implemented, this due to the human perception bandwidth. The haptic rendering algorithm could provide the user with virtual object's information about its *Object Properties* and *Surface properties*. The object properties represents the internal static or dynamic behavior, like hardness, stiffness, weight or shape, internal viscosity, etc., the surface properties represents the surface texture, tangent friction, etc. The steps of a common haptic rendering algorithm are shown in Figure 3.

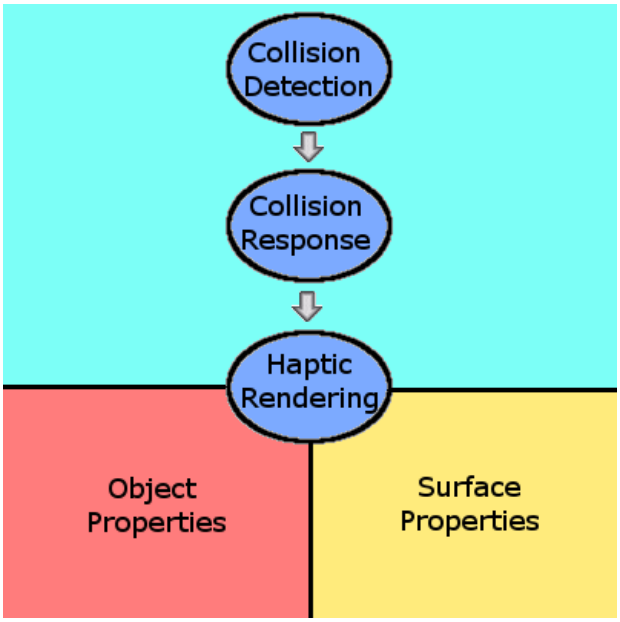


Fig. 3: Haptic rendering algorithm steps

The work in [19] presents six procedures for haptic exploration:

H1: Contour following: **global shape, exact shape**
H2: Pressure: **hardness**
H3: Unsupported holding: **weight**
H4: Lateral motion: **texture**
H5: Static contact: **temperature**
H6: Enclosure: **global shape, volume**

One point interaction haptic device, like Novint Falcon, Sensable Phantom or Delta, could provide H1-H4 haptic perception of virtual objects. H6 is achieved using multipoint devices like CyberGrasp. Temperature is still a virtual object property still not included in virtual reality simulators.

4.1 CGA Collision Detection

The first step in haptic rendering algorithms is to detect a collision of the 3D representation of the haptic device's probe called *proxy* and the virtual objects. A variety of works about 3D collision in dynamic environments has been done [20], [17], [16], [22]. For haptic interaction an interesting approach to collision detection is presented in [17]. This approach is based on implicit functions that represents the contour of virtual objects, this provides with an easy inside-outside test for collision test.

An example using the implicit function approach is the collision with a sphere of radius r and center at the point $C \in \Re^3 = he_1 + ke_2 + le_3$. Given the following sphere:

$$\varphi_1 = (x-h)^2 + (y-k)^2 + (z-l)^2 - r^2 \tag{26}$$

The equation (26) represents the manifold of all points $X \in \Re^3 | \varphi_1 = 0$, see Figure 4. The evaluation of equation (26) using the position of the proxy gives three possible results: $\varphi_1 > 0$ the proxy is outside the virtual object, $\varphi_1 = 0$ the proxy is on the virtual object's surface and $\varphi_1 < 0$ the proxy is inside the virtual object.

Using CGA representation of a sphere (equation (15)) the inside-on-outside test could be done using the position p of the proxy in the following equation:

$$\varphi_{1_{CGA}} = p \cdot B_3^* \tag{27}$$

then in the same way, $\varphi_{1_{CGA}} > 0$ the proxy is outside the virtual object, $\varphi_{1_{CGA}} = 0$ the proxy is on the virtual object's surface and $\varphi_{1_{CGA}} < 0$ the proxy is inside the virtual object. The same is also true for a plane. Using the magnitude value of this operation, it could be tested inside-outside of a point with a line, it is due to the line is a 2 dimensional object so a point just has two possible states (in/out):

$$\varphi_{2_{CGA}} = |p \cdot L^*| \tag{28}$$
$$\varphi_{3_{CGA}} = p \cdot \Pi^* \tag{29}$$

It is important to denote that B_3^* and Π^* are 4-blades and L^* is a 3-blade, so the sphere B_3^* and the plane Π^* subdivide the space in 3 regions (in-on-out) and the line L^* subdivide it in 2 (in-out).

4.2 Orthogonal Decomposition Approach

The step after the collision detection is the collision response. In this step the interaction forces send to the haptic device are calculated. This forces provides the user with a stimuli that represents the objects normal and tangent properties that stands for the haptic exploration procedures H1-H3 and H4 respectively, see Figure 5.

