2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Conformal Geometric Algebra Method for Detection of Geometric Primitives

Gerardo Altamirano-Gomez and Eduardo Bayro-Corrochano
Department of Electrical Engineering and Computer Science
Cinvestav Campus Guadalajara
Jalisco, Mexico
Email: galtamira@gdl.cinvestav.mx, edb@gdl.cinvestav.mx

Abstract—**In this paper, we present a geometric algebra approach for detection of geometric entities in images. Our algorithm is grounded on two methodologies: representation of geometric entities and perceptual properties using Conformal Geometric Algebra, and a voting scheme which is implemented using a clustering algorithm. Our method is applied in a hierarchical way, so that, we extract local and global information from images. Experimental results show the application of our approach to detection of circles, lines, complex structures of shape, and symmetry axis. In addition, we show an FPGA implementation that speed-up the execution time of the algorithm.**

I. Introduction

Perceptual organization is the process of grouping features arising from a common underlying cause [1], [2]. From a computer science perspective, a set of tokens that share certain property, is the support of a structure. This idea leads to the developing of voting schemes, where according to some criteria, each token cast a vote on certain structure; so that, the winner of the voting process is a salient structure.

A popular voting scheme is Hough Transform, which produce analytic representations of features, and the voting procedure is implemented a s a counter in an accumulator cell [3]–[5]. Another algorithm is called Tensor Voting, which uses a tensor representation to represent features, and uses a perceptual saliency function to codify Gestalt properties of proximity, co-curvilinearity, and constancy of curvature [6]–[8].

In this work, we use propose a generalization of voting schemes, using Conformal Geometric Algebra. The role of Geometric Algebra and Tensor Voting was for the first time published in [9] and [10]; motivated by these works, we have generalized the Tensor Voting framework to extract any kind of geometric entities or geometric flags, so that we can formulate complex perceptual saliency functions in terms of k-vectors of Conformal Geometric Algebra. In addition, our method can be applied in a hierarchical way, extracting multiple geometric entities at each level; after this pre-processing, machine learning methods can be also applied.

The organization of the paper is as follows: Section II presents an introduction to Conformal Geometric Algebra, our voting scheme is presented on Section III. Applications of the method for detection of complex shapes and detection of bilateral symmetry are presented in Sections IV and V, respectively. Finally, conclusions and future work are stated in Section VI.

II. Conformal Geometric Algebra

Geometric Algebra (GA) is a coordinate-free approach to geometry based on the algebras of Grassmann and Clifford. A specialized version, called Conformal Geometric Algebra (CGA), allows an homogeneous 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 [11], [12].

Let $\mathbb{R}^{n+1,1}$ be a real vector space, which is associated with geometric algebra $\mathcal{G}_{n+1,1}$, then its vector bases $\{e_1, \ldots, e_n, e_+, e_-\}$ 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 $i = 1, \ldots, n$.

The set of all $\binom{n+2}{k}$ elements produced by the geometric product of k linear independent vectors span a vector space, denoted by $\bigwedge^k V^{n+2}$. Each element of this space, is called k-*vector*, denoted by $\langle A \rangle_k$, where k indicates the grade. A linear combination of k-vectors of mixed grade is called *multivector*, expressed as follows:

$$A = \langle A \rangle_0 + \langle A \rangle_1 + \cdots + \langle A \rangle_{n+2} \,. \tag{1}$$

Next, we define two null basis: the point at infinity, $e_\infty = e_- + e_+$, and the origin, $e_0 = \frac{e_- - e_+}{2}$; which satisfy: $e_0^2 = e_\infty^2 = 0$, 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:

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

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

$$x_c = x_e + \frac{x_e^2}{2} e_\infty + e_0 \,, \tag{3}$$

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

$$I_e = e_1 e_2 \ldots e_n; E = e_\infty \wedge e_0; I = I_e \wedge E \,. \tag{4}$$

TABLE I
REPRESENTATION OF GEOMETRIC ENTITIES IN CGA $\mathcal{G}_{n+1,1}$.

