Conformal Geometric Method for Voting

Gerardo E. Altamirano-Gómez and Eduardo Bayro-Corrochano

Department of Electrical Engineering and Computer Science, Cinvestav Guadalajara,
Av. del Bosque 1145, Colonia el Bajío, 45019, Zapopan, Jalisco, México
{galtamira,edb}@gdl.cinvestav.mx

Abstract. In this paper, we introduce a novel geometric voting scheme that extends previous algorithms, like Hough transform and tensor voting, in order to tackle perceptual organization problems. Our approach is grounded in three methodologies: representation of information using Conformal Geometric Algebra, a local voting process, which introduce global perceptual considerations at low level, and a global voting process, which clusters salient geometric entities in the whole image. Since geometric algebra is the mathematical framework of our approach, our algorithm infers high-level geometric representations from tokens that are perceptually salient in an image.

Keywords: Perceptual organization, Conformal Geometric Algebra, Tensor voting, Hough transform.

1 Introduction

When humans look at a scene, our visual system extracts features that are prominent, or attract our attention; this features are called *salient structures* [1]. In computer vision systems, its extraction is useful for high-level activities like object recognition, 3D reconstruction, or for taking decisions in perception and action systems. In this way, *perceptual organization* consists in obtaining meaningful representations from row data, and its principles for the human visual system are stated in the Gestalt psychology theory [2], [3].

From the computer science perspective, a set of tokens with a common property supports a specific structure; this is the idea behind voting schemes. For example, Hough transform [4], [5], is a voting technique that produces analytic representations of features in an image, and the voting procedure is implemented as a counter in an accumulator cell. Similarly, in the tensor voting framework [6] each token receives a vote from tokens within its neighbourhood. Magnitude and orientation of the vote are codified in a tensor. The orientation represents the normal of a token, while the magnitude of the vote is assigned by a voting pattern defined according to Gestalt principles.

In this paper, we address the perceptual grouping problem by proposing a novel voting scheme with a geometric approach. Our algorithm combines the biological motivation of inferring salient structures that satisfy specific perceptual properties, and a geometric representation of them. Thus, it extends Hough

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 802–809, 2014.

transform by adding voting patterns, and including a local and global voting process. In addition, it extends tensor voting scheme, since we codify geometric entities, e.g. lines or circles, using multivectors of conformal geometric algebra, instead of the normal of a token in a tensor. Furthermore, our voting scheme is presented as a geometric computing process in which the voting is computed by intersecting geometric elements.

The organization of the paper is as follows: Section 2 presents an introduction to Conformal Geometric Algebra, the voting scheme and the corresponding complexity analysis is presented on Section 3. Then, we show experimental results in Section 4, followed by conclusions in Section 5.

2 Conformal Geometric Algebra

Conformal Geometric Algebra (CGA) allows the representation of geometric entities and their properties, by embedding an euclidean space $\mathbb{R}^n$ in a higher dimensional vector space $\mathbb{R}^{n+1,1}$. Here, we summarize the construction of CGA, for a detailed study see [7].

Let $\mathbb{R}^{n+1,1}$ be a real vector space, which has associated a geometric algebra $\mathcal{G}_{n+1,1}$, then its vector basis satisfy: $e_+^2 = 1$, $e_-^2 = -1$, and $e_i^2 = 1$, for $i = 1, \ldots, n$. In addition, the following properties are satisfied: $e_+ \cdot e_- = 0$, $e_i \cdot e_+ = 0$, and $e_i \cdot e_- = 0$, for $i = 1, \ldots, n$.

Then, we define two null basis: $e_\infty = e_- + e_+$, and $e_0 = 0.5(e_- - e_+)$, with the properties: $e_0^2 = e_\infty^2 = 0$, and $e_\infty \cdot e_0 = -1$.

