

Conformal Hough Transform for 2D and 3D Cloud Points

Gehová López-González¹, Nancy Arana-Daniel²,
and Eduardo Bayro-Corrochano¹

¹ Department of Electrical Engineering and Computer Science, CINVESTAV,
Guadalajara, Jalisco, México

² CUCEI, University of Guadalajara, Guadalajara, Jalisco, México

Abstract. This work presents a new method to apply the Hough Transform to 2D and 3D cloud points using the conformal geometric algebra framework. The objective is to detect geometric entities, with the use of simple parametric equations and the properties of the geometric algebra. We show with real images and RGB-D data that this new method is very useful to detect lines and circles in 2D and planes and spheres in 3D.

1 Introduction

The Hough transform is an algorithm for feature extraction used in image analysis, computer vision, and digital image processing [4]. This technique collects imperfect instances of objects within a certain class of shapes by a voting procedure in accumulators or cells. This voting scheme is carried out in a parameter space, where candidate objects are obtained as local maximas in the accumulator space. The selection of the maxima of possible clusters are identified by a type of K-means algorithm. The Hough transform was developed for the identification of lines in the image [1], but later works extended the Hough transform to identifying positions of different shapes, for example circles or ellipses[2][3].

In this work, using the conformal geometric framework, we extend the randomized Hough transform to detect lines and circles in 2D cloud points of images and lines, planes, circles and spheres in 3D cloud points, obtained by 3D scanners and RGB-D sensors. We show the efficiency of our algorithm using real images and data of a RGB-D sensor.

This paper is organized as follows: Section II explains the Randomized Hough Transform. Section III presents a short introduction to geometric algebra and the operations that will be used in this work. Section IV outlines the conformal geometric algebra and the concepts used in the algorithm. Section V includes a description of the algorithm, and section VI discusses some algorithmic details. Section VII presents some experimental results. Finally section VIII is devoted to the conclusions.

2 Randomized Hough Transform

This work is an extension of the Randomized Hough Transform, RHT, [9]. The RHT is an algorithm proposed to solve the problems of the Hough Transform,

HT. In the HT algorithm for each image pixel not only the cells of the possible entities are increased, but also of many other. This creates a problem to find the local maxima. Also the accumulator array is predefined by windowing and sampling the parameter space. For the correct detection of the entities we need a good parameter resolution. For this we need a big array that takes too much storage and computing time. Without some previous knowledge of the image is very hard to determine the size of the accumulator array. A bad accumulator array can lead to the next problems: a failure to detect some specific entities, difficulties in finding local maxima, low accuracy, large storage and low speed.

The RHT solves these problems using parametric equations to only compute the possible entities and a dynamic accumulator array to solve the problems with the storage. By doing this the storage space is greatly reduced. Other concept added is a scalar δ used as tolerance for similitude. When the difference between two computed entities is smaller than δ , then we consider that the two are the same. This scalar is used to set the resolution of the accumulator array. If we increase δ the resolution and storage space will be lower.

The steps of the algorithm are:

- 1) Randomly take n points from the set, being n the number of points needed to define the entity.
- 2) Solve the parametric equations to get a candidate.
- 3) Search for the candidate in the dynamic accumulator array. If the candidate is found increase the accumulator by one. Otherwise add a new cell for the candidate and set its accumulator to one.
- 4) If an accumulator surpasses a threshold, we check if the entity exists in the image. If it exists, we add it to a list of detected entities and delete from the set all the points that belong to it.

Finally, we must note that the RHT is a stochastic method which its performance depends on the selection of δ and the randomized selection of the points.

3 Geometric Algebra

Geometric algebra is a coordinate-free approach to geometry based on the algebras of Grassmann and Clifford [5][6][7].

Let \mathbb{V} be a vector space of dimension n . We will define an algebra \mathbb{G}_n , called a geometric algebra. Let e_1, e_2, \dots, e_n be a set of basis vectors of \mathbb{V}^n .