Entity	$\mathbb{IPNS}$	$\mathbb{OPNS}$
Point (p_c)	$p_e + 0.5p_e^2 e_\infty + e_0$	$\bigwedge_{i=1}^{n+1} S_i$
Pair of Points (PP)	$\bigwedge_{i=1}^{n} S_i$	$p_{c1} \wedge p_{c2}$
Hypersphere (S)	$c_c - 0.5\rho^2 e_\infty$	$\bigwedge_{i=1}^{n+1} p_{ci}$
Hyperplane (π)	$n_e + d_H e_\infty$	$e_\infty \bigwedge_{i=1}^{n} p_{ci}$

TABLE II
GROUP OF TRANSFORMATIONS IN CGA $\mathcal{G}_{n+1,1}$.

Transformation	Expression
Inversion	$S = c_c - 0.5\rho^2 e_\infty$
Reflection	$L = n + de_\infty$
Translation	$T = 1 + dne_\infty$
Rotation	$R = \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right) l$
Transversion	$K = 1 + de_0$
Conformal	$G = KTR$

A. Representation of Geometric Entities

A geometric entity is defined by a set of points that satisfy a geometric constraint. Let $x \in \mathcal{G}_{p,q}^1$ be a vector, and let $\mathbf{X}_k \in \mathcal{G}_{p,q}^k$ be a k-blade; then, all vectors that satisfy $x \cdot \mathbf{X}_k = 0$, lie on the geometric entity that $\mathbf{X}_k$ represents. This set of vectors is called *inner product null space* (IPNS), denoted by $\mathbb{NI}(\mathbf{X}_k)$, thus:

$$\mathbb{NI}(\mathbf{X}_k) = \{x \in \mathcal{G}_{p,q}^1 : x \cdot \mathbf{X}_k = 0\} \ . \tag{5}$$

Similarly, we define an *outer product null space* (OPNS), denoted by $\mathbb{NO}(\mathbf{X}_k)$, as follows:

$$\mathbb{NO}(\mathbf{X}_k) = \{x \in \mathcal{G}_{p,q}^1 : x \wedge \mathbf{X}_k = 0\} \ . \tag{6}$$

Inner and outer null spaces have a dual relationship, which is summarized in the following equation:

$$(x \wedge \mathbf{X}_k)^* = (x \wedge \mathbf{X}_k) \cdot I^{-1} = x \cdot \mathbf{X}_k^* \tag{7}$$

where I^{-1} is the inverse of the pseudoscalar of $\mathcal{G}_{p,q}$.

Table I summarizes the representation of geometric entities in CGA $\mathcal{G}_{n+1,1}$; where ρ and c_c represents the radius and center of a hypersphere in conformal representation, n_e represents the normal of a hyperplane, and d_H its Hesse distance.

B. Conformal Transformations

A transformation of a geometric figure is said to be conformal if it preserves the angles of the figure. To characterize orthogonal transformations, it is convenient to introduce the concept of *versor* [13]. A versor, G, is any multivector of a geometric algebra that can be factorized into a geometric product:

$$G = v_1 v_2 \ldots v_k, k \leq n \tag{8}$$

of vectors $v_i \in \mathbb{R}^{p,q}$. In CGA, a conformal transformation is applied as a sandwiching product:

$$O' = G\ O\tilde{G}, \tag{9}$$

where $O \in \mathbb{R}^{n+1,1}$ is any object listed in Table I, and G is a versor. Table II shows the group of transformations available in CGA $\mathcal{G}_{n+1,1}$.

III. CONFORMAL GEOMETRIC ALGEBRA VOTING SCHEME

In this section, we introduce a framework for automatic perceptual organization. The essential components of our approach are summarized in two methodologies: To represent information using CGA, and a voting process which is implemented using a clustering algorithm.

A. Representation of Information Using CGA

Let $\mathbb{R}^n$ be a real n-dimensional vector space; then, a token, denoted by t, and a geometric structure, denoted by F, are represented as multivectors of CGA $\mathcal{G}_{n+1,1}$. The possible combinations between tokens and geometric structures can constitute *flags* [12]. Then, a token t on a geometric structure F satisfies: $F \cdot t = 0$. Consequently, for a set of tokens $\{t_1, t_2, \cdots, t_N\}$, the flag, F, that satisfy $F \cdot t_i = 0$, for $i = 1, 2, \ldots, N$ define a minimum for:

$$\frac{1}{N}\sum_{i=1}^{N} W(a_1, \cdots, a_m, t_i)\frac{(F \cdot t_i)^2}{|F|^2} \ , \tag{10}$$

where W is a function that maps a set of parameters $\{a_1, \cdots, a_m, t_i\}$ to a scalar value. We call W a *perceptual saliency function*, since it is used to codify perceptual properties.

