The Spherical Radial Function Network for Reconstruction in Medical Robotics

E. Castillo-Muñiz and E. Bayro-Corrochano

Abstract—The main goal of this work is to develop a geometric neural network which can be used as an interface between sensors and robot mechanisms. For this goal, we have developed a new geometric network called Spherical Radial Basis Function Network using the conformal geometric algebra framework. The motivation to use circles or spheres as activation functions is due to the fact that the sphere is the computational unity of the conformal geometric algebra, as a result a Spherical Radial Basis Network can be advantageously used as interface between the sensor domain and the robotic mechanism so that all the computing can be done in the same mathematical framework. In fact, there will be no need to abandon the system for the interpolation or reconstruction using a network. This article presents the design principles and a comparison with a standard Radial Basis Function Network. Medical robotics is an interesting domain to apply our network for capturing data and reconstruct automatically the shape of a human organ.

I. INTRODUCTION

The modeling of objects using image processing techniques is a research area which has grown rapidly. Several high quality algorithms have been developed for that purpose [1]. In fact, there has been a big advance in 3D object reconstruction (surface approximation) since marching cubes algorithm, as in the work [2]. The object reconstruction problem is the focus of many researchers in various fields such as computer vision, medical imaging, CAD, etc. Specifically in the field of medicine, researchers are currently working on the reconstruction of different organs such as bones or soft tissue [3]. In general, the medical community requires the reconstruction of organs for diagnosis, simulation, medical training, surgical planning, etc. The use of haptic interfaces in medicine is an area of growing interest, especially in robot-assisted endoscopic surgical and endoscopic robot-assisted surgical training [4]. Radial Basis Networks (RBFN) are a special kind of networks using radial basis function (RBF) as activation functions. Such networks are widely used in pattern recognition, control and for the approximation of functions. The authors of this paper have developed a new geometric network called Spherical Radial Basis Function Network (SRBFN) using the conformal geometric algebra framework. This article presents the design principles and a comparison with a standard RBFN. Our motivation to use circles or spheres is due to the fact that the sphere is the computational unity of the conformal geometric algebra, as a result a SRBFN can be advantageously used as interface between the sensor domain and the robotic mechanism so that all the computing can be done in the same mathematical framework. In fact, there will be no need to abandon the system for the interpolation or reconstruction using a network.

The article [5] showed the advantages of using spheres to represent any body using an algorithm called Inner Sphere Trees, specifically the paper proposed a unified language for haptic rendering and as a proposed solution to the problem of virtual collisions. Authors in [6] intends to make haptic rendering using circles in 2D or 3D spheres joining their centers by a skeleton. In this paper, we show that the SRBFN can be applied to reconstruct automatically the shape of a human organ using visual data. This paper presents the reconstruction results using our new geometric neural network.

The paper is organized as follows: the second section describes briefly medical imaging. Section third outlines the conformal geometric algebra. Section fourth presents the design principles of the SRBFN. Section fifth is devoted to the experimental analysis which is followed by a conclusion section.

II. MEDICAL IMAGING

As stated in [7], in manual minimally invasive surgery (MIS), surgeons feel the interaction of the instrument with the patient via a long shaft, which eliminates tactile cues and masks force cues. However, robot-assisted minimally invasive surgery (RMIS) holds great promise for improving the accuracy and dexterity of a surgeon while minimizing trauma to the patient. Nevertheless, the clinical success with RMIS has been marginal. It could be due to the lack of haptic (force and tactile) feedback presented to the surgeon. The goal of haptic interfaces in RMIS is to provide the surgeon the sensation of his own hands are contacting the patient. Force feedback systems for RMIS typically measure or estimate the forces applied to the patient by the surgical instrument, and provide resolved forces to the hand via a force feedback device. There are commercially available force sensors, and some researchers have created specialized grippers that can attach to the jaws of existing instruments. In this work, we use a haptic interface to approximate the mechanical properties of an organ which is touched, as well as to obtain information about the surface of such organ. For this purpose, we use a model of the object of interest, feel well, sense it by touching it in different points using the haptic device. As a result, we obtain a set of points with certain mechanical properties. Then, our new network SRBFN,
is used to interpolate/extrapolate in order to approximate the mechanical properties of the entire surface.