The product, called the geometric product, of 2 basis vectors is anticommutative, $e_j e_k = -e_k e_j, \forall j \neq k$. These basis vectors must square to 1 or -1; this means that there are positive integers, p and q , such that $n = p + q$ and $e_i^2 = 1, i = 1 \dots p, e_i^2 = -1, i = p + 1, \dots, n$.

The product of elements of the basis of \mathbb{G}_n will simply be denoted by juxtaposition. In this way, from any two basis vectors, e_j and e_k , a new element of the algebra is obtained, and denoted as $e_j e_k = e_{jk}$. This introduces the concept of grade. The product of n non-equal basis vectors will result in a new generator of grade n . An algebra with n basis vectors will have 2^n generators $\{1, e_1, \dots, e_n, e_{12}, \dots, e_{(n-1)n}, \dots, e_{1\dots n}\}$. The generator of grade n is called the unit

pseudoscalar of the algebra, and is generally denoted by the letter I . The \mathbb{G}_n is a direct sum of the linear subspaces of grades $0,1,2,\dots,n$.

$$\mathbb{G}_n = \bigwedge^0 \mathbb{V}_n \oplus \bigwedge^1 \mathbb{V}_n \oplus \bigwedge^2 \mathbb{V}_n \oplus \dots \bigwedge^n \mathbb{V}_n \tag{1}$$

A multivector $A \in \mathbb{G}_n$ is separated in its blade elements of grade r as:

$$A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2 + \dots + \langle A \rangle_n = \sum_{r=0}^n \langle A \rangle_r \tag{2}$$

The most commons operations in geometric algebra are

- Geometric product

$$C = AB \tag{3}$$

It is associative, and distributive with respect to addition

For each $A \in \mathbb{G}_n$ it may exist an element A^{-1} which is called the multiplicative inverse of A and it is defined as follows:

$$AA^{-1} = 1 \tag{4}$$

$$A^{-1} = \frac{A}{A^2} \tag{5}$$

- Inner product, also known as dot product

$$\langle A \rangle_r \cdot \langle B \rangle_s = \langle AB \rangle_{|r-s|} \tag{6}$$

$$A \cdot B = \sum_r \sum_s \langle A \rangle_r \cdot \langle B \rangle_s \tag{7}$$

If r or s are 0, the result is 0. This is a grade decreasing operation.

- Outer product, also known as wedge product

$$\langle A \rangle_r \wedge \langle B \rangle_s = \langle AB \rangle_{r+s} \tag{8}$$

$$A \wedge B = \sum_r \sum_s \langle A \rangle_r \wedge \langle B \rangle_s \tag{9}$$

This is a grade increasing operation. If a and b are blades of grade 1 then $a \wedge b = -b \wedge a$. This means that the wedge product of two vectors is anti-commutative.

- Reverse and Norm

The reverse operation is denoted by \dagger . As its name suggests, it reverse the order of the elements of a k -blade, e.g. the reverse of e_{123} is e_{321} , also $e_{321} = -e_{123}$, so to compute the reverse of a k -blade, we must count the number of

signs caused by the interchange of elements. This can be computed with the next formula

$$\langle A \rangle_r^\dagger = (-1)^{(r(r-1)/2)} \langle A \rangle_r \quad (10)$$

The norm of a multivector is computed as follows

$$|A| = \sqrt{A^\dagger A} \quad (11)$$

This operation is defined to be applied in euclidean spaces, where all the basis square to 1. In this work, all the equations that involve the norm in conformal space are written as $|A| = \sqrt{AA}$.