Equation (10) is the key to generalize voting schemes, since the inner product relates a set of tokens with a geometric structure via an incidence relationship; moreover, each product is limited by a function W, which introduce perceptual restrictions on F. Thus, we have to design flags and perceptual saliency functions according to the kind of information that we want to extract.

For example, for detection of circles and lines, we can use the incidence relationship of point-line flag and point-sphere flags:

$$L \cdot p = 0, \tag{11}$$

$$S \cdot p = 0. \tag{12}$$

In this case, given a set of points, we can compute L or S can be from subsets of 3 points, using the OPNS representation of Table I. Alternatively, we can map each pair of points to a line equidistant to both points, and the intersection of each pair of lines is the center of a circle.

The perceptual saliency for circles or lines, is assigned using the decay function of the Tensor Voting algorithm [8]:

$$W(s, \rho, c, \sigma) = \exp\left(-\frac{s^2 + c\rho^2}{\sigma^2}\right), \tag{13}$$

where c controls the degree of decay with curvature, σ determine the neighbourhood size of the voting, s represents the arc length, and ρ the degree of curvature of the circle, see Figure 1.

On the other hand, for detecting bilateral symmetry, we can use the conformal representation of a line to represent the

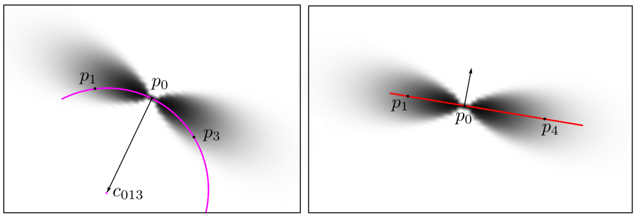

Fig. 1. Saliency assignment. For each geometric structure, a saliency field is constructed using (13).

symmetry axis, and for computing it, we can use a pair of circles, lines, or points, as follows:

For each pair of points, its symmetry axis is:

$$l_i = -\frac{u_i}{|p_{ie}|}e_1 - \frac{v_i}{|p_{ie}|} + \frac{|p_{ie}|}{2}e_\infty, \tag{14}$$

where the first point is the origin of the coordinate system, (u_i, v_i) are the coordinates of the second point, and $|p_{ie}| = \sqrt{u_i^2 + v_i^2}$.

In addition, for each pair of circles, its symmetry axis is:

$$L = [(S_1 \wedge S_2)^* \wedge e_\infty]^*, \tag{15}$$

where S_i and S_j are two different circles with equal radius.

On the other hand, for each pair of parallel lines l_i and l_j, the symmetry axis is given by:

$$L_d = T_d\ l_i\ \tilde{T}_d, \tag{16}$$

where $d = |(l_i \wedge l_j)^*|n_i$, n_i is the normal of the lines, and T is a translator of CGA $\mathcal{G}_{3,1}$.

For lines that intersect with each other, we have two symmetry axis:

$$L_\theta = R_\theta\ l_i\ \tilde{R}_\theta, \tag{17}$$

where R is a rotor of CGA $\mathcal{G}_{3,1}$, and θ is the angle between the lines, computed as follows:

$$\theta_1 = \arctan\left(\frac{|l_i \wedge l_j|}{|l_i \cdot l_j|}\right),\ \theta_2 = \theta_1 + \frac{\pi}{2}. \tag{18}$$

Finally, the perceptual saliency of a symmetry axis can be assigned using the following equation:

$$W(l, D) = \sum_{p_{ic} \in D} \exp\left[\frac{(l\ p_{ic}\ \tilde{l}) \cdot p_{jc}}{\sigma^2}\right], \tag{19}$$

where $D = \{p_{1c}, ..., p_{kc}\}$, is the set of conformal points that support the symmetry axis, l is the symmetry axis applied as a versor transformation on a point p_{ic}, p_{jc} is the nearest neighbour to point $l\ p_{ic}\ \tilde{l}$, and σ is a parameter that sets the size of the neighbourhood in which we search for point p_{jc}.