A complete force-response model to provide virtual objects with normal and tangent properties simultaneously using CGA is:

$$F = F_n + F_t \tag{30}$$

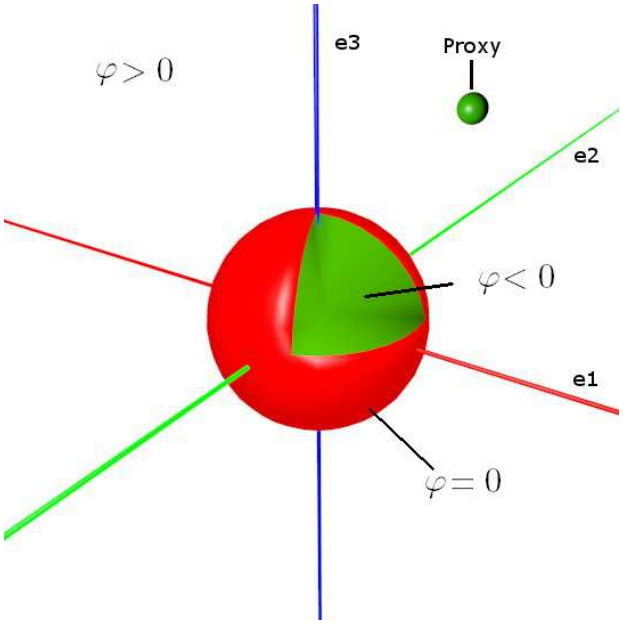


Fig. 4: Representation of a sphere with center at the origin. In vector algebra the implicit function representation is given by (26). In CGA the representation is given by (15).

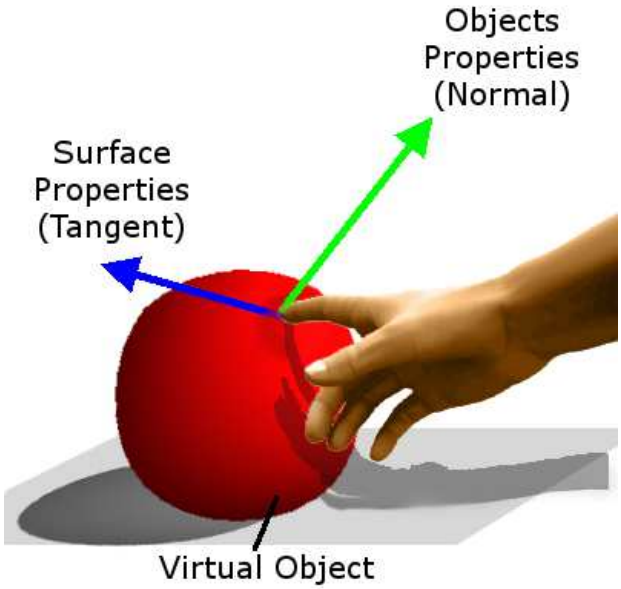


Fig. 5: Representation of virtual object's normal (procedures H1-H3) and tangent properties (procedure H4).

where F is the total force exerted by the haptic device, F_n is the force-response due to normal interaction with the virtual object (H1-H3) and F_t is the force-response due to the interaction with the object surface (H4).

4.3 CGA Normal Properties

Most of the common interaction with 3D virtual objects could be simplified to the interaction with geometric primitives like, points, lines, planes and spheres, eg. the contour of a 3D virtual object cold be represented by a triangle mesh. Then the interaction with a triangle mesh could be reduce to interact with each of its triangles and the interaction with a triangle could be reduce to the interaction with a plane defined by the triangle vertex represented in CGA by (29), see Fig 6, this procedure along with a force-response model accomplish the contour following task (procedure H1).

In order to provide the virtual object with hardness (procedure H2), a force-response model should be executed by the haptic device. To achieve this the variables φ_n and $\dot{\varphi}_n$ are defined as follows:

$$\varphi_{plane} = \frac{x \cdot \Pi}{|\Pi|} \tag{31}$$

$$\dot{\varphi}_{plane} = \frac{\dot{x} \cdot \Pi}{|\Pi|} \tag{32}$$

$$\varphi_{sphere} = |(x \wedge C) \cdot e_\infty| - \rho \tag{33}$$

$$\dot{\varphi}_{sphere} = |(\dot{x} \wedge C) \cdot e_\infty| - \rho \tag{34}$$

where $x, \dot{x}$ stands for the position and velocity of the proxy within the virtual environment, Π is the CGA representation of a plane, equation (18), C, ρ stands for the sphere center in conformal model and the sphere radius respectively. Equation (31-34) are mathematical representation of the interaction with a geometric plane and sphere in $\Re^3$ and its change over time, using CGA, this representation is useful for haptic rendering algorithms.

The simplest model to implement hardness to a virtual object is to apply a linear proportional force dependant of the depth penetration, this is called an elastic model [16], this model is given by:

$$F = k\,|\,\Delta x\,|\,\hat{n} \tag{35}$$

where F is the force exerted by the haptic device, k stands for the elastic coefficient of the virtual object, and $|\,\Delta x\,|$ stands for the depth penetration and $\hat{n}$ is the normal direction respect to the object's surface.