Before working with the real sensors, we needed to test our proposal on a real object. For that we used a 3D model as ground truth, as well as a haptic interface to simulate the acquisition of points and their mechanical properties. The National Library of Medicine has a project called the Visible Human. It has already produced computed tomography, magnetic resonance imaging and physical cross-sections of a human male cadaver. They use surface connectivity and isosurface extraction techniques to create polygonal models of the skin, bone, muscle and bowels. The goal of such project is the creation of complete, anatomically detailed, three-dimensional representations of the normal male and female human bodies. Acquisition of transverse CT, MR and cryosection images of representative male and female cadavers has been completed. The male was sectioned at one millimeter intervals, the female at one-third of a millimeter intervals. Figure 1 shows an example of the cloud of points and the surface corresponding to a human liver. This image was taken of the digital National Library of Medicine [12].

In order to determine the accuracy of the surface approximation and the mechanical properties estimated by the network, we used as ground truth the 3D model depicted in Figure 1. In addition to the cloud of points and the mesh of such model, we included certain mechanical properties for the entire surface of the model used in the experiments. Basically, we define stiffness, damping and mass properties for points on the surface.

III. GEOMETRIC ALGEBRA

Geometric Algebra is defined over a vector space. Let \( \mathbb{R}^{p,q} \) denote a \((p+q)\)-dimensional vector space over the reals \( \mathbb{R} \). The canonical basis of \( \mathbb{R}^{p,q} \) is defined as the ordered set \( e_1, e_p, e_{p+1}, \ldots, e_{p+q} \), where the \( e_i \) have the property

\[
e_i \star e_j = \begin{cases} +1 & 1 \leq i = j \leq p \\ -1 & p < i = j \leq p + q \\ 0 & i \neq j \end{cases}
\]  

(1)

where \( \star \) denotes a commutative scalar product. The geometric algebra over \( \mathbb{R}^{p,q} \) is denoted by \( G_{p,q} \), and the algebra product is called the Clifford or geometric product, which will be denoted by juxtaposition of symbols, and it is defined as

\[
ab = a \cdot b + a \wedge b.
\]  

(2)

where \( a \cdot b \) indicates the standard inner product (which is equal to the scalar product for vectors), and \( \text{wedge} a \wedge b \) indicates the antisymmetric outer product, which is interpreted as a new entity: the parallelogram spanned by the vectors \( a \) and \( b \).

In the following let \( A[i] \) denote the \( i \)-th element of an ordered set \( A \). A basis blade in \( G_{p,q} \) is the geometric product of \( p \) and \( q \) vectors. The grade of a basis blade is defined as \( gr(e_A) = |A| \).

The canonical algebraic basis of \( G_{p,q} \) is given by the ordered set \( e_A : A \in P^k_{O}(I) \), where \( I = 1, \ldots, p + q \).

Let \( E_i \) denote the \( i \)-th element of \( G_{p,q} \); then a general multivector of \( G_{p,q} \) may be written as

\[
A = a^i E_i, \quad i \in 1, 2, \ldots, 2^n
\]  

(3)

where \( a^i \in \mathbb{R} \) and a summation over the range of \( i \) is implied. Given a vector space \( \mathbb{R}^{p,q} \) with canonical basis, there are \( 2^n \), \( n = p + q \) ways to combine the \( e_i \) with the geometric product such that they are linearly independent. The collection of such \( 2^n \) basis blades forms an algebraic basis of \( G_{p,q} \). The element of highest grade of the canonical basis of \( G_{p,q} \) is said to be the pseudoscalar of that algebra.

