

Machine Learning and Geometric Technique for SLAM

Miguel Bernal-Marin and Eduardo Bayro-Corrochano

Department of Electrical Engineering and Computer Sciences,
CINVESTAV Unidad Guadalajara, Av. Científica 1145, Col. El Bajío,
Zapopan, Jalisco 45015, Mexico
{mbernal,edb}@gd1.cinvestav.mx

Abstract. This paper describes a new approach for building 3D geometric maps using a laser rangefinder, a stereo camera system and a mathematical system the Conformal Geometric Algebra. The use of a known visual landmarks in the map helps to carry out a good localization of the robot. A machine learning technique is used for recognition of objects in the environment. These landmarks are found using the Viola and Jones algorithm and are represented with their position in the 3D virtual map.

1 Introduction

Mobile robots are equipped with multiple input devices to sense the surrounding environment. The laser rangefinder is widely used for this task due to its precision, and its wide capture range. In this paper we merged the data obtained by the laser and the stereo camera system to build a 3D virtual map with the shapes obtained by these devices. The 3D objects seen by the stereo camera system can be modeled by geometric entities, which are easy to represent and to combine. Some of these 3D objects can act as a landmarks for the robot navigation and relocalization. Line segments are used to build a 3D map and they are the most widely used features [1] [2].

Using the Conformal Geometric Algebra we can represent different geometric shapes including the line segments (as a pair of points) and the data captured by the stereo camera system (landmarks as labeled spheres). This framework also allows us to formulate transformations (rotation, translation) using spinors or versors.

2 Geometric Algebra

The Geometric algebra $\mathcal{G}_{p,q,r}$ is constructed over the vector space $\mathcal{V}^{p,q,r}$, where p,q,r denote the signature of the algebra; if $p \neq 0$ and $p = r = 0$, the metric is Euclidean; if only $r = 0$, the metric is pseudo Euclidean; if $p \neq 0$, $q \neq 0$, $r \neq 0$, the metric is degenerate. The dimension of $\mathcal{G}_{n=p+q+r}$ is 2^n , and \mathcal{G}_n is

constructed by the applications of the *geometric product* over the vector basis \mathbf{e}_i . The geometric product between two vectors \mathbf{a}, \mathbf{b} is defined as

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$$

and the two parts; the inner product $\mathbf{a} \cdot \mathbf{b}$ is symmetric part, while the wedge product (outer product) $\mathbf{a} \wedge \mathbf{b}$ is the antisymmetric part.

In $\mathcal{G}_{p,q,r}$ the geometric product of two basis is defined as

$$\mathbf{e}_i \mathbf{e}_j := \begin{cases} 1 \in \mathbb{R} & \text{for } i = j \in \{1, \dots, p\} \\ -1 \in \mathbb{R} & \text{for } i = j \in \{p + 1, \dots, p + q\} \\ 0 \in \mathbb{R} & \text{for } i = j \in \{p + q + 1, \dots, n\} \\ \mathbf{e}_{ij} = \mathbf{e}_i \wedge \mathbf{e}_j & \text{for } i \neq j. \end{cases}$$

this lead in a basis for \mathcal{G}_n that contains elements of different grade called *blades* (e.g. scalars, vectors, bivectors, trivectors, etc.): $1, \{\mathbf{e}_i\}, \{\mathbf{e}_i \wedge \mathbf{e}_j\}, \{\mathbf{e}_i \wedge \mathbf{e}_j \wedge \mathbf{e}_k\}, \dots, \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \dots \wedge \mathbf{e}_n$ which is called *basis blade*; where the elements of maximum grade is the pseudoscalar $\mathbf{I} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \dots \wedge \mathbf{e}_n$. A linear combination of basis blades, all of the same grade k , is called k -vector. The linear combination of such k -vectors is called *multivector*, and multivectors witch certain characteristics represent different geometric objects or entities (as points, lines, planes, circles, spheres, etc.), depending on the GA where we are working (for example, a point (a, b, c) is represented in $\mathcal{G}_{3,0,0}$ [the GA of the 3D-Euclidean space \mathcal{E}^3] as $\mathbf{x} = a\mathbf{e}_1 + b\mathbf{e}_2 + c\mathbf{e}_3$, however a circle can not be defined in $\mathcal{G}_{3,0,0}$, but it is possible to define it in $\mathcal{G}_{4,1,0}$ (CGA) as a 4-vector $\underline{z} = \underline{s}_1 \wedge \underline{s}_2$ [the intersection of two spheres in the same space]). Given a multivector M , if we are interested in extracting only the blades of a given grade, we write $\langle M \rangle_r$ where r is the grade of the blades we want to extract (obtaining an homogeneous multivector M^r or a r -vector).