B. Communication of information

This stage has two parts: a local voting process, which extracts salient geometric entities supported in a local neighbourhood, and a global voting process, which clusters the output obtained by the local voting process.

1) Local Voting: Given a perceptual saliency function W, and a set of tokens $T = \{t_1, t_2, \cdots, t_N\}$, the local voting step consist in: selecting a token, $t_0 \in T$, defining a subset, T_0, that contains all tokens in the neighbourhood of t_0, and computing the geometric structure, F, that minimizes (10) for the tokens in subset T_0. To find F, we apply a voting methodology to take the outliers out, and compute F using the rest of the tokens.

The voting procedure is performed as follows: Let t_0 be the selected token, then each token on its neighbourhood casts a vote in the form of a perceptually salient geometric element.

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

Using CGA, a perceptually salient geometric element is represented by:

$$\bar{F} = \{p_c : p_c \cdot F = 0\},\ W : \mathcal{R}^m \to \mathcal{R}, \tag{20}$$

where p_c and F are a point and a geometric element, respectively, in conformal representation; whereas $\bar{F}$ is a set of conformal points, and W is a function that assigns a scalar value to each element of set $\bar{F}$. Therefore, F codifies the geometric structure of an object, while the function W codifies its perceptual saliency.

Thus, the voting consists in mapping each token on the neighbourhood of t_0 to a perceptually salient geometric element. In the voting space, each geometric structure F is a point that has associated a perceptual saliency value or density. Then we use DBSCAN algorithm [14] to separate votes, and cluster those that has similar geometric structure (e.g. hyperplanes with the same normal and Hesse distance, or hyperspheres with the same center and radius).

Next, we compute the perceptual saliency of each cluster:

$$\bar{W} = \sum_i W_i. \tag{21}$$

where W_i is the perceptual saliency of each geometric element in the cluster. Finally, we select clusters that surpasses a threshold value, or the cluster with highest perceptual saliency.

In contrast with the Tensor Voting algorithm, which extracts only the feature with maximum support, the use of a clustering technique in our algorithm allow us to extract several features at the same time.

2) Global Voting: The local voting process delivers a set of salient geometric structures for each token. Let $\bar{O} = \{O_1, \cdots, O_n\}$ be the output of the local voting process, where each element O_i is a salient geometric structure; then, the global voting process consists in grouping similar geometric structures using DBSCAN algorithm. Thus, $\bar{O}$ is partitioned into subsets $\bar{O}_1, \cdots, \bar{O}_q$, where each $\bar{O}_i$ is a cluster of geometric entities obtained by DBSCAN. Next, we compute the perceptual saliency of each cluster using Equation 21, and select clusters that surpasses a threshold value.

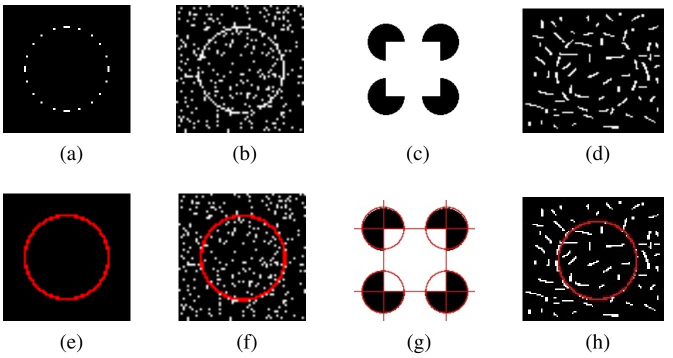

Fig. 2. Synthetic images (a, b, c, d) and the results obtained by our method (e, f, g, h)

IV. Application for Extraction of Circles and Lines in Images

The goal of this application is to find a representation of the contour of objects in images; hence, the input to our algorithm is an edge image. In addition, our method considers an image as a vector space $\mathbb{R}^2$, and each edge pixel is represented as a conformal point of CGA $\mathcal{G}_{3,1}$. The output is a set of perceptual structures, i.e. circles and lines, associated with a saliency value. These structures are represented as elements of CGA $\mathcal{G}_{3,1}$, according to Table I.