Some of the basic operations in any geometric algebra are reversion, conjugation and duality. The reverse of \( e_A \), denoted as \( e_A^r \) is defined as

\[
e_A^r = \prod_{j=1}^{k} e_A[k-j+1]
\]  

(4)

For example, if \( e_A = e_1 e_2 e_3 \), then \( e_A^r = e_3 e_2 e_1 \). The conjugate of \( e_A \), denoted by \( e_A^* \), is defined as

\[
e_A^* = (-1)^r e_A
\]  

(5)

where \( r \) is the number of basis vectors in a basis blade that square to \(-1\). The dual of a basis blade \( E_i \) is denoted by \( E_i^* \) and defined as

\[
E_i^* = E_i I^{-1}
\]  

(6)

where \( I \) denotes the unit pseudoscalar of \( G_{p,q} \), and \( I^{-1} \) denotes its inverse.

A blade is defined as the outer product of a number of 1-vectors in \( G_{p,q} \). The outer product of \( k \) 1-vectors is called a \( k \)-blade or blade of grade \( k \), and is denoted by

\[
A_{(k)} = a_1 \wedge a_2 \wedge \ldots \wedge a_k = \bigwedge_{i=1}^{k} a_i
\]  

(7)

Note that a \( k \)-blade is a linear combination of basis blades of grade \( k \). However, not every linear combination of basis blades of grade \( k \) is a \( k \)-blade.

The projection of a blade \( A_{(k)} \in G_{p,q}^k \) (given that it is not null blade), onto a blade \( N(l) \in G_{p,q}^l \), with \( 1 \leq k \leq l \leq p+q \), is defined as

\[
P_{N(l)}(A_{(k)}) = (A_{(k)} \cdot N^{-1}(l))N(l)
\]  

(8)
If null blades are considered, then the projection of $A_{(k)}$ onto $N_{(l)}$ is defined as

$$P_{N_{(l)}}(A_{(k)}) = (A_{(k)} \cdot N_{(l)}^+)N_{(l)}$$  \hspace{1cm} (9)

where $N_{(l)}^+$ represents the pseudoinverse of a blade, given by

$$N_{(l)}^+ = \frac{N_{(l)}^\dagger}{N_{(l)} \cdot N_{(l)}^\dagger}$$  \hspace{1cm} (10)

A versor (key concept in the conformal versor neuron), is a multivector that can be expressed as the geometric product of a number of non-null 1-vectors. For each versor $V$, there exists an inverse versor, denoted by $V^{-1}$. A versor $V$ such that $V^{-1} = V$ is called unitary; i.e., $VV = +1$. The set of unitary versors forms the so called Pin Group. In addition, if a unitary versor $V$ can be expressed as the geometric product of an even number of 1-vectors, then it is called a spinor. The set of spinors forms the so called spin group.

A. Conformal Geometric Algebra

To work in conformal space means to embed the Euclidean space in a higher dimensional space, where the extra dimensions have particular properties and can represent geometric entities in Euclidean space (geometric objects or transformations). A conformal transformation preserves angles locally. All conformal transformations can be expressed by means of combinations of inversions, and all Euclidean transformations can be expressed by conformal transformations formed as combinations of two reflections, because they are a subset of conformal transformations.

For the 3D space, the corresponding conformal geometric algebra is denoted by $G_{3+1,1} = G_{4,1}$. Given a 3D Euclidean vector $x \in G_3$, its embedding in the conformal space is given by

$$X = x + \frac{1}{2}x^2e_\infty + e_0$$ \hspace{1cm} (11)

where $e_\infty = e_+ + e_+$ and $e_0 = \frac{1}{2}(e_- - e_+)$, and correspond to the point at infinity, and the origin, respectively. In addition

$$e_\infty^2 = e_0^2 = 0, \quad e_\infty \cdot e_0 = -1, \quad E = e_\infty \wedge e_0$$ \hspace{1cm} (12)

In CGA $G_{4,1}$, a vector with the form $S = A - \frac{1}{2}\rho^2e_\infty$ represents a sphere in the 3D Euclidean space (by its Inner Product Null Space), and its dual $S^*$ represents the same sphere, but it can be constructed by the outer product of four conformal vectors on the sphere; i.e. $S^* = A \wedge B \wedge C \wedge D$ (the representation in the so called Outer Product Null Space).