4 Conformal Geometric Algebra

Conformal geometric algebra is a representation of the vector space as presented in [8]. The objective is to expand the vector space \mathbb{R}^n with the Minkowski space $\mathbb{R}^{1,1}$. The algebra of the Minkowski space has an orthonormal basis $\{e_+, e_-\}$ with the properties:

$$e_+^2 = 1, e_-^2 = -1 \quad (12)$$

$$e_+ \cdot e_- = 0 \quad (13)$$

A null basis, formed by $\{e_\infty, e_0\}$, is used to represent the point at infinity and the origin of a projective space and they can be defined as:

$$e_\infty = e_- - e_+ \quad e_0 = \frac{e_- + e_+}{2} \quad (14)$$

The vectors of the null basis have the properties:

$$e_\infty^2 = e_0^2 = 0 \quad e_\infty \cdot e_0 = -1 \quad (15)$$

The Minkowski Plane E is the unit pseudoscalar of this space and is defined as:

$$E = e_+ \wedge e_- = e_\infty \wedge e_0 \quad (16)$$

\mathbb{G}_3 is the algebra of the 3D space. The orthonormal basis of this algebra is defined as $\{e_1, e_2, e_3\}$. A point in this algebra is called an euclidean point, and is denoted by x_E .

$$x_E = X e_1 + Y e_2 + Z e_3 \quad (17)$$

The unit pseudoscalar of this algebra is denoted by I_E and is defined as

$$I_E = e_1 \wedge e_2 \wedge e_3 \quad (18)$$

Conformal geometric algebra can be defined as:

$$\mathbb{G}_{4,1} = \mathbb{G}_3 \oplus \mathbb{G}_{1,1} \quad (19)$$

The unit pseudoscalar of this algebra is I_c and is defined as

$$I_c = e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \wedge e_0 = I_E \wedge E \tag{20}$$

The sphere is the basic unit of calculus in $\mathbb{G}_{4,1}$. The sphere is defined as a set of points that are at the same distance (radius) from one point called center. If this definition is applied to \mathbb{R}^n the entity that fulfills it changes in each vector space, i.e. in \mathbb{R}^0 this entity is the point, in \mathbb{R}^1 it is the point pair, in \mathbb{R}^2 it is the circle, in \mathbb{R}^3 it is the sphere and in \mathbb{R}^4 and higher dimensions is called a hypersphere.

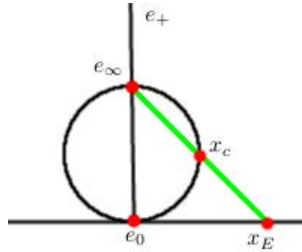


Fig. 1. Projection of the point $x_E \in \mathbb{G}_1$ into the conformal space $\mathbb{G}_{2,1}$ (which is a circle)

In CGA the Euclidean space is embedded in a higher dimension. Because $\mathbb{G}_{2,1}$ is possible to visualize, we use it to exemplify this process, see figure 1. First the basis e_+ expand the space by one dimension. In this dimension we can draw a unitary sphere centered in the origin. The basis e_- allows us to lift the bottom of the sphere to the origin of the space. Now we make a line from the top of the sphere to the euclidean point x_E . The intersection of this line with the sphere is the conformal point x_c . The conformal point x_c can be obtained with:

$$x_c = x_E + \frac{x_E \cdot x_E}{2} + e_0 = X e_1 + Y e_2 + Z e_3 + \frac{X^2 + Y^2 + Z^2}{2} e_\infty + e_0 \tag{21}$$

Now a sphere in \mathbb{R}^0 , a point, is a blade of grade 1. As defined before, the wedge product is a grade increasing operation. If we take 2 spheres of grade 1, points, the result is a sphere of grade 2, a point pair. Therefore the wedge product of 3 points is the circle and with 4 points we get a sphere.

There exist 2 special cases to consider: the first one occurs when the points are not in general position (*GP*), this means that there are 3 collinear points or 4 coplanar points. In this case the result will be a line with 3 points and a plane with 4 points. The line and plane are a circle and a sphere respectively with infinite radius. The other is known as the dual property of the conformal space: The result of the wedge product between an entity and a point will be 0 if the point lies on the entity.

The combination of the wedge product results and the dual property give us the Outer Product Null Space, $\mathbb{O}PN\mathbb{S}$, representation. A visualization of this is presented in the figure 2.