Figure 2 shows four images for which we have applied our voting scheme. Input images contain incomplete data (Figure 2a), noisy data[1] (Figure 2b), an image with illusory contours (Figure 2c), and an image with noise and an incomplete geometric entity (Figure 2d). Output of our algorithm is shown in Figures 2e, 2f, 2g, and 2h, where in each case, as opposite to any current algorithms, our method recovers simultaneously the equations of the circles and lines that are perceptually salient.

Figure 3 shows a set of images that contain real objects, the input images were taken from image databases [15], and [16]. For each input image we apply a pre-processing step, which consist in a Canny edge detector [17], and a mean filter; after that, we use our algorithm to extract salient geometric structures. The second row of Figure 3 shows the output of the local voting step, while the third row show the output of our algorithm for each image; salient structures has been superimposed to the edge images obtained after the pre-processing step.Unfortunately, we have not found a standard procedure yet to adjust the parameters of the DBSCAN algorithm and the threshold values for the local and global voting steps, and they were setup separately for each image.

The set of images contains objects which contours describe non-linear curves. Experimental results show that our algorithm makes a non-uniform sampling with circles and lines, in order to describe this objects; note that each geometric entity obtained by our algorithm is a *local descriptor of shape*, and the total output describes the object by a sort of an expansion of spherical wavelets [12].

[1]each pixel has a 0.09 probability of being an erroneous site.

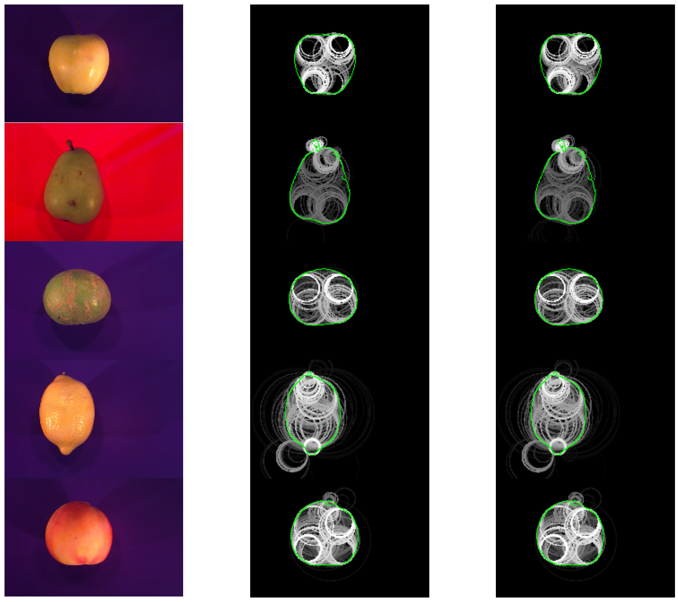

Fig. 4. Experimental results for local stage. Input images in the first column, the output obtained by the PC implementation in the second column, and the output obtained by the FPGA in the third one.

A. Speeding-up the algorithm in a Field-Programmable Gate Array (FPGA)

An FPGA is a reprogrammable device that can adopt the functionality of a digital circuit. They can process a large amount of data with a low power consumption, and allows the use of pipelines and parallelism to accelerate computations over batches of data.

In our voting algorithm, the must computational expensive process is the local voting step. For testing the speed-up and accuracy that can be obtained when implementing in an FPGA, the local voting process for detecting circles was implemented in a ZC706 evaluation board featured with the Xilinx FPGA Zynq 7000 based on an APSoC architecture running at 100 MHz.

The output obtained in the FPGA implementation has sub-pixel accuracy, and is three to four times faster than the PC implementation, see Table III. Results for a set of images is shown in Figure 4.