A vector of the form

$$Pl = A - e_0 - \frac{1}{2}\rho^2e_{\infty} = \hat{a} + \alpha e_\infty$$ \hspace{1cm} (13)

represents a plane with an orthogonal distance $(a^2 - \rho^2)/(2\|a\|)$ from the origin and a normal $a$. Its dual (the outer product representation) is easy to compute with three conformal vectors and the point at infinity: $Pl = A \wedge B \wedge C \wedge e_\infty$.

Given two spheres, its intersection (if any), is a circle, and it is given by

$$C = S_1 \wedge S_2$$ \hspace{1cm} (14)

It can be also computed by the outer product of three vectors $C = A \wedge B \wedge C$.

Let $Pl_1$ and $Pl_2$ be two (inner product) planes; then the intersection line is given by

$$C = Pl_1 \wedge Pl_2$$ \hspace{1cm} (15)

Its dual (outer product) representation can be obtained with the outer product of two vectors and the point at infinity; that is, the line can be conceived as a circle of infinite radius; i.e., $L = A \wedge B \wedge e_\infty$.

In Conformal Space, reflections are represented by planes, and it is implemented by $PlXPi$. An inversion of a vector in $G_{4,1}$ in the unit sphere centered at the origin is given by a reflection in $e_+;$ that is $e_+xe_+$. A translation by a 3D vector $t$, expressed in CGA is implemented by the so called translator, which is given by the product of two planes $Pl_1$ and $Pl_2$

$$T = Pl_2Pl_1 = 1 - \frac{1}{2}te_\infty$$ \hspace{1cm} (16)

which can be given in exponential form as $T = exp(-\frac{1}{2}te_\infty)$.

Rotations in 3D Euclidean space are carried out in the conformal space by the so called rotor. The rotor which can rotate not only on the origin, but also on a different position is given by

$$G = TR\tilde{T}$$ \hspace{1cm} (17)

where $R$ is a rotor on the origin given by

$$R = \hat{m}\hat{n} = \cos(\theta/2) - \sin(\theta/2)\hat{n} \wedge \hat{m} = \exp(-\frac{\theta}{2}\hat{n}e_\infty).$$ \hspace{1cm} (18)

A dilation is an isotropic scaling and can be achieved by two consecutive inversions over spheres of different radii. The resulting multivector is called dilator; this operator representing a dilation by a factor $d$ about the origin is defined as

$$D = 1 + \frac{1 - d}{1 + d}e_\infty \wedge e_0$$ \hspace{1cm} (19)

Such operator dilates about the origin. The general dilator about a point $t$ is given by $D = TDT^\dagger$.

3D rigid motion can be represented in terms of the called motor as follows:

$$\theta' = M\theta\hat{M} = (TR)\theta(\tilde{R}\tilde{T}).$$ \hspace{1cm} (20)

where $M = R(1 + e_\infty\hat{a})$ and $\theta$ stands for any geometric entity like points, lines, circles or spheres.

The reader is encouraged to see the CGA representation of other entities consulting [8].

IV. SPHERICAL RBFN

A. Radial Basis Function Network

The Radial Basis Functions Networks (RBFN) are widely used for function interpolation, regression and as well as classifiers. The neuron model of a RBFN is shown in figure 2 (a). It consists of an input layer connected to a hidden layer which in turn is connected to an output layer. At the hidden
layer the activation function is a radial basis function, i.e. a
real valued function which actual value depends only on the
distance to a centre \( c \) as follows
\[
f(x, c) = \| x - c \|
\]
where \( \| . \| \) is the Euclidian norm.

There are several of these functions in the literature, more
references can be found in [10]. The most classically RBF
uses the Gaussian function and it is known as an universal

\[
\text{approximator } [11].
\]
uses the Gaussian function and it is known as an universal
references can be found in [10]. The most classically RBF
uses the Gaussian function and it is known as an universal

\[
\text{approximator } [11].
\]
uses the Gaussian function and it is known as an universal
references can be found in [10]. The most classically RBF
uses the Gaussian function and it is known as an universal

Fig. 2. In a) the neuron model of a RBFN, in b) the architecture of a RBFN implemented in MATLAB,

The problem of approximation using Gaussian functions
is first to select a small number of these functions and then
to find their centres and standard deviations. The amount if
functions can be increased. Then the weights of the output
layer are adjusted so that their sum approximate well the
original function in every point of it. The \( i-th \) neuron of
the hidden layer has the activation function \( f_i(x) \) given by
\[
f_i(x) = a_i e^{-b_i (x-c_i)^2}.
\]
(21)
The goal of the training algorithm is to find the value of the
parameters \( a_i, b_i \) and \( c_i \) such that the value of
\[
| F - \sum_{i=1}^{n} (a_i e^{-b_i (x-c_i)^2}) |
\]
Will be minimal. \( F \) is the function to approximate and \( n \) the
number of neurons in the hidden layer.