Using CGA representation of geometric primitives a visco-elastic force-response model could be implemented using:

$$F_n = K_n \varphi_n + B_n \dot{\varphi}_n \tag{36}$$

where F_n stands for the force calculated for the normal interaction, K_n and B_n stands for the objects stiffness and viscosity respectively. The sub index n is added because this force is exerted for an interaction in a direction normal to the virtual object surface. The variable $\varphi_n, \dot{\varphi}_n$ could be substituted by equations (31, 32) respectively for haptic render a plane or by equations (33, 34) for a sphere

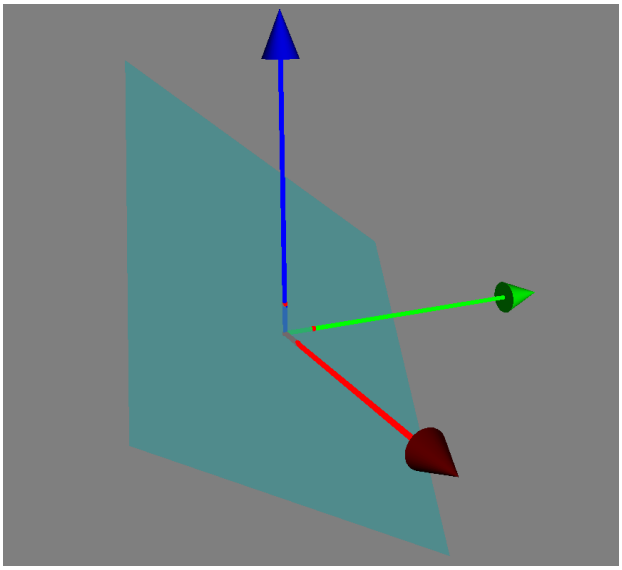


Fig. 6: A triangle in $\Re^3$ is defined by three points P,Q,R, using this points and equation (29) it is defined the virtual object representation of a plane Π in CGA. The vector V is orthogonal decompose in its normal component $V_{\perp}$ and tangent component $V_{\parallel}$ respect to plane Π.

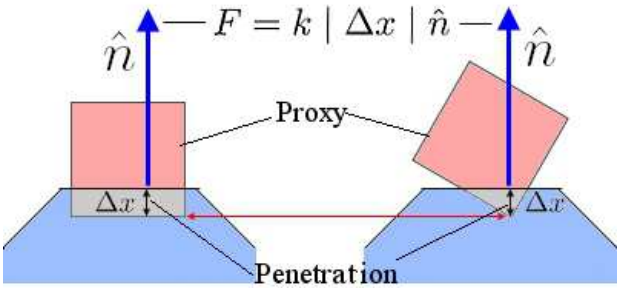


Fig. 7: Proxy's penetration depth used to calculate linear elastic haptic rendering.

with dynamic properties. Note that $\varphi_n = 0$ represents the contour of the geometric object.

The normal vector to the object surface represented in equation (31, 33) could be calculated as follows:

$$\hat{n}_{plane} = \frac{\Pi^*}{\mid \Pi \mid} \tag{37}$$

$$\hat{n}_{sphere} = \frac{x - C}{\mid x - C \mid} \tag{38}$$

4.4 CGA Tangent Properties

In order to haptic render surface properties for virtual objects using the CGA representation, a tangent dynamic respect to the object's surface must be calculated. The CGA representation of the tangent interaction with (31) and (33) are given by:

$$\xi_{plane} = (x \wedge \hat{n}_{plane})\hat{n}^*_{plane} \tag{39}$$

$$\dot{\xi}_{plane} = (\dot{x} \wedge \hat{n}_{plane})\hat{n}^*_{plane} \tag{40}$$

$$\xi_{sphere} = (x \wedge \hat{n}_{sphere})\hat{n}^*_{sphere} \tag{41}$$

$$\dot{\xi}_{sphere} = (\dot{x} \wedge \hat{n}_{sphere})\hat{n}^*_{sphere} \tag{42}$$

where $x, \dot{x}$ stands for the position and velocity of the proxy within the virtual environment. The variables $\xi, \dot{\xi}$, represents the tangent displacement and tangent velocity of the proxy respect to the object surface. Equations (39-42) are used to provide the virtual object with surface properties, procedure H4.

In order to provide the object with surface friction a force-response model is defined as follows:

$$F_t = B_t \dot{\xi}_t \tag{43}$$

where F_t stands for the force calculated for the tangent interaction,B_n stands for the objects surface friction coefficient. The sub index t is added because this force is exerted for an interaction in a direction tangent to the virtual object surface. The variable $\dot{\xi}_t$ could be substituted by equation (40) for haptic render a plane's surface or by equations (42) for a sphere's surface with linear friction.

Both the normal and tangent force-response is applied by the haptic device while the proxy is in contact with the virtual object, that is while $\varphi_n < 0$.

4.5 Combined algorithm

To provide virtual objects with surface texture a combined force-response model could be applied. First the collision calculation is modified in order to provide the object surface with bumps and irregularities, this is achieved by using:

$$\varphi_n - A sin\,(2\pi f \mid \xi_t \mid) = 0 \tag{44}$$

where A stands for the depth of the object surface bumps, f is a variable that determines the object's bumpiness. The force-response model applying this behavior is as follows:

$$F = K_n\,(\varphi_n - A sin\,(2\pi f \mid \xi_t \mid)) + B_n \dot{\varphi}_n + B_t sign(\dot{\xi}_t) \tag{45}$$

The equation (45) is a force-response model that provides the virtual object with visco-elastic behavior and dry surface friction with bumps over objects surface.