The set of all null vectors in $\mathbb{R}^{n+1,1}$ is called the *null cone*, and its intersection with an hyperplane with normal e_∞, and containing point e_0, is a surface called *horosphere*, defined as:

$$\mathbf{N}_e^n = \{x_c \in \mathbb{R}^{n+1,1} : x_c^2 = 0, x_c \cdot e_\infty = -1\} \ . \tag{1}$$

Now, all points that lie on the horosphere are called *conformal points*, represented by:

$$x_c = x_e + 0.5x_e^2 e_\infty + e_0 \ , \tag{2}$$

where, $x_e \in \mathbb{R}^n$. In addition, three unit pseudoscalars are defined: I_e for $\mathcal{G}_n$, E that represents the Minkowski plane, and I for $\mathcal{G}_{n+1,1}$:

$$I_e = e_1 e_2 \ldots e_n ; E = e_\infty \wedge e_0 ; I = I_e \wedge E \ . \tag{3}$$

Finally, Table 1 summarizes the representation of geometric entities in CGA $\mathcal{G}_{3,1}$, where IPNS stands for inner-product null space, and OPNS stands for outer-product null space.

3 Voting Scheme

Here, we present a geometric framework for automatic perceptual organization. Its essential components can be summarized in three methodologies: to represent information using CGA, a local voting process, which extract salient geometric entities supported in a local neighbourhood, and a global voting process, which clusters the output obtained by the local voting process.

Table 1. Representation of geometric entities in CGA $\mathcal{G}_{3,1}$

Entity	IPNS	OPNS
Circle	$S = c_e + 0.5(c_e^2 - \rho^2)e_\infty + e_0$ $\rho =$ radius	$S^* = x_{c1} \wedge x_{c2} \wedge x_{c3}$
Line	$l = nI_e + e_\infty d_H I_e$ $n = x_{e1} - x_{e2}$ $d_H = x_{e1} \wedge x_{e2}$	$l^* = e_\infty \wedge x_{c2} \wedge x_{c3}$
Point Pair	$PP = S_1 \wedge S_2$	$PP^* = x_{c1} \wedge x_{c2}$

3.1 Representation of Information Using CGA

Let $\mathbb{R}^n$ be a real n-dimensional vector space; then, a token, denoted as t, is represented as a multivector of CGA $\mathcal{G}_{n+1,1}$. In the same way, a perceptual structure of $\mathbb{R}^n$ is represented as a geometric entity of CGA $\mathcal{G}_{n+1,1}$, and it is denoted by F. The possible combinations between tokens and perceptual structures can constitute *flags* [8]. Then, a token t on a perceptual structure F satisfies: $F \cdot t = 0$. Consequently, a set of tokens $\{t_1, t_2, \cdots, t_n\}$, that satisfy $F \cdot t = 0$, define a minimum for:

$$\frac{1}{N} \sum_{i=1}^{N} W_i(a_1, \cdots, a_m)(F \cdot t_i)^2 F^{-2} \ , \tag{4}$$

where W_i is a function that maps a set of parameters $\{a_1, \cdots, a_m\}$ to a scalar value, and acts like a weight for each flag $F \cdot t_i$.

Equation 4 suggests that we can estimate F using fitting techniques; even though, we prefer to follow a geometric approach to construct F. In this way, we solve non-linear cases of Equation 4, without using non-linear fitting techniques.

In this work, we consider an image as a vector space $\mathbb{R}^2$. In addition, we restrict our treatment to the simplest token: the pixel, and we analyse two specific perceptual structures: circle and line. Then, a pixel p_i is represented as a conformal point of CGA, and perceptual structures like circles and lines in the image, are represented as elements of CGA $\mathcal{G}_{3,1}$, according to Table 1.

3.2 Local Voting

Let p_0 and F be elements of CGA $\mathcal{G}_{n+1,1}$; they represent a token and a perceptual structure, respectively. In order to compute F, all tokens in the neighbourhood of p_0 must cast a vote.

Now, we introduce the fundamental difference between our voting scheme and others; we propose that the vote of a token p_i on p_0 has two parts: geometric structure, and perceptual saliency. Where *geometric structure* is represented by a multivector of CGA $\mathcal{G}_{n+1,1}$, and codifies properties like dimension, orientation, direction, and scalar magnitude of a geometric entity; while *perceptual saliency* is a function that codifies properties such as proximity, co-curvilinearity, and constancy of curvature. In this way, the vote cast by a token is not a number that is sum on a geometric entity, but is a geometric entity itself.