The *dual* \mathbf{X}^* of a r -blade \mathbf{X} is defined by $\mathbf{X}^* = \mathbf{X}\mathbf{I}_n^{-1}$. It follow that the dual of a r -blade is an $(n - r)$ -blade.

The *reverse* of any multivector M is defined as

$$\langle \widetilde{M} \rangle_i = (-1)^{\frac{i(i-1)}{2}} \langle M \rangle_i, \text{ for } M \in \mathcal{G}_n, 0 \leq i \leq n. \tag{1}$$

The reader should consult [3] to detailed explanation about CGA and its applications.

2.1 Conformal Geometric Algebra

To work in Conformal Geometric Algebra (CGA) $\mathcal{G}_{4,1,0}$ means to embed the Euclidean space in a higher dimensional space with two extra basis vectors which have particular meaning; in this way we represent particular entities of the Euclidean space with subspaces of the conformal space. The extra vectors we add are \mathbf{e}_+ and \mathbf{e}_- , defined by the properties $\mathbf{e}_+^2 = 1, \mathbf{e}_-^2 = -1, \mathbf{e}_+ \cdot \mathbf{e}_- = 0$. With this two vectors, we define the null vectors $\mathbf{e}_0 = \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+)$ and $\mathbf{e} = \mathbf{e}_- + \mathbf{e}_+$ interpreted as the origin and the point at infinity, respectively. From now on

and in the rest of the paper, points in the 3D-Euclidean space are represented in lowercase, while conformal points in underline letters; also the conformal entities will be expressed in the *Outer Product Null Space* (OPNS) (noted with an asterisk beside, also know as the dual of the entity), and no in the *Inner Product Null Space* (IPNS) (without asterisk) unless it is specified explicitly. To go from OPNS to IPNS we need to multiply the entity by the pseudoscalar. To map a point $\mathbf{x} \in \mathcal{V}^3$ to the Conformal space in $\mathcal{G}_{4,1}$ (using IPNS) we use

$$\underline{x} = \mathbf{x} + \frac{1}{2}\mathbf{x}^2\mathbf{e} + \mathbf{e}_0 \tag{2}$$

Applying the wedge operator "∧" on points, we can express new entities in CGA. All geometric entities from CGA are show in the table 1 for a quick reference.

The pseudoscalar in CGA $\mathcal{G}_{4,1,0}$ is defined as $\mathbf{I} = \mathbf{I}_E\mathbf{E}$, where $\mathbf{I}_E = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ is the pseudoscalar from \mathcal{G}_3 and $\mathbf{E} = \mathbf{e}_+ \mathbf{e}_-$ is the pseudoscalar from the Minkowski plane.

In GA there exist specific operators to model rotations and translations called *rotors* and *translators* respectively. In CGA such operator are called *versor* and are defined by (3) being \mathbf{R} the rotor, \mathbf{T} the translator.

$$\mathbf{R} = e^{-\frac{1}{2}\underline{l}\theta}; \quad \mathbf{T} = e^{\frac{\mathbf{e}\mathbf{t}}{2}}, \tag{3}$$

where the *rotation axis* $\underline{l} = l_1\mathbf{e}_{23} + l_2\mathbf{e}_{31} + l_3\mathbf{e}_{12}$ is a unit bivector which represents a line (in IPNS) through the origin in CGA, θ is the rotation angle, $\mathbf{t} = t_1\mathbf{e}_1 + t_2\mathbf{e}_2 + t_3\mathbf{e}_3$ is the translation vector in \mathcal{V}^3 . The equations (3) can also be expressed as

$$\mathbf{R} = \cos\left(\frac{\theta}{2}\right) - \text{sen}\left(\frac{\theta}{2}\right)\underline{l}; \quad \mathbf{T} = \left(1 + \frac{\mathbf{e}\mathbf{t}}{2}\right) \tag{4}$$