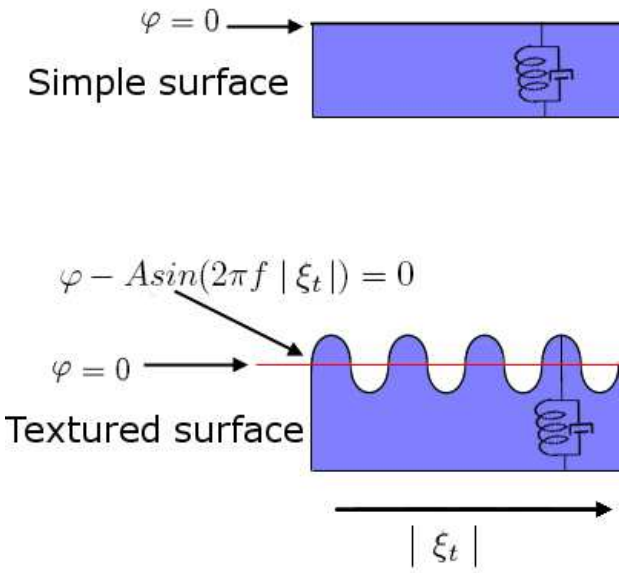


Fig. 8: Combined algorithm using normal and tangent variables to haptic render bump surface objects as textured surface objects

5 Experimental Platform and Results

The experimental platform is integrated by a Laptop Asus G53J, running at 1.73 GHz with Intel i7 processor, 8Gb RAM and 1.5 Gb Nvidia GeForce GTX460m video card. A Novint Falcon haptic device is used, see Figure 9. The application was built in Visual Studio 2012, with Gaigen 2.5 Test Suite [14] implementing CGA classes and methods and Novint Falcon SDK over Visual C++.

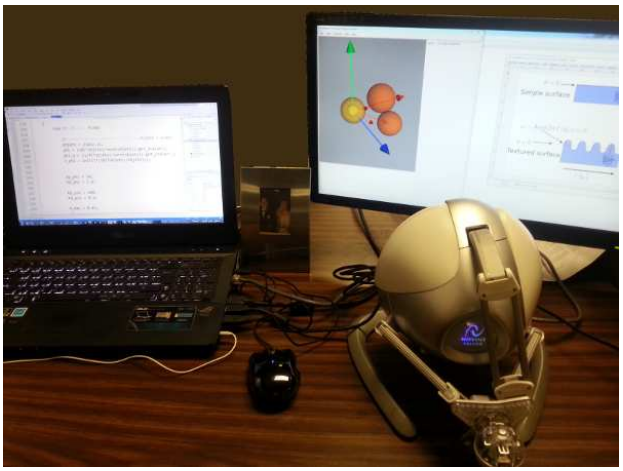


Fig. 9: Experimental platform

Table 1: Experiments Setup

Object	Normal Model	Tangent Model
Plane	$K_n\varphi_n + B_n\dot{\varphi}_n$	$B_t\dot{\xi}_t$
Sphere	$K_n\varphi_n + B_n\dot{\varphi}_n$	$B_t sign(\dot{\xi}_t)$
Plane	$K_n(\varphi_n - Asin(2\pi f \mid \xi_t \mid)) + B_n\dot{\varphi}_n$	$B_t sign(\dot{\xi}_t)$

In order to test the proposed haptic rendering algorithm, three experiments were conducted. The information of each experiment: a) virtual object type, b) normal force-model and c) tangent force-model are presented in the Table 1.

The interaction variables: $x, \dot{x}\varphi_n, \dot{\varphi}_n, \xi_t, \dot{\xi}_t, F$ and the 3D movement of the proxy for each of the three experiments are presented in Figures 10-14.

In Figure 10 and 11, it could be seen the collision detection around $t = 2.7sec$ when the variable φ_n change from greater than zero to lower than zero, also the linear dynamic dependence of the reaction force is presented in 10C and 11C. The linear surface friction is presented in 11A and 11C.

The Figures 12 and 13 present the dynamic behavior during the interaction with a sphere, the force calculated for the haptic device has the sum of both normal and tangent interaction.

In Figure 14 and 15, it could be seen the interaction with a bumped floor Fig 15D shows the bumps when moving in to directions over the textured plane.

6 Conclusion

A new haptic rendering algorithm base in Conformal Geometric Algebra is presented with a stable interaction on a 1KHz haptic thread. Different parts of the methodology provide collision detection, force-response model and virtual object surface properties, simultaneously without multiple threads running at a time. This new approach permits the orthogonal decomposition of the interaction forces using simple algebraic operations given two haptic rendering spaces, normal and tangent. The methodology could be extended to more complex dynamic properties for normal and surface behaviors, depending on positions and velocities.

The preliminary results indicate that this approach is programmer friendly and intuitive, the inner structure of Conformal Geometric Algebra simplifies the geometric analysis of the interaction. The experiments shows the possibility to couple or decouple normal and tangent dynamics.

Future work will include the introduction of bio-mechanic models of tissue for normal interaction and complex friction models for tangent interaction, the extension to orientation of the haptic device probe and multipoint interaction.