Geometric Structure. Let p_0 be the pixel that is receiving a vote, without loss of generality, we set its euclidean coordinates to the origin. Then, the set of pixels on the neighbourhood of p_0 is defined as:

$$P_0 = \{p_i : d_{0i} < \textit{neighborhood size}\}, \tag{5}$$

where d_{0i} is the euclidean distance between pixels p_i and p_0. The coordinates of pixel p_i in the image are: (u_i, v_i), taking p_0 as origin of the coordinate system. Moreover, each pixel p_i of P_0, together with pixel p_0 defines a point pair:

$$PP_{0i} = (p_0 \wedge p_i)^* \,. \tag{6}$$

In addition, each pixel on P_0, together with point pair PP_{0i}, define a pencil of circles; where PP_{0i} is the vertex of the pencil, and each circle is represented by:

$$S_{0ij} = (p_0 \wedge p_i \wedge p_j)^* \,;\; p_i, p_j \in P_0. \tag{7}$$

Lemma 1. *For a pencil of circles, in which all circles meet in two real base points, the centers of the circles lie on a line.*

Then, pixels p_0, and p_i support all circles S_{0ij}; using Lemma 1, we represent all this circles by the equation of the line containing all centres:

$$l_i = \frac{u_i}{|p_{ie}|}e_1 + \frac{v_i}{|p_{ie}|} + \frac{|p_{ie}|}{2}e_\infty \,. \tag{8}$$

Thus, l_i is the vote cast by pixel p_i on pixel p_0, since it codifies all possible circles containing p_0 and p_i.

Perceptual Saliency. Now, we are going to assign a density value that represents the perceptual saliency of the geometric structure. In [6], a vector field is proposed to codify perceptual properties according to Gestalt theory. Such vector field associates to each pixel, a direction vector and a saliency value. Since we have already define the geometric structure of the vote, we are going to take only the saliency value of a stick vector field.

Definition 1. *A* ***line with density****, is a set of points together with a function, that assigns a scalar value to each element of the set.*

Using CGA $\mathcal{G}_{3,1}$, a line with density is represented by:

$$L = \{p_c : p_c \cdot l_c = 0\},\; W : \mathcal{R}^m \rightarrow \mathcal{R} \,. \tag{9}$$

where p_c and l_c are a point, and a line, respectively, in conformal representation; then, L is a set of conformal points and W is a function that assigns a value to each element of set L.

Thus, we map each point p_i on P_0 to a line l_i, and we obtain a set of lines denoted as L_0. After that, we assign a density value to each point of l_i using the following function [6]:

$$W(u,v) = \exp\left(-\frac{s^2 + c\rho^2}{\sigma^2}\right), \; s = \frac{\theta d}{\sin\theta}, \; \rho = \frac{2\sin\theta}{d} \; . \tag{10}$$

In addition, we align the y-axis of the stick field with the point with coordinates $(u,\ v)$. Hence, we rotate the stick field an angle $\phi = \tan^{-1}(-u/v)$, thus:

$$\theta = \tan^{-1}\left(\frac{u\cos\phi - v\sin\phi}{-u\sin\phi - v\cos\phi}\right), \; d = \sqrt{u^2 + v^2} \; . \tag{11}$$

Finally, the vote cast by a pair of pixels p_i, and p_j on p_0, are lines with density l_i, and l_j respectively. The intersection of this lines, is a point that represents the center of the circle passing through points p_0, p_i and p_j. We calculate the intersection using equation:

$$C_{0ij} \wedge e_\infty = (l_i \wedge l_j)^* \tag{12}$$

and the result, is a point with density:

$$W_{0ij} = W(u_i, v_i) + W(u_j, v_j). \tag{13}$$

Thus, the voting process consists in calculating the intersection of each pair of lines in L_0. After that, using the DBSCAN algorithm [9] we cluster the resulting points, compute the mean of each cluster, and select the mean of the cluster with maximum density.

3.3 Global Voting

The local voting process delivers a salient geometric structure for each pixel. The global voting process consists in cluster this structures, which is done using DBSCAN algorithm. Finally, we extract the mean of each cluster.

3.4 Complexity Analysis

The input to our algorithm is a binary image, with a set of k white pixels, and the output is a set of r geometric entities in conformal representation.