Table 1. Entities in CGA

Entity	IPNS	OPNS
Sphere	$\underline{s} = \mathbf{p} + \frac{1}{2}(\mathbf{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0$	$\underline{s}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \underline{d}$
Point	$\underline{x} = \mathbf{x} + \frac{1}{2}\mathbf{x}^2\mathbf{e} + \mathbf{e}_0$	$\underline{x}^* = (-\mathbf{E}\mathbf{x} - \frac{1}{2}\mathbf{x}^2\mathbf{e} + \mathbf{e}_0)\mathbf{I}_E$
Plane	$\underline{P} = \mathbf{N}\mathbf{I}_E - d\mathbf{e}$ $\mathbf{N} = (\mathbf{a} - \mathbf{b}) \wedge (\mathbf{a} - \mathbf{c})$ $d = (\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c})\mathbf{I}_E$	$\underline{P}^* = \mathbf{e} \wedge \underline{a} \wedge \underline{b} \wedge \underline{c}$
Line	$\underline{L} = \underline{P}_1 \wedge \underline{P}_2$ $= \mathbf{r}\mathbf{I}_E + \mathbf{e}\mathbf{M}\mathbf{I}_E$ $\mathbf{r} = \mathbf{a} - \mathbf{b}$ $\mathbf{M} = \mathbf{a} \wedge \mathbf{b}$	$\underline{L}^* = \mathbf{e} \wedge \underline{a} \wedge \underline{b}$
Circle	$\underline{z} = \underline{s}_1 \wedge \underline{s}_2$ $\underline{s}_z = (\mathbf{e} \cdot \underline{z})\underline{z}$ $\rho_z = \frac{z^2}{(\mathbf{e} \wedge \underline{z})^2}$	$\underline{z}^* = \underline{a} \wedge \underline{b} \wedge \underline{c}$
P-pair	$\underline{PP} = \underline{s}_1 \wedge \underline{s}_2 \wedge \underline{s}_3$	$\underline{PP}^* = \underline{a} \wedge \underline{b}$

due to the exponential properties. Such operator are applied to any entity of any dimension by multiplying the entity by the operator from the left, and by the *reverse* of the operator from the right, as show in (5).

$$\underline{x}' = \sigma \underline{x} \tilde{\sigma} \quad (5)$$

where \underline{x} is any entities mentioned in table 1, and σ is a versor (*rotor*, *translator* or *motor* mentioned below). Using (5) is easily to transform any entities from CGA (points, point-pair, lines, circles, planes, spheres), not only points as is usual in other algebras.

In CGA it is possible to use the rotors and translator to express general rotation and screw motions in space. To model a screw motion, the entity has to be translated during a general rotation with respect to the rotation axis. The implementation consecutive of a translator and rotor can be written as the product of them. Such operator is called *motor* and expressed as

$$\mathbf{M} = \mathbf{TR} \quad (6)$$

The translator, rotor and motor (all of them *versors*) are elements from $\mathcal{G}_{4,1}^+$, and they defines an algebra called *motor algebra*. This algebra greatly simplifies the successive computation of rotations and translation, applying only the geometric product in consecutive versors, giving the final result another versor of this algebra, where all the transformations are together in one element.

Vector calculus is a coordinate dependent mathematical system and its cross product can not be extended to higher dimensions. The representation of geometric primitives is based in lengthy equations and for linear transformations one uses matrix representation with redundant coefficients. In contrast conformal geometric algebra a coordinate free system provides a fruitful description language to represent primitives and constraints in any dimension and by using successive reflections with bivectors one builds versors to carry out linear transformations avoiding redundant coefficients.

3 3D Map Building

Using an equipped mobile robot with a laser rangefinder sensor and stereo camera system mounted on a pan-tilt head, each one with their own coordinate system. We apply *the method of hand-eye calibration* [4] to get the center coordinates of each devices related to a global robot coordinate system. Using the perpendicular line to plane (x, y) as rotation axis and the angle of rotation of the robot, and adding a third fixed coordinate to the robot's movement in the plane we can apply this values in (3) to make \mathbf{T}_{pos} and \mathbf{R}_{pos} that represent the movement of the robot in the 3D environment.

Line segments from range points are extracted using recursive line splitting method as show in [1] [2], this is a speedy and correctness algorithm that performs *divide-and-conquer* algorithms [5]. For every endpoints of the line segments, we maps them to CGA to get the pair of points entity (see table 1) and

store in a map. To translate the position of the entities we use the motor (6) which is defined as the translation and rotation of the mobile robot.

To get the position of any entity extracted by the stereo camera system or laser rangefinder in the 3D environment, we apply a specific transformation using motors in CGA. Then place the entity in the 3D map.