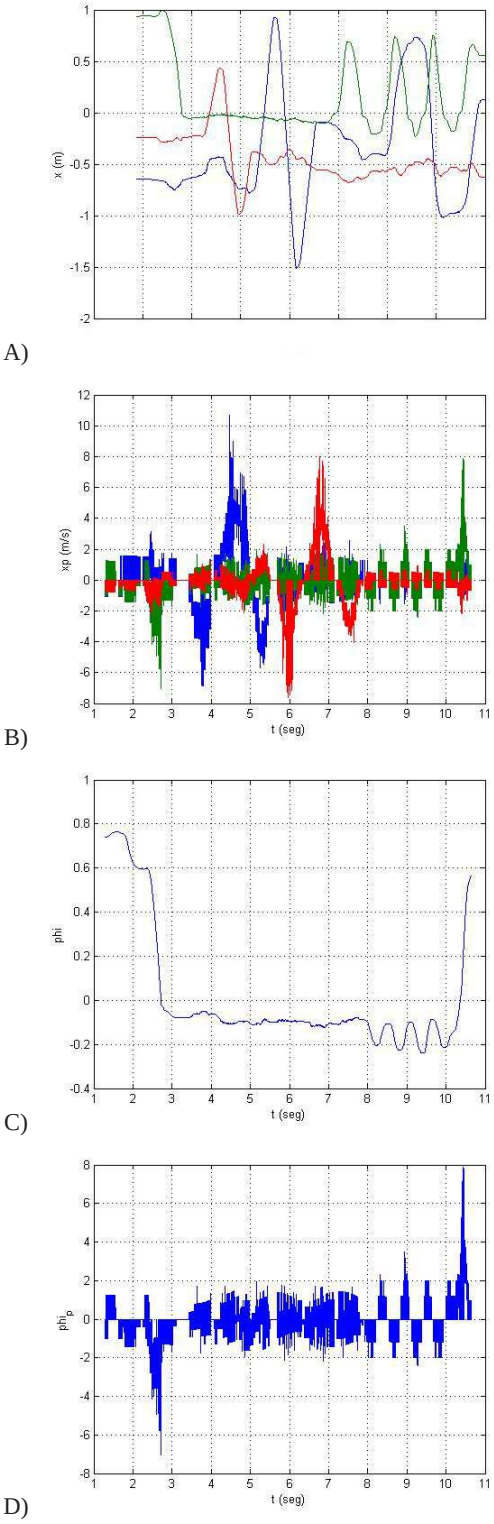


Fig. 10: Interaction with a plane with visco-elastic properties and surface linear friction. A) Proxy position, B) Proxy velocity, C) φ_n, D) $\dot{\varphi}_n$

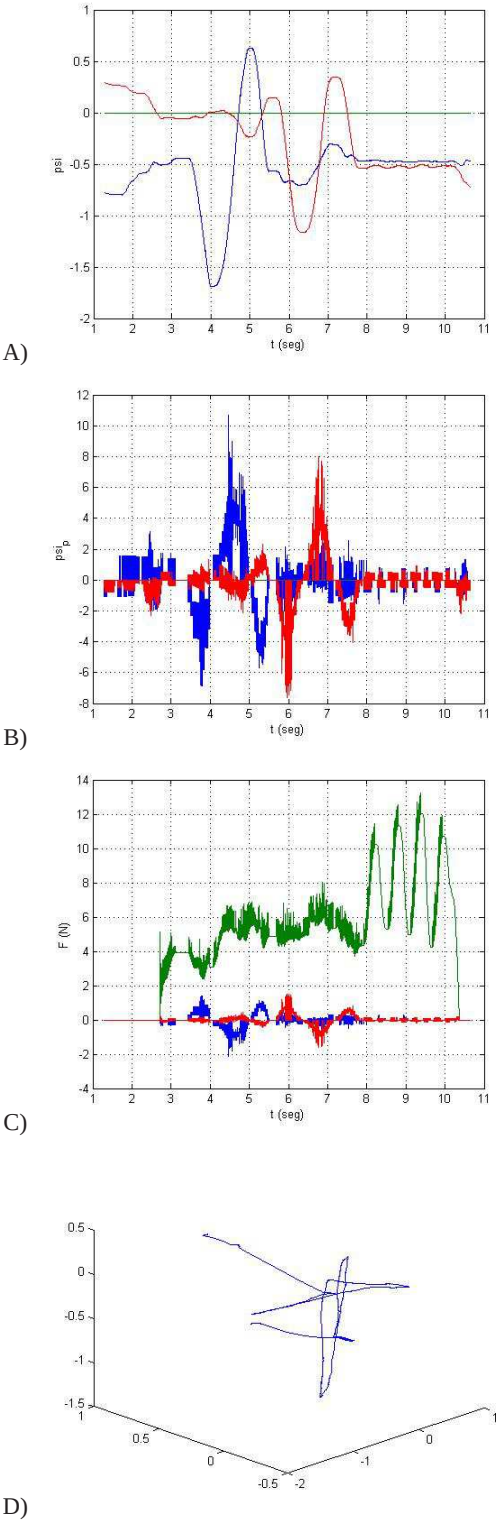


Fig. 11: Interaction with a plane with visco-elastic properties and surface linear friction. A) ξ_t, B) $\dot{\xi}_t$, C) Haptic device force, D) 3D movement