TABLE III
Execution times for PC and FPGA implementations [ms]

Platform	Apple	Pear	Avocado	Lemon	Peach
PC	1826	10627	1914	14309	3861
FPGA	317	4130	351	4200	930

V. Application for Detection of Bilateral Symmetry

Using the CGA voting scheme as building blocks, we have designed a three-level architecture for detection of bilateral

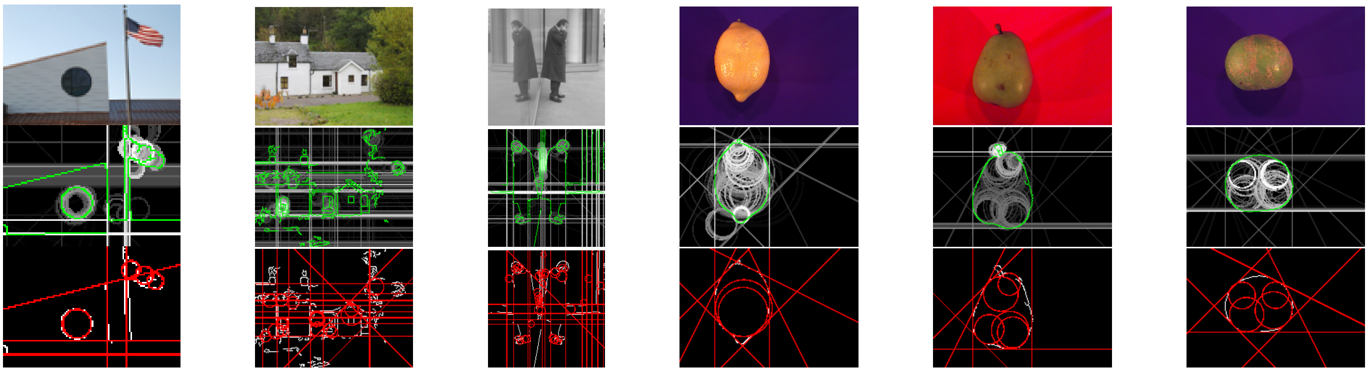

Fig. 3. Images that contain real objects and the results obtained by our method

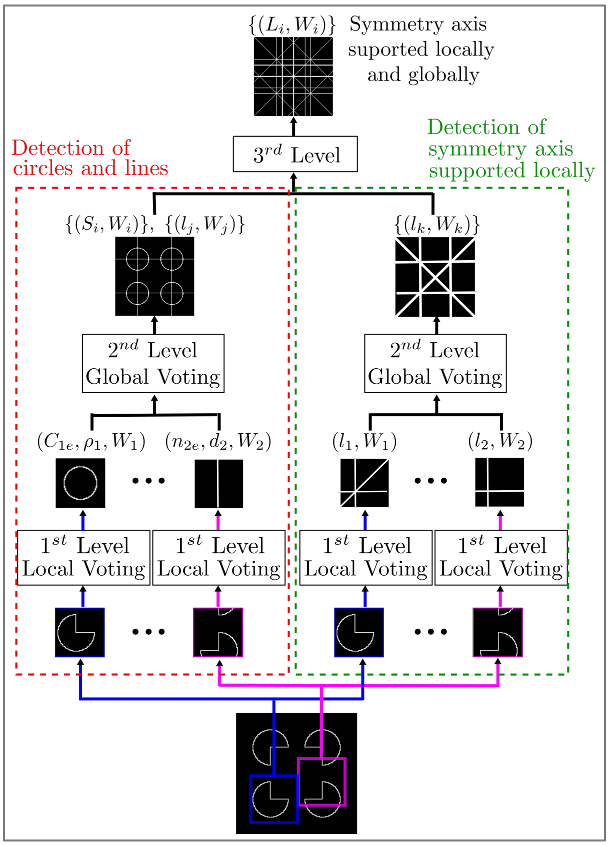

Fig. 5. Three-level architecture for extraction of symmetry axis.

symmetry in images. Figure 5 shows an overview of the architecture. The goal of the first two levels at the right side of the architecture is to extract symmetry axis with local support, for doing this, we apply a local and global voting procedure using Equations 14 and 19; the output is add to a set $\bar{L}$. On the other hand, the left side of the architecture extracts salient circles and lines. The third-level maps circles and lines to symmetry axis using Equations 15 to 19, and add them to the set $\bar{L}$; then we apply a global voting procedure. The result is a set of symmetry axis with local and global support.