The local voting step selects a white pixel, and define a set P_0 using Equation 5. Let m be the size of set P_0, then the computing of intersections takes $O(m^2)$; their clustering is done with DBSCAN algorithm, implemented with R-trees [10], and takes $O(m\ lg(m))$. In addition, extracting the mean of the cluster with highest density takes $O(m)$. Since local voting is executed for each white pixel in the image, then the local voting step takes $O(km^2)$.

In addition, the global voting step takes as input a set of k geometric entities, one for each pixel, and gives as output a set of r geometric entities. Clustering with DBSCAN algorithm takes $O(k\ lg(k))$, and computing the mean of each cluster takes $O(k)$. Thus, our algorithm has a complexity of $O(km^2 + k\ lg(k))$.

3.5 Relationship with Hough Transform, Tensor Voting, and Fitting Approaches

Our method is a flexible technique that extends previous voting schemes. Then, changing some parameters in the local voting process we obtain equivalent results to Hough transform, tensor voting, and fitting techniques. For example, if we set the saliency field to a constant value, the output of our algorithm is equivalent to that obtained by Hough transform, applied in a local neighbourhood. Moreover, if the saliency is set with a stick field, the normal of the pixel is the unit vector that goes from the pixel that receives the vote, to the center of the circle with maximum density. Then, the result obtained by the local voting step is equivalent to that obtained with tensor voting; and the global voting step should be replace by a marching cubes algorithm as is done by the tensor voting framework.

Furthermore, for a set of tokens, if we use a fitting method to estimate the must likely geometric structure, by minimizing Equation 4, we will face non-linearity problems, and a bias due to outliers. Using our geometric approach, we take out the outlier pixels using a saliency field, and compute the geometric structure using the rest of the points.

4 Experimental Analysis

In this section we present a set of experiments designed to work as a proof-of-concept.

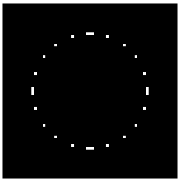
4.1 Experiments with Synthetic Images

Experiment 1: *Input with an incomplete geometric entity.* Figure 1(a) shows a set of pixels that lie on a circle; using our algorithm we are able to recover the equation that describe the circle, with sub-pixel error.

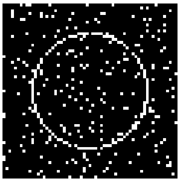
Experiment 2: *Input with noise.* Figure 1(b) shows an image that contains a complete circle, to which we add a 7% of noise, that is, each pixel has a 0.07 probability of becoming an erroneous site. After applying our algorithm, we obtain the equation of the circle with sub-pixel error. Noise is removed due to the saliency field used in the local voting step.

Experiment 3. *Input with occluded geometric entities.* Figure 1(c) shows an image that contains a set of lines that are occluded by a circle. The output of our algorithm is a set of 5 equations, one for each line and one for the circle, as Figure 1(g) shows. In this case, the global voting step of our algorithm, allows to relate the pixels that lie on the same geometric entity, even though they are in opposite sides of an image, or some pixels are occluded by other element. Lines and circles are found with sub-pixel error.

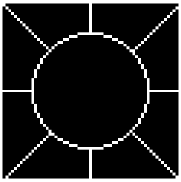
Experiment 4. *Input with illusory contours.* Figure 1(d) shows the Kanizsa square: four incomplete circles are shown, and a square can be seen in the image, even thought is not explicitly draw. After applying Canny edge detector, our algorithm extracts the equations of the 4 circles, and the equations of the lines that describe the illusory contours of the Kanizsa square.



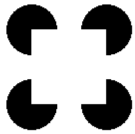
(a) Input image for Experiment 1.



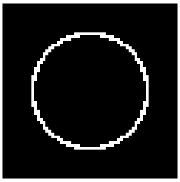
(b) Input image for Experiment 2.



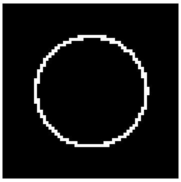
(c) Input image for Experiment 3.



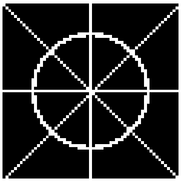
(d) Input image for Experiment 4.



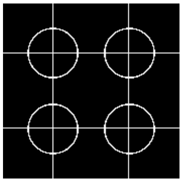
(e) Output image for Experiment 1.



(f) Output image for Experiment 2.