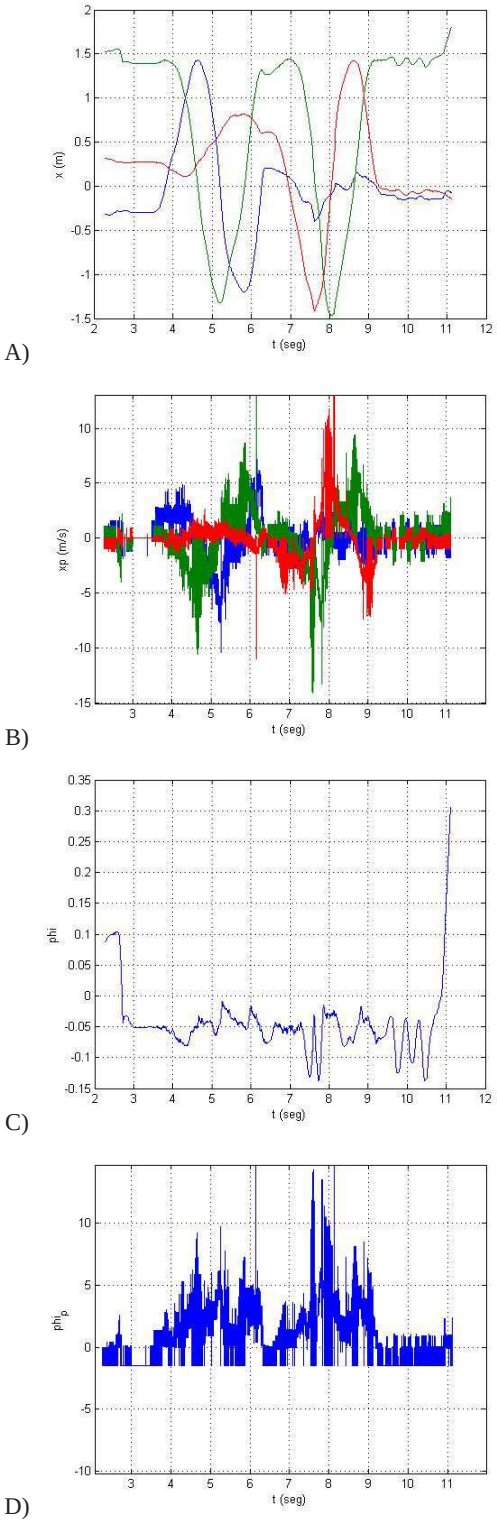


Fig. 12: Interaction with a sphere with visco-elastic properties and surface linear friction. A) Proxy position, B) Proxy velocity, C) φ_n, D) $\dot{\varphi}_n$

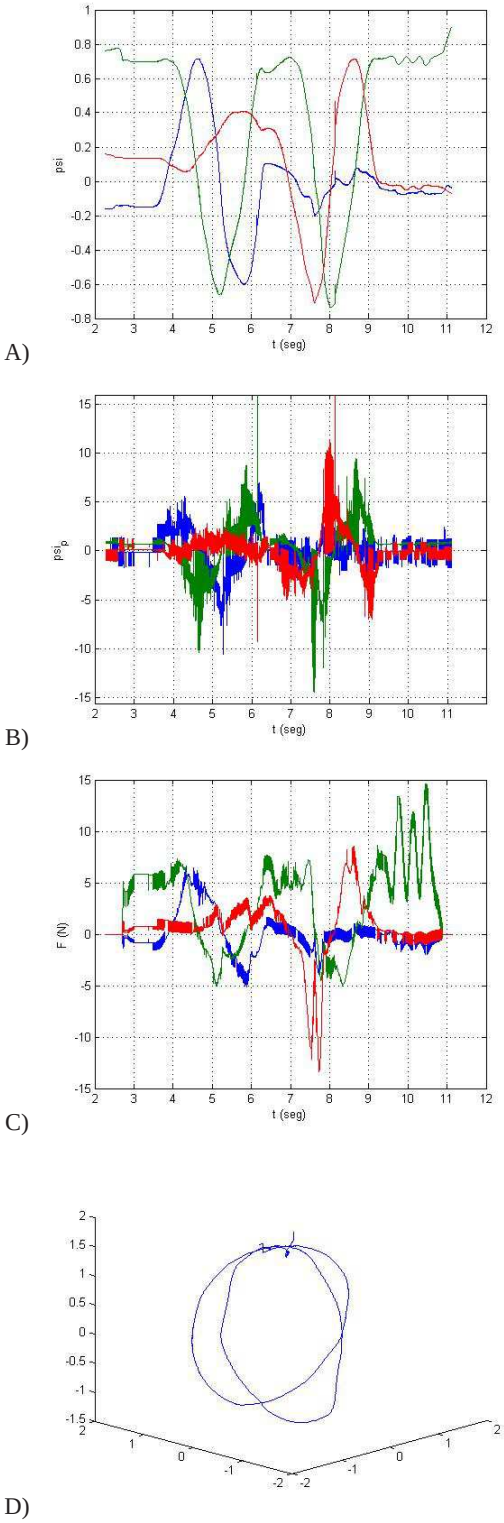
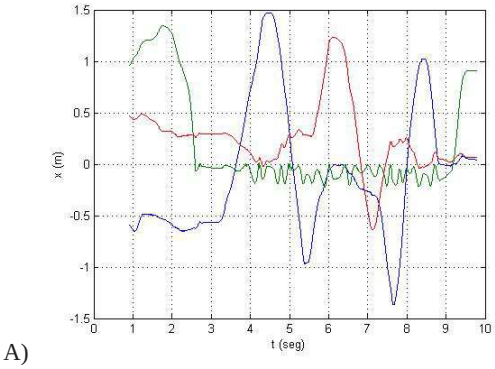
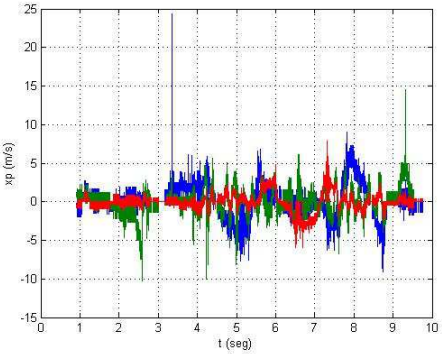