For the experiments, we applied our voting scheme in images that contain real objects, as well as synthetic images; we used the benchmark proposed by [18], which contains a set of 30 test images, divided in 4 categories: synthetic vs. real images, and images with a single vs. multiple reflection axis. For each input image we apply a pre-processing step, which consist in a Canny edge detector [17] and a mean filter; after that, we use our algorithm for detection of bilateral symmetry. The output is shown in Figure 6, where the detected symmetry axis are superimposed to the input image.

To evaluate our algorithm, we calculate the precision and recall rates:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}; \tag{22}$$

where TP, FP and FN are the number of true positives, false positives, and false negatives, respectively. Figure 7 shows a comparison between precision and recall rates obtained by our voting scheme and the bilateral symmetry detection algorithms of Loy and Eklund [19], Mo and Draper [20], and Kondra *et al.* [21]. Our algorithm has a similar performance to these methods, and is slightly better than three of them.

VI. Conclusions and Future Work

This paper is introducing a generalization of voting schemes using the concept of inner products with respect to a flag. Considering the Equation (10), we see that we can design suitable flags for detection of complex structures involving transformation as well; in addition, we use functions to apply constraints on the flag, according to perceptual properties. Moreover, our voting scheme can be used to design a multi-level architecture, as is shown in the applications sections. Finally, the principles of Gestalt cannot be easily put into mathematics, this paper is a successful attempt to do so, and we have introduced the use of geometric algebra to model perceptual properties, like symmetry, as well as geometric structures; we hope this effort will lead to future developments in this direction.

Future works should focus on the design of new perceptual saliency functions and flags, the implementation of new stages of the architecture for high-level tasks, and the integration of other information channels like color, movement, disparity, etc.

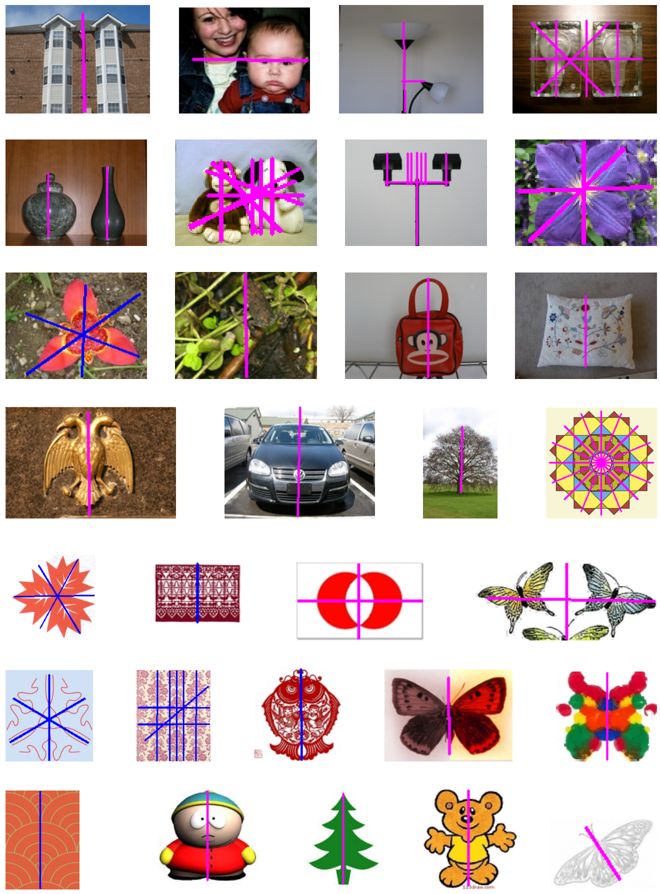

Fig. 6. Detection of bilateral symmetry using our voting method.

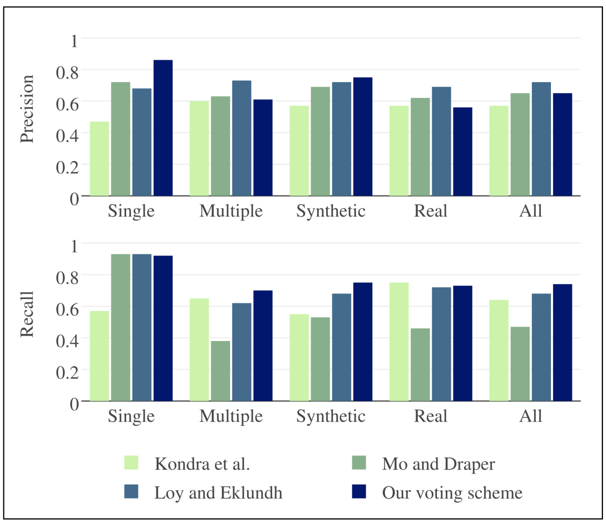

Fig. 7. Recall and precision rates for symmetry axis detection.