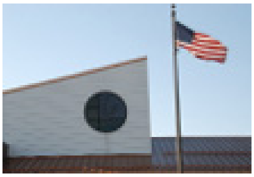


(g) Output image for Experiment 3.



(h) Output image for Experiment 4.

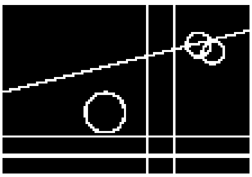
Fig. 1. Results obtained by CGAV with synthetic images



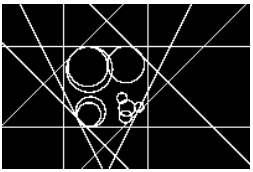
(a) Input image for Experiment 5.

(b) Input image for Experiment 6.

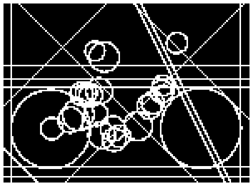
(c) Input image for Experiment 7.



(d) Output image for Experiment 5.



(e) Output of CGAV for Experiment 6.



(f) Output of CGAV for Experiment 7.

Fig. 2. Results obtained for images with real objects

4.2 Experiments with Images with Real Objects

Figures 2(a), 2(b), and 2(c) are images with real objects. We apply a pre-processing step, which consist in a Canny edge detector and a mean filter, and after that, we use our algorithm to extract salient geometric structures.

Figure 2(d) shows the output for Experiment 5, our algorithm extracts salient geometric entities, so that we obtain a description of the scene using 3 circles and 5 lines. Since input image uses 470 pair of coordinates to describe the scene, we have obtained a data compression ratio of 39 : 1 approximately. We note that the flag, which contour in the image looks like a non-linear curve, is describe by an expansion of spherical wavelets [8]. A similar result is obtained in Experiments 6 and 7, where Figures 2(e), and 2(f) shows how our algorithm makes a non-uniform sampling with circles and lines, in order to describe a non-linear curve. This behaviour is similar to algorithms like marching spheres [11], or marching cubes [12].

5 Conclusion

The algorithm presented, is a voting scheme that unifies several approaches in a single geometric framework. The results show the ability of our algorithm to generate high-level geometric representations of salient features in images, even though they present incomplete or noise data, illusory contours, or non-linear surfaces. Work in future will focus on the generalization to other geometric entities, the design of perceptual fields, and the generalization of this approach to higher dimensions.

References

1. Sha'asua, A., Ullman, S.: Structural Saliency: The Detection Of Globally Salient Structures using A Locally Connected Network. In: IEEE International Conference on Computer Vision, pp. 321–327. IEEE Press, Tampa (1988)
2. Ellis, W.: A source book of Gestalt psychology. The Gestalt Journal Press, Inc., Gouldsboro (1997)
3. Koffka, K.: Principles of Gestalt Psychology. Harcourt/Brace, New York (1935)
4. Hough, P.: Method and means for recognizing complex patterns. USA Patent 3069654 (1962)
5. Duda, R., Hart, P.: Pattern Recognition and Scene Analysis. Wiley (1973)
6. Mordohai, P., Medioni, G.: Tensor Voting. A perceptual organization approach to computer vision and machine learning. Morgan & Claypool Publishers (2006)
7. Li, H., Hestenes, D., Rockwood, A.: Generalized Homogeneous Coordinates for Computational Geometry. In: Sommer, G. (ed.) Geometric Computing with Clifford Algebras, pp. 27–59. Springer, Heidelberg (2001)
8. Bayro-Corrochano, E.: Geometric Computing: For Wavelet Transforms, Robot Vision, Learning, Control and Action. Springer, London (2010)
9. Ester, M., Kriegel, H., Jrg, S., Xu, X.: Grid Information Services for Distributed Resource Sharing. In: AAAI 2th International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press, Portland (1996)
10. Guttman, A.: GR-Trees: A Dynamic Index Structure for Spatial Searching. In: 1984 ACM SIGMOD International Conference on Management of Data, pp. 47–57. ACM Press, Boston (1984)
11. Bayro-Corrochano, E., Rivera-Rovelo, J.: The Use of Geometric Algebra for 3D Modeling and Registration of Medical Data. J. Math. Imaging Vis. 34, 48–60 (2009)
12. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. In: 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 163–169. ACM Press, New York (1987)