Entity	Notation	$\mathbb{O}PN\mathbb{S}$ representation	Grade
Point	x_c	$x_E + \frac{x_E}{2}e_\infty + e_0$	1
Point Pair	$\mathbb{O}N(P\mathcal{P})$	$x_{c1} \wedge x_{c2}$	2
Circle	$\mathbb{O}N(Z)$	$x_{c1} \wedge x_{c2} \wedge x_{c3}$; in GP	3
Line	$\mathbb{O}N(L)$	$x_{c1} \wedge x_{c2} \wedge x_{c3}$	3
Sphere	$\mathbb{O}N(S)$	$x_{c1} \wedge x_{c2} \wedge x_{c3} \wedge x_{c4}$; in GP	4
Plane	$\mathbb{O}N(P)$	$x_{c1} \wedge x_{c2} \wedge x_{c3} \wedge x_{c4}$	4

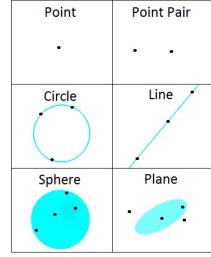


Fig. 2. $\mathbb{O}PN\mathbb{S}$ representation

In $\mathbb{G}_{4,1}$ there exist an alternate representation. This representation is called the Inner Product Null Space, $\mathbb{I}PN\mathbb{S}$. To change from $\mathbb{O}PN\mathbb{S}$ to $\mathbb{I}PN\mathbb{S}$ representations one multiply by the unit pseudoscalar I_c . This multiplication is called the dual operation. Because $I_c^2 = -1$ its inverse is $-I_c$, so to return to $\mathbb{O}PN\mathbb{S}$ we multiply by $-I_c$. This is done to avoid a change of sign between both representations.

$$\mathbb{O}N(X)(I_c) = \mathbb{I}N(X) \quad \mathbb{I}N(X)(-I_c) = \mathbb{O}N(X) \tag{22}$$

A special case is the point representation x_c . The $\mathbb{O}PN\mathbb{S}$ representation of the conformal point is also valid in $\mathbb{I}PN\mathbb{S}$.

The combinations of these two representations allow us to obtain the information of the blades that define the geometric entities, as it is shown next along with the equations to obtain the parameters.

– Sphere

$$\mathbb{I}N(S) = \pm\alpha(C_c - \frac{r^2}{2}e_\infty) \tag{23}$$

Where C_c is the conformal point of the center C_E , r is its radius and α is some scale. Because of the fact that the blade e_0 can only be equal to 1 in the sphere case, the scale can be obtained as:

$$\pm\alpha = -\mathbb{I}N(S) \cdot e_\infty \tag{24}$$

The equations to compute the parameters of the normalized sphere are

$$C_E = (\mathbb{I}N(S) \cdot (-I_E))I_E \tag{25}$$

$$r^2 = \mathbb{I}N(S)^2 \tag{26}$$

– Plane

$$\mathbb{I}\mathbb{N}(P) = \pm\alpha(n_E + de_\infty) \quad (27)$$

Where n_E is the normal of the plane, d is the distance to the origin and α is a scale factor equal to $|n_E|$. The equations to compute the parameters of the normalized plane are:

$$n_E = (\mathbb{I}\mathbb{N}(P) \cdot (-I_E))I_E \quad (28)$$

$$d = -\mathbb{I}\mathbb{N}(P) \cdot e_0 \quad (29)$$

– Circle

To obtain the plane in which the circle lies we use:

$$\mathbb{O}\mathbb{N}(P) = e_\infty \wedge \mathbb{O}\mathbb{N}(Z) \quad (30)$$

The norm of $\mathbb{O}\mathbb{N}(Z)$ is the same of $\mathbb{O}\mathbb{N}(P)$. The sphere with the same center and radius as the circle is obtained by:

$$\mathbb{I}\mathbb{N}(S) = \mathbb{O}\mathbb{N}(Z) \cdot \mathbb{O}\mathbb{N}(P)^{-1} \quad (31)$$

– Line

$$\alpha = \sqrt{\mathbb{O}\mathbb{N}(L) \cdot \mathbb{O}\mathbb{N}(L)} \quad (32)$$

where α is the scale of factor used to normalize the line. Once we have normalized the line we can get its direction d_E , momentum m and closes point to the origin O_E .

$$d_E = \mathbb{O}\mathbb{N}(L) \cdot E \quad (33)$$

$$m = (0.5e_\infty - e_0) \cdot \mathbb{I}\mathbb{N}(L) \quad (34)$$

$$O_E = -d_E \cdot mI_E \quad (35)$$

5 Conformal Geometric Hough Transform

The steps for the algorithm are the same as the RHT. These steps can be described as follows:

1) Transform the euclidean points in the cloud to conformal points. Then randomly take sets of 4 points, x_{ci} .