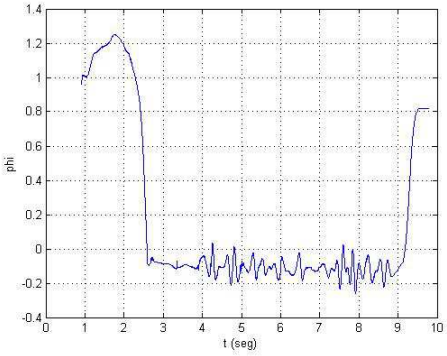
Fig. 13: Interaction with a sphere with visco-elastic properties and surface linear friction. A) ξ_t, B) $\dot{\xi}_t$, C) Haptic device force, D) 3D movement



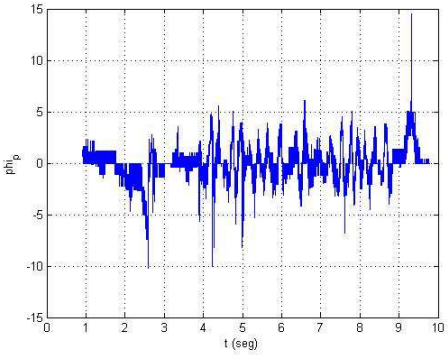
A)



B)

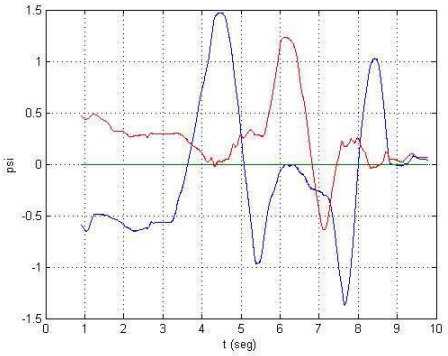


C)

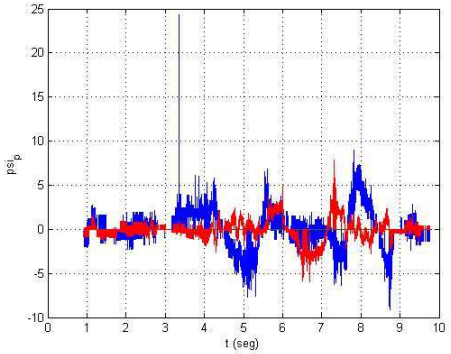


D)

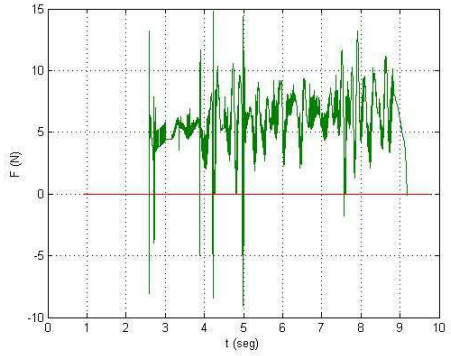
Fig. 14: Interaction with a plane with visco-elastic properties and textured surface. A) Proxy position, B) Proxy velocity, C) φ_n, D) $\dot{\varphi}_n$



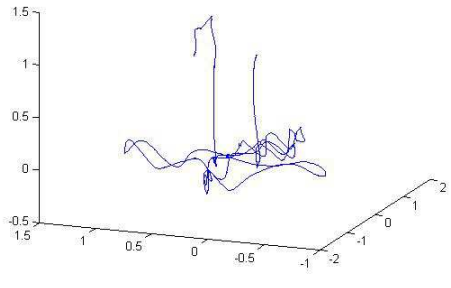
A)



B)



C)



D)

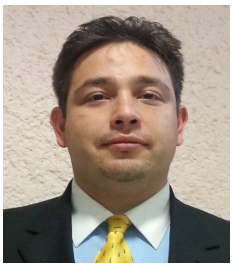
Fig. 15: Interaction with a plane with visco-elastic properties and textured surface. A) ξ_t, B) $\dot{\xi}_t$, C) Haptic device force, D) 3D movement

Acknowledgement

The authors would like to thank Instituto Politécnico Nacional CIDETEC and CONACyT for the support.