4 Getting 3D Positions Based on Visual Landmarks

A landmark literally is a geographic feature used by explorers and others to find their way back or move through an area. In the map building process a mobile robot can use these landmarks to remember the place where it was before while it explore its environment. Also the landmarks can be used to find robot position in a previous building map facilitating the relocalization. As we are using a camera stereo system, the 3D position of any object can be also calculated and it can be represented in the 3D virtual environment. Using these objects as a landmarks, the robot gets its relative position.

4.1 Machine Learning Phase

A natural or artificial landmark located in the actual environment helps to the mobile robot to know its position on the map. Viola and Jones present a new and radically faster approach to face detection based on the AdaBoost algorithm from machine learning [6], and this approach can be used to detect our statics landmarks. Once the landmarks have been selected and trained, the mobile robot can use them to navigate in the environment performs the Viola an Jones algorithm. If a landmark is found we get a sub-image I_L from the left camera image. This I_L is the region of the image where the landmark was found (fig. 2).

When a landmark is identified in one image (left camera), we must be sure that the landmark is in the other image as well (right camera of the stereo camera system). To get the 3D position, the landmark must be detected in both images. The landmark in the right image is also detected by Viola and Jones algorithm, and identify its region by a sub-image I_R (fig. 1).

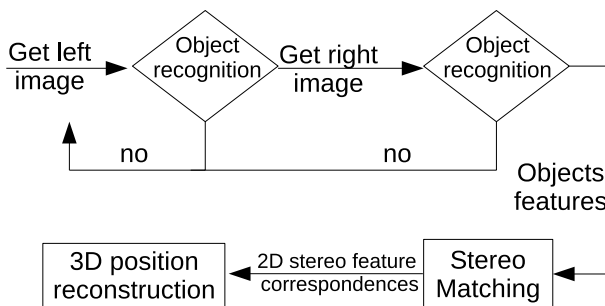


Fig. 1. The flowchart of the landmark position estimation

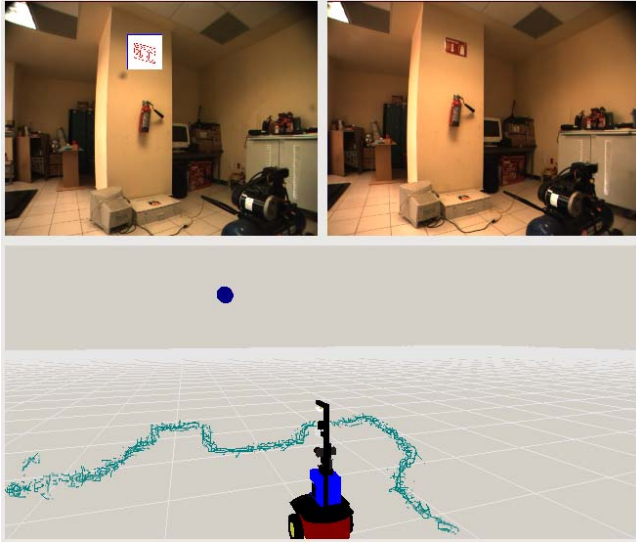


Fig. 2. Mobile robot founding landmarks while it is navigating its environment. On the top see stereo view (on left image a sign landmark found in white rectangle) and its representation in the 3D map.

4.2 Landmark Position Estimation

When we talk about the landmark position estimation, we are looking for the 3D location of these landmark in the environment and not for the pose (position and orientation) of the object found. Figure 1 illustrates the flowchart of the landmark position estimation.

Getting the landmark identified in both images, we proceed to calculate the points of interest. To do this we use Canny edge detection operator on I_L and a correlation. A number of correlation-based algorithms attempt to find points of interest on which to perform the correlation. In fact, the normalization embodied into the *Normalized Cross Correlation* (NCC) and *Zero Mean Normalized Cross Correlation* (ZNCC) allows for tolerating linear brightness variations. Furthermore, thanks to the subtraction of the local mean, the ZNCC provides better robustness than the NCC [7] since it tolerates uniform brightness variations as well.

Correspondences of an image patch are searched for along the epipolar line by calculating the ZNCC only in a given interval (d_{min}, \dots, d_{max}) of so-called disparities [8] [9]. The term *disparity* denotes the Euclidean distance from one point on the epipolar line to a given point in the other camera image [10]. A small disparity represents a large distance to the camera, a large value a small distance (parallax).