ACKNOWLEDGMENT

The authors are very thankful to CONACYT-Mexico for the Ph.D. scholarship with number 231213, and the project funding CB-2012-01 # 178222. They also are grateful with Gerardo Soria-García for providing the FPGA implementation of the voting scheme.

REFERENCES

[1] K. Boyer and S. Sarkar, "Perceptual organization in computer vision: Status, challenges, and potential," *Comput. Vis. Image Underst.*, vol. 76, no. 1, pp. 1–5, Oct. 1999.

[2] A. Sha'asua and S. Ullman, "Structural saliency: The detection of globally salient structures using a locally connected network," in *Proc. of the IEEE International Conference on Computer Vision (ICCV'88)*, Tampa, FL, USA, dec 1988, pp. 321–327.

[3] P. Hough, "Method and means for recongnizing complex patterns," U.S. Patent 3 069 654, dec 18, 1962.

[4] R. Duda and P. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[5] D. Ballard and C. Brown, *Computer Vision*. New Jersey, USA: Prentice Hall, 1982.

[6] G. Guy and G. Medioni, "Perceptual grouping using global saliency-enhancing operators." in *Proc. of the IAPR International Conference on Pattern Recognition. (ICPR'92)*, The Hague, Netherlands, Aug 1992, pp. 99–103.

[7] ——, "Inferring global perceptual contours from local features," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition. (CVPR'93)*, New York, USA, Jun 1993, pp. 786–787.

[8] P. Mordohai and G. Medioni, *Tensor Voting. A perceptual organization approach to computer vision and machine learning.*, ser. Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan & Claypool Publishers, 2006.

[9] L. Reyes, G. Medioni, and E. Bayro, "Registration of 3d points using geometric algebra and tensor voting," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 351–369, 2007.

[10] G. Altamirano-Gómez and E. Bayro-Corrochano, "Conformal geometric method for voting," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, E. Bayro-Corrochano and E. Hancock, Eds. Springer International Publishing, 2014, vol. 8827, pp. 802–809.

[11] H. Li, D. Hestenes, and A. Rockwood, "Generalized homogeneous coordinates for computational geometry," in *Geometric Computing with Clifford Algebras*, G. Sommer, Ed. Springer-Verlag, 2001, pp. 27–59.

[12] E. Bayro-Corrochano, *Geometric Computing: For Wavelet Transforms, Robot Vision, Learning, Control and Action*. Springer-Verlag, 2010.

[13] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, ser. Fundamental Theories of Physics. Dordrecht: Kluwer Academic Publishers, 1987.

[14] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'96)*. Portland, Oregon: AAAI Press, Aug 1996, pp. 226–231.

[15] L. Williams and K. Thornber, "A comparison of measures for detecting natural shapes in cluttered backgrounds," *International Journal of Computer Vision*, vol. 34, no. 2-3, pp. 81–96, 1999.

[16] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3485–3492.

[17] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[18] I. Rauschert, K. Brocklehurst, S. Kashyap, J. Liu, and Y. Liu, "First symmetry detection competition: Summary and results," The Pennsylvania State University, PA, Tech. Rep. CSE11-012, Oct. 2011.

[19] G. Loy and J. Eklundh, "Detecting symmetry and symmetric constellations of features," in *Proc. of the 9th European Conference on Computer Vision - Volume Part II*, ser. ECCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 508–521.

[20] Q. Mo and B. Draper, "Detecting bilateral symmetry with feature mirroring," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Colorado, USA, June 2011.

[21] S. Kondra, A. Petrosino, and S. Iodice, "Multi-scale kernel operators for reflection and rotation symmetry: Further achievements," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013, pp. 217–222.