References

[1] Diana Garroway, ”6DOF Haptic Redering Using Geometric Algebra”, Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE, pp. 103-107
[2] Kenneth Salisbury, Francois Conti, Federico Barbagli, ”Haptic Redering Introductory Concepts”, IEEE Computer Graphics and Applicatios, pp. 24-32, January, 2004,
[3] Gabriel Sepulveda-Cervantes, Vicente Parra-Vega, Omar Dominguez-Ramirez, ”Dynamic Coupling Haptic Suturing Based on Orthogonal Decomposition”, Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems Salt Lake City, UT, USA, March 18-20, 2009
[4] D. Hestenes, and G. sobezyk, ”Clifford Algebra to Geomtric Calculus: A Unified Language for Mathematics and Physics”, in D. Reidel (ed): Dordrecht, 1984
[5] Alan Macdonald, ”A Survey of Geometric Algebra and Geometric Calculus”, Luther College, Decorah, IA 52101 USA, 2012
[6] Joan Lasenby, Anthony Lasenby and Chris J.L. Doran, ”A Unified Mathematical Language for Physics and Engineering in the 21st Century”, Royal Society Typescript, 1996
[7] Jing Fei Jiang and Jian Qiu Zhang, ”Geometric Algebra of Euclidean 3-Space for Electromagnetic Vector-Sensor Array Processing Part I: Modeling”, IEEE Transaction on Antennas and Propagation, Vol 58, No 12, December, 2010
[8] Leandro A. F. Fernandes and Manuel M. Oliveira, ”Geometric Algebra: A powerful tool for solving geometric problems in visual computing”, XXII Brazilian Symposium on Computer Graphics and Image Precessing, IEEE, 2009
[9] A. Gentile, S. Segreto, F. Sorbello, G. Vassallo, S. Vitabile, V. Vullo, ”CliffoSor: a Parallel Embedded Architecture for Geometric Algebra and Computer Graphics”, Seventh Workshop on Computer Architecture for Machine Perception, 2005
[10] Minh Tuan Pham, Tomohiro Yoshikawa, Takeshi Furuhashi and Kanta Tachibana, ”Robust Feature Extractions from Geometric Data using Geometric Algebra”, IEEE Internationa, Conference on Systems, Man and Cybernetics, USA, 2009.
[11] Luis Enrique Gonzlez-Jimnez, Oscar Eleno Carbajal-Espinosa, and Eduardo Bayro-Corrochano, ”Geometric Techniques for the Kinematic Modeling and Control of Robotic Manipulators”, IEEE International Conference on Robotics and Automation, 2011
[12] Leo Dorst, Daniel Fontijne, Stephen Mann, ”Geometric Algebra for Computer Science (Revised Edition): An Object-Oriented Approach to Geometry”, Morgan Kaufmann, 2007
[13] Chaim Zonnenberg, ”Conformal Geometric Algebra Package”, Master Thesis, Utrecht University, 2007
[14] Daniel Fontijne, ”Gaigen 2: a Geometric Algebra Implementation Generator”, GPCE’06, USA 2006
[15] Daniel Fontijne, Leo Dorst, Tim Bouma, Stephen Mann, ”GAViewer 0.85”University of Amsterdam, University of Waterloo
[16] Cagatay Basdogan, Mandayam A. Srinivasan, ”Haptic Rendering in Virtual Environments”, IEEE International Conference on Robotics and Automation, and IEEE Virtual Reality Conference, 1996
[17] Kenneth Salisbury Christopher Tarr, ”Haptic Rendering of Surfaces Defined by Implicit Functions”, Proceedings of the ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, USA, 1997
[18] Gabriel Seplveda-Cervantes, Vicente Parra-Vega, Omar A. Dominguez-Ramirez, ”Dynamic Coupling Haptic Suturing Based on Orthogonal Decomposition”, Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems Salt Lake City, UT, USA, March 18-20, 2009
[19] Adapted from R.L. Klatzky, et al. , Procedures for haptic object exploration vs. manipulation, Vision and action: The control of grasping, ed. M.Goodale, New Jersey: Ablex, 1990, pp. 110-127
[20] Gino van den Bergen, ”Collision Detection in Interactive 3D Enviroments”,Morgan Kaufmann Publishers, Ed. 1, 2004
[21] M. Moore and J. Wilhelms, ”Collision Detection and Response for Computer Animation”, in Proc. of Intl. Conf. on Computer Graphics and Interactive Techniques, 1988, pp. 289 to 298
[22] A. Petersik and B Pflesser and U. Tiede and K.H. Hoehne. and R. Leuwer, ”Haptic Volume Interaction with Anatomic Models at Sub-Voxel Resolution”, Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002.



G. S. Cervantes Received the B.S. in Communications and Electronic Engineering from Instituto Politécnico Nacional, and the PhD degree at CINVESTAV IPN Mexico in 2009. He is a researcher in haptics, virtual reality and mechatronics in CIDETEC. He is member of the National Researcher System in Mexico.

E. A. P. Flores Received the B.S. in Communications and Electronic Engineering from Universidad Autonoma Metropolitana, the M.S. at Instituto Tecnológico de Puebla, Mexico, and PhD degree at CINVESTAV IPN Mexico in 2006. He is a researcher in Optimization and Concurrent Design in CIDETEC. He is member of the National Researcher System in Mexico.