The training of the RBFN’s can be called hybrid, since
the structure of the same search for the centers and standard
deviations of the functions is a nonlinear problem, while the
determination of the weights of the output layer is a linear
problem. Normally two different algorithms are used for each of
these problems.

B. Spherical Radial Basis Function Network

In this work, we propose to use hyperspheres (circles,
spheres) as activation functions of the neurons in hidden
layer. Here the network will strictly use the half of the circles
or spheres in order to avoid violation of the definition of the
involved function. Our motivation to use circles or spheres
is due to the fact that the sphere is the computational unity
of the conformal geometric algebra, as a result a Spherical
Radial Basis Network can be greatly used as interface
between the sensor domain and the robotic mechanism so
that all the computing can be done in the same mathematical
framework. There will be no need to abandon the system for
the interpolation or reconstruction using a network.

In the literature, is no work reported that it used this kind
of activation functions. In this work, the one-dimensional
radial function considered is as follows
\[
f(x) = a_i + \sqrt{b_i^2 - (x - c_i)^2}
\]
(22)
where
\[
c_i - b_i < x < c_i + b_i
\]
Since this kind of activation functions can be expanded by
polynomials, so, conditions of the Stone-Weierstrass theorem
are satisfied.

In the Geometric Algebra framework, the Outer Product
Null Space (OPNS) representation of a circle is given by
\( p1 \wedge p2 \wedge p3 \) where \( p1, p2 \) and \( p3 \) are three points lying on
the circumference and \( \wedge \) is the wedge product of geometric
algebra.

As we can see, the equation (22) has two parameters, but
we add the parameter \( a \), which may push each semicircle
along the vertical axis, thus as far the amount of unknowns
the problem of adjusting semicircles is similar to the one
by Gaussian functions. In the figure 3 (a), we see one
interpolation consisting of three Gaussian functions, and in
3 (b), a function consisting of three semicircles. Note that,
there are three circles depicted, but only the semicircles
contribute to the sum.

![Fig. 3. a) The weighted sum of three Gaussian functions. b) The weighted sum of three semicircles.](image)

This network was implemented in MATLAB, and was
scheduled so that training in the nonlinear stage (parameters
\( b \) and \( c \)) is done using the algorithm Levenberg-Marquardt,
while the adjustment of the linear part (parameter \( a \)) is
linear least squares. During the training of the network given
a set of \( N \) points, the semicircles are added one by one
until the mean squared error function reach an expected
lower value. The maximum number of neurons is \( N \) and
the allowed maximum number of epochs is $M$. There is another parameter, the bias $b$, which defines the sensitivity of each neuron to be adjusted; this parameter should be set for each problem in order to obtain the best results. This implementation is exactly analogous to the function of MATLAB newrb, however we modified that newrb to adjust Gaussian functions.

![Fig. 4. Points to be adjusted.](image)

To measure the performance of both networks, we used a sample of 21 points, which are shown in the figure 4. The interpolation of those points was done using 4, 7, 10, ... 19 Gaussians and the same amount of semicircles. The MSE was measured for each approach. Figure 5 shows the results obtained. It should be mentioned that the bias parameter was taken equal to 1 for the Gaussian and equal to 2 for the semicircles.

![Fig. 5. Performance of the adjustment of points with Gaussian and semicircles functions.](image)

The table below shows the Mean Squared Error (MSE) for each amount of points used.