2) Do the wedge product between the first 3 points to get a circle $\mathbb{O}\mathbb{N}(Z)$. Do $\mathbb{O}\mathbb{N}(S) = \mathbb{O}\mathbb{N}(Z) \wedge x_{c4}$.

2.1) If $\mathbb{O}\mathbb{N}(S)$ is 0 then x_{c4} lies on $\mathbb{O}\mathbb{N}(Z)$. Do the wedge product between $\mathbb{O}\mathbb{N}(Z)$ and e_∞ . If the result is 0, $\mathbb{O}\mathbb{N}(Z)$ is a line, otherwise a circle.

2.2) If $\mathbb{O}\mathbb{N}(S)$ is not 0 then do the wedge product between $\mathbb{O}\mathbb{N}(S)$ and e_∞ . If the result is 0, $\mathbb{O}\mathbb{N}(S)$ is a plane, otherwise a sphere.

3) After we detect the entity that the points x_{ci} form, we must eliminate two ambiguities. The first one is the scale factor and the second is a variant sign caused by the anticommutative behavior of the wedge product between 2 vectors. To eliminate these ambiguities we work in \mathbb{IPNS} for the sphere and plane. A division by the $\pm\alpha$ obtained in equation 24, solves both ambiguities for the sphere. For the plane we can get $\alpha = |n_E|$ to solve the scale factor. For the variant sign we use the function atan2 with n_E , because it can distinguish between diametrically opposite directions. The function atan2 has the interval $[-\pi, \pi]$, so the sign of the angle obtained is used to eliminate the variant sign. In the only exception to this method, where $X = Y = 0$, we use the sign of Z . To eliminate these ambiguities we work in \mathbb{OPNS} for the line and circle. For the line we get α with equation 32. We also solve the variant sign with the function atan2 and its direction d_E . The circle can be converted to a plane and use the same steps.

Once we have discarded the ambiguities we search for the candidate in its correspondig dinamyc accumulator array. If the candidate is found increase the accumulator by one. Otherwise add a new cell for the candidate and set its accumulator to one.

4) If an accumulator surpasses a threshold k we check if it exists in the cloud. If it exist we added it to a list of detected entities and delete all the points that belong to it from the set. To eliminate those points we comput their distance to the entity to see if they are close enough with the next formulas:

$$D = ||C_E - x_E| - r| \quad (36) \qquad D = |x_C \cdot \mathbb{IN}(P)| \quad (37)$$

Equation 37 is also valid for the line. With the circle we use both.

There are some minimal changes to apply this algorithm in the planar case. The first one is that we only detect lines and circles, so we take sets of 3 points instead of 4. We only have 2 coordinates, X and Y , so Z is set to 0 by default. The elimination of the ambiguities of the circle will be diferent, because all the circles lie on the image plane $\mathbb{IN}(Pimg) = 0e_1 + 0e_2 + e_3 + 0e_\infty$. If we obtain the plane of the circle this will be like $\mathbb{IN}(P) = \pm\alpha e_3$, then both ambiguities can be discarded with a division by $\pm\alpha$.