When all the points are matched in both images we proceed to calculate its 3D position using the triangulation. Then we integrate this set of points to get its center of gravity and place the center of a sphere on it. The radius of the sphere

is calculated taking the highest number of points of the landmark. The sphere is stored in the 3D virtual map using CGA and it is labeled as a landmark.

5 Experiments

The fig. 3 shows some signs as landmarks in a laboratory. A sign was trained with Viola-Jones algorithm. The result is: where a sign is placed in the wall, it is recognised by the robot, and place it at its 3D position in the environment (fig. 4). The first and last image of figure 3 represent the same landmark, but taken on different place of the laboratory.



Fig. 3. Signs as Landmarks in a laboratory

The figure 4 shows the map built with lines, the raw data of the laser (points), and the landmarks found.

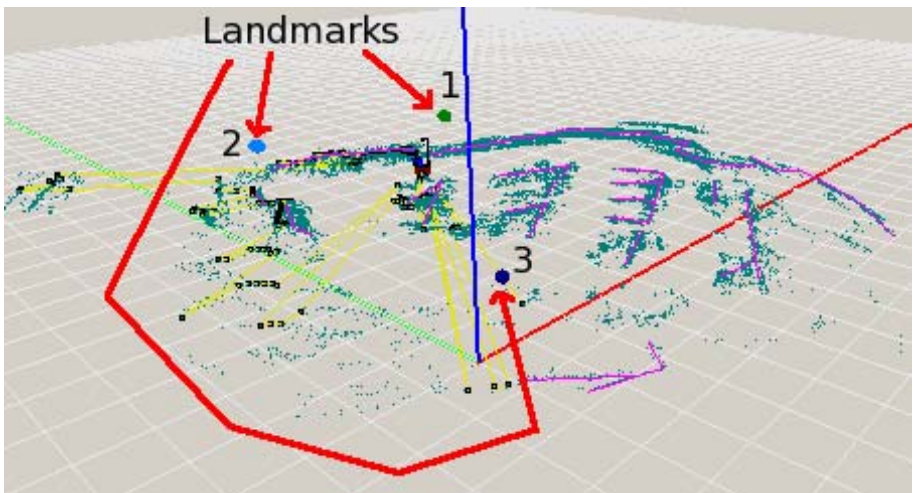


Fig. 4. 3D representation of the landmarks and the laser readings as lines and points cloud

6 Conclusions

In this paper the authors have shown the use of geometric entities in Conformal Geometric Algebra (CGA) for modeling input data for a 3D virtual environment, in this way merging in a global coordinate system, the laser rangefinder and stereo camera system (mounted over a pan-tilt unit). The machine learning technique is used for the object's recognition. The detected objects are used as a landmarks witch greatly help in the interaction with the environment. The experiments with a real robot validate our method. We believe that our approach can be of great use for mobile robots or upper body humanoids installed on mobile platforms.

References

1. Zhang, L., Ghosh, B.K.: Line segment based map building and localization using 2d laser rangefinder. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2538–2543 (2000)
2. Siadat, A., Kaske, A., Klausmann, S., Dufaut, M., Husson, R.: An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation. In: Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications, pp. 153–158 (1997)
3. Bayro-Corrochano, E.: Robot perception and action using conformal geometry. In: Bayro-Corrochano, E. (ed.) *The Handbook of Geometric Computing. Applications in Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, ch. 13, pp. 405–458. Springer, Heidelberg (2005)
4. Bayro-Corrochano, E., Daniilidis, K., Sommer, G.: Motor algebra for 3d kinematics: The case of the hand-eye calibration. *Journal of Mathematical Imaging and Vision* archive 13, 79–100 (2000)
5. Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. *Auton. Robots* 23(2), 97–111 (2007)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, December 2001, pp. 511–518 (2001)
7. Di Stefano, L., Mattoccia, S., Tombari, F.: ZNCC-based template matching using bounded partial correlation. *Pattern Recogn. Lett.* 26(14) (2005)
8. Faugeras, O., et al.: Real-time correlation-based stereo: algorithm, implementation and applications. INRIA Technical Report no. 2013 (1993)
9. Azad, P., Gockel, T., Dillmann, R.: *Computer Vision: Principles and Practice*. Ed. Elektor Electronics (2008)
10. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Ed. Cambridge University Press, Cambridge (2004)