<table>
<thead>
<tr>
<th>AMOUNT</th>
<th>Semicircles</th>
<th>Gaussians</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>32.7628</td>
<td>25.1344</td>
</tr>
<tr>
<td>7</td>
<td>14.2517</td>
<td>3.2563</td>
</tr>
<tr>
<td>10</td>
<td>1.4038</td>
<td>3.2563</td>
</tr>
<tr>
<td>13</td>
<td>0.1042</td>
<td>3.2563</td>
</tr>
<tr>
<td>16</td>
<td>0.0198</td>
<td>3.2563</td>
</tr>
<tr>
<td>19</td>
<td>0.0196</td>
<td>3.2563</td>
</tr>
</tbody>
</table>

![Fig. 6. At left the adjustment of points with 4 and 19 Gaussian functions, at right the same but using semicircles.](image)

A. A Visual Application

The model of the liver is 3D. In order to visualize better the results, we start cutting the liver into two surfaces, the top and bottom, this is achieved "cutting" the liver with planes. To get the equation of the plane we take 3 points (these points were selected visually): $p1(43, 51, 88), p2(220, 56, 117)$ and $p3(103, 200, 158)$. The equation of this plane in Conformal Geometry in the OPNS representation is $p1 \wedge p2 \wedge p3 \wedge e_\infty$. Then we take all points up the plane to form the top surface and the others to form the bottom surface. These surfaces are showed in figure 7.

![Fig. 7. a) Point cloud of the top surface. b) Point cloud of the bottom surface.](image)

Now we reconstruct a visual cut on the top surface of the liver, will call sagittal cut to one perpendicular to the axis X and a coronal cut to one perpendicular to the Y axis as shown in the figure 8 (a). The goal of this is to see how many points would be sufficient for our network to fit the surface on along these cuts. The experiment was performed by taking random points on each slice and calculated the normalized mean square error between the reconstruction and the cloud of points. In each cut, we took different amounts of random points. The results are shown in the figure 8 (b).

V. EXPERIMENTAL ANALYSIS

To show an application we will use the 3D model of liver above mentioned. Our first goal will be to see how well the network is capable to fit the liver body surface. After that we conducted simulated experiment which aims to assess the possibility of reconstructing the elasticity properties of the liver surface using a haptic interface together with our network.
spheres. As opposite to RBF, our network codes nonlinear surfaces using the geometric entities like spheres, which are essential to relate perceptual data with geometric constrains of robot mechanisms; i.e, the robot manipulation guided by stereo vision can follow a tissue path supported by these spheres which in turn are represented in our network. Furthermore if we want to relate haptics cues with the surface, the spheres will be of great use. The changing elasticity parameter of the surface can be represented as normals of the spheres. The weights of these normals vary smoothly as the network can also interpolate these values. This extension of the representation capability of our network is matter of future work though. The relation between the robot end effector and the surface is established by a rigid motion represented in terms of algebra of motors (rigid motion representation in geometric algebra) using as a reference of the robot mechanism a plane, circle or sphere and any sphere of the network representing the surface. The end effector can then follow a path on the surface visiting individual spheres through the surface, as it is shown in figure 9. Note that:

$$ S'_r = M_i S_{surf} \hat{M}_i $$

$$ Z' = \Pi \wedge S'_r $$

The results obtained in the previous section suggest us to consider whether it would be possible apart of the surface interpolation to adjust data representing the elastic properties the object surface as well. In the case of an organ, if this possible, we can think that it could be represented virtually combining its shape and its surface elastic properties in real time using stereo cameras and force sensor attached to a haptic interface.

**VI. CONCLUSION**

This articles presents the design and implementation of the Spherical Radial Function Network using the geometric algebra framework. The motivation to use circles or spheres is due to the fact the sphere is the computational entity of the conformal geometric algebra, as a result a Spherical Radial Basis Network can be of great use as interface between the sensor domain and the robotic mechanism so that all the computing can be done in the same mathematical framework. In fact, there is no need to abandon the system for the interpolation or reconstruction using a network. In the experimental part, we carried out a comparison of its performance against of that of a standard Radial Basis Function Network. We believe that the Spherical Radial Function Network is a promising computing approach for diverse applications like in haptics based medical robotics.

**REFERENCES**