6 Analysis

One good reason to use CGA to attack this problem is the parametric equations. They are more simple than the regular equations. For example the equations used for circles in 2D with center (a, b) and radius r with 3 given points are:

$$m_a = \frac{y_2 - y_1}{x_2 - x_1} \quad (38) \qquad m_b = \frac{y_3 - y_2}{x_3 - x_2} \quad (39)$$

$$a = (m_a(m_b(y_1 - y_3)) + m_b(x_1 + x_2)) - \frac{m_a(x_2 + x_3)}{2(m_b - m_a)} \quad b = \frac{-a}{m_a} + \frac{x_1 + x_2}{2m_a} + \frac{y_1 + y_2}{2} \quad (41)$$

$$r = \sqrt{(x - a)^2 + (y - b)^2} \quad (42)$$

These equations are more complicated than the equation used to represent a circle in $\mathbb{O}PNS$. Other advantage is that the $\mathbb{O}PNS$ representations are related between them. For instance, the equations of the circle and line are the same, the result only depends of the position of the points, this also occurs with the sphere and plane. Also the result of the equation with tree points partially solves the one with four. As has been stated in other works [10], to maintain this advantage we need an optimal way to implement the equations in CGA. For this purpose we use GAALOP [11]. GAALOP is a precompiler for C/C++ and OpenCL. This tool allows to make an efficient implementation. It reduces the storage space by using only the space needed to store the nonzero coefficients of the multivectors, and also ignores all the unnecessary operations.

In the worst case scenario this algorithm has a computational complexity of $\mathcal{O}\binom{n}{s}$, where n is the numbers of points in the cloud, and s is the number of points in each set, 4 for 3D cloud points and 3 in the planar case. For comparison this is similar to $\mathcal{O}(n^s)$ but discarding permutations and repetitions. This is reduced by only taking a sample of the points from the cloud and eliminate those that already has been detected. In the 3D case other viable option will be to use a deep segmentation and then apply the algorithm to each zone.

7 Experimental Results

For the first experiments we used 2D images. In the first we take the picture of a candy, see figure 3a, to detect the circle shaped candy and its stick. We first apply a gaussian filter to reduce noise and then the Canny [12] algorithm to detect borders and then our algorithm. In the figure 3b, we observe the detected circle in color green and the stick lines in color red.

For the next image we use the picture of a bicycle, see figure 3c. The original image shows a bicycle in a white background. The algorithm was able to detect the 2 weels and 3 lines that form the frame.

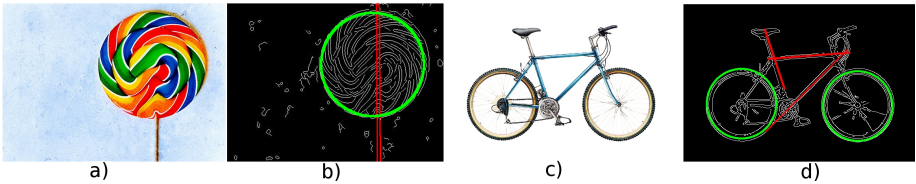


Fig. 3. Experiments with the candy and bicycle images. Detected lines are in red and circles in green.

In the last experiment, we used a RGB-D sensor, see figure 4.

As proposed in the analysis section, we use depth segmentation to reduce the computational cost, see figure 5. We also use the Canny algorithm in the RGB image to reduce the number of points, see figure 5. By doing this we have been able to detect a sphere, the ball, and two planes, the table and the wall.

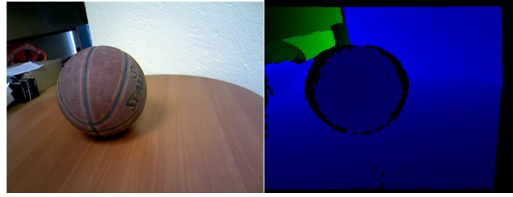


Fig. 4. (left) The RGB image obtained with the sensor. (right) the depth image.

To delimitate the table, the quality of the points were not enough to detect a circle, but by using the parameters of the detected plane and the points that are close to it as constrains, we can get an aproximation. The final result is visualized using the software Webots [13], see figure 5.

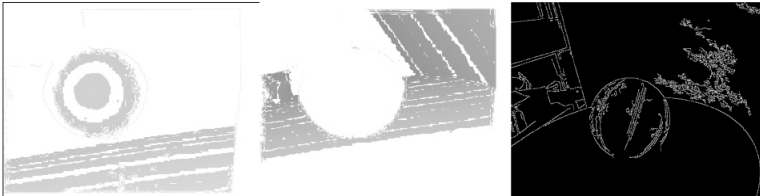


Fig. 5. Depth segmentation and edge image used for the detection process

Although the aproximation of the table is good, we observe that it overlaps with the wall and ball. We can correct this with the use of conformal transformations. In order to do this we construct a translator

$$T = 1 + \frac{\lambda}{2} n_E e_\infty \tag{43}$$

Where n_E is the direction of the translation, in this case the norm of the planes, and λ is the overlapping distance. We can see the result in figure 6.

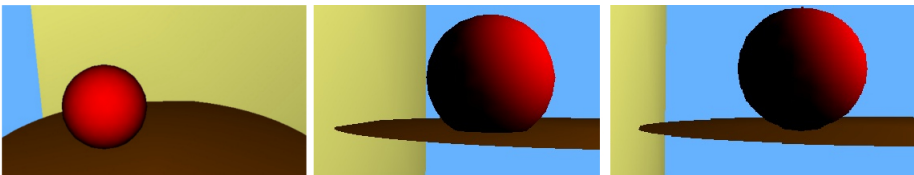


Fig. 6. (Left)Detection of the gemetric entities. (Middle)Angle used to show the error in the aproximation of the table. (Right)Rectification of the table.

8 Conclusion

In this work we have presented the advantages of use CGA to implement the Hough Transform. This mathematical framework let us detect different kind

of shapes with very simple equations even in 3D. We must remark the representations of the entities are vectors and also have an algebraic and geometric interpretations that can be used for detection algorithms at higher levels of complexity.

Future development of the algorithm will be focused in solving the high computational cost. Other extension to be developed is to work in higher dimension algebras to detect more complex entities, e.g. $\mathbb{G}_{6,3}$ to detect ellipses.

References

1. Hough, P.V.C.: Machine Analysis of Bubble Chamber Pictures. In: Proc. Int. Conf. High Energy Accelerators and Instrumentation (1959)
2. Duda, R.O., Hart, P.E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Comm. ACM* 15, 11–15 (1972)
3. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* 13(2), 111–122 (1981)
4. Shapiro, L., Stockman, G.: *Computer Vision*. Prentice-Hall, Inc. (2001)
5. Hestenes, D., Sobczyk, G.: *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. D. Reidel, Dordrecht (1984)
6. Bayro-Corrochano, E.: *Geometric Computing: For Wavelet Transforms, Robot Vision, Learning, Control and Action*. Springer, London (2010)
7. Perwass, C.: *Geometric Algebra with Applications in Engineering*. Springer, Heidelberg (2009)
8. Li, H., Hestenes, D., Rockwood, A.: Generalized homogeneous coordinates for computational geometry. In: Sommer, G. (ed.) *Geometric Computing with Clifford Algebra*, pp. 27–52. Springer (2001)
9. Kultanen, P., Xu, L., Oja, E.: Randomized Hough transform (RHT). In: *Proceedings of the 10th International Conference on Pattern Recognition, ICPR 1990, Atlantic City, USA, June 16–21, vol. 1*, pp. 631–635 (1990)
10. Li, Z., Hong, X., Liu, Y.: Detection Geometric Object in the Conformal Geometric Algebra Framework. In: *Proceedings of the 2011 12th International Conference on Computer-Aided Design and Computer Graphics, CADGRAPHICS 2011*, pp. 198–201. IEEE Computer Society, Washington, DC (2011)
11. Hildenbrand, D., Pitt, J., Koch, A.: Gaalop High Performance Parallel Computing based on Conformal Geometric Algebra. In: Bayro-Corrochano, E., Sheuermann, G. (eds.) *Geometric Algebra Computing for Engineering and Computer Science*, ch. 22, pp. 477–494 (2010)
12. Canny, J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*(6), 679–698 (1986)
13. Michel, O.: Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems* 1(1), 39–42 (2004)