INTELLIGENT AUTOMATION & SOFT COMPUTING, 2018
https://doi.org/10.1080/10798587.2017.1299356

Check for updates

Structure from Motion Using Bio-Inspired Intelligence Algorithm and Conformal Geometric Algebra

Nancy Arana-Daniel, Carlos Villaseñor, Carlos López-Franco, Alma Y. Alanís and Roberto Valencia-Murillo

Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Guadalajara, México

ABSTRACT

Structure from Motion algorithms offer good advantages, such as extract 3D information in monocular systems and structures estimation as shown in Hartley & Zisserman for numerous applications, for instance; augmented reality, autonomous navigation, motion capture, remote sensing and object recognition among others. Nevertheless, this algorithm suffers some weaknesses in precision. In the present work, we extent the proposal in Arana-Daniel, Villaseñor, López-Franco, & Alanís that presents a new strategy using bio-inspired intelligence algorithm and Conformal Geometric Algebra, based in the object mapping paradigm, to overcome the accuracy problem in two-view Structure form motion algorithms. For this instance, we include two new experiments and the inclusion of the circle entity; the circle carries stronger information about its motion than other geometric entities, as we will show.

KEYWORDS

Structure from motion; Bio-inspired algorithms; Conformal Geometric Algebra

1. Introduction

The computer vision field has been a very active research topic in last decades, but the problems to solve have proven to be difficult. The research of new techniques and the exploration of new paradigms bring benefits in the development of the field. In the present work, we propose a new strategy to overcome the accuracy problem in Structure from Motion (SfM) algorithms.

SfM algorithms (Hartley & Zisserman, 2003) are a family of computer vision algorithms whose paradigm is based on the relation between the motion and structures, and how this relation is implicit in the images. Using SfM, over two or more images we can calculate structures that represent objects, when we know the motion of the camera or the geometric relation between two or more cameras, as well as we can calculate the motion of known structure.

SfM is based in the epipolar restriction (1.1), where x_1 and x_2 represent a couple of images of a euclidian point X, with respect to two different cameras; E is called the *Essential matrix*, which relates corresponding points in stereo images, assuming that the cameras satisfy the pinhole camera model.

$$x_2^T E x_1 = 0 \tag{1.1}$$

However, in order to use this useful equation, we have to compute the *Essential matrix*. This problem is the main difference between SfM variants and has been achieved by a serial of algorithms like Random Sample Consensus (RANSAC) or the eight-points algorithm as is shown on (Hartley & Zisserman, 2003) and (Ma, Soatto, Kosetska, & Sastry, 2004), despite their velocity, they present a serious accuracy problem as we present in the next sections.

In the present time, the Bio-inspired Intelligence Algorithms (BIA) offers good solutions to complex problems, giving competitive times and small errors (Simon, 2013). The principal task in this kind of algorithm is to establish a suitable objective function, this is commonly performing in the linear algebra mathematical framework, but we can obtain good advantages using more general mathematical framework; like Conformal Geometric Algebra (CGA) as we will see in this work.

In this work, we propose a novel technique to overcome the accuracy problem in SfM algorithms with the use of BIA (Hernandez-Vargas, Lopez-Franco, Rangel, Arana-Daniel, & Alanis, 2014) and CGA. With this work, we extend the previous result shown in Arana-Daniel, Villaseñor, López-Franco, & Alanís (2014) to include a projection of the motion estimation error over the image, and in this way to be able to compare the proposal and the classical approach more intuitively and how this can affect the reconstruction of 3D structures. Moreover, we extended the initial proposal to include circles that are rich information geometric entities presented in many real objects. The circle detection is a very spread out research topic (Jiang, 2009; Semeikina & Yurin, 2011).

This paper is organized as follows: Section 2 presents a brief introduction to SfM algorithms, its fundaments, variants, benefits and problems. In Section 3, we present the basis of Geometric Algebra, its representations and operators. Section 4 presents the bio-inspired intelligent algorithm used in order to compute the *Essential matrix*. Section 5 presents the development of the proposed algorithm. Section 6 shows a comparison between the proposed algorithm and a classic approach with a couple of experiments, while Section 7 is devoted to conclusion and future work.

2. Structure from Motion

As we mentioned in the introduction, if the epipolar restriction holds, then the cameras satisfy the pinhole camera model

CONTACT Nancy Arana-Daniel nancyaranad@gmail.com

© 2018 TSI® Press

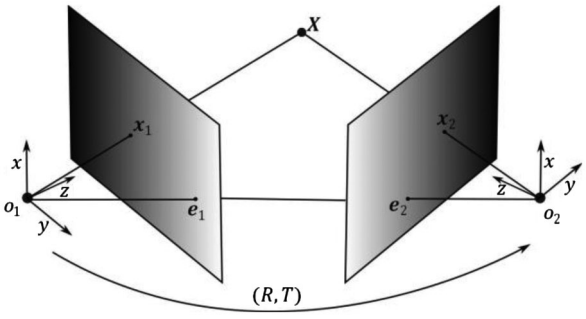


Figure 1. Epipolar Geometry.

(Ma et al., 2004). This model is given by (2.1), where K_f contains the focal length f, and Π_0 stands for the canonical projection matrix, g is the rigid transformation that contains R (rotation) and T (translation) of the camera with respect to the world coordinate, and finally X represents a homogeneous tridimensional point whose dimensional image is λx.

$$\begin{aligned}\lambda x &= K_f \Pi_0 g X \\ &= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &\quad \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}\end{aligned} \tag{2.1}$$

In the previous model, X is related to x for the so-called ideal perspective projection, but this is never the case due to the intrinsic parameters of the camera. In order to fix the model, we substitute K_f for $K = K_s K_f$, where K_s contains the intrinsic parameters of the camera. The matrix K is the so-called *Calibration matrix*.

Figure 1 shows a general scheme for epipolar geometry, where the euclidian point X is projected over two images. In this case, $x_1$and x_2 are related by the Equation (2.2). Then, as shown in Hartley & Zisserman (2003) and Ma et al. (2004) it is easy to reach (2.3), where $\hat{T}$ represents the screw symmetric matrix of the vector T. Notice that the *Essential matrix* $E = \hat{T}R$, contains the movement that relates two images of the same point.

$$\lambda_2 x_2 = R\lambda_1 x_1 + T \tag{2.2}$$

$$x_2^T \hat{T} R x_1 = x_2^T E x_1 = 0 \tag{2.3}$$

Although, in Equation (2.3), it is needed that the cameras satisfy the pinhole camera model (i.e. the cameras have to be calibrated), we can work with an uncalibrated version of the epipolar restriction, and this is shown in Equation (2.4), where c is the so-called *Fundamental matrix*.

$$x_2^T K^{-T} \hat{T} R K^{-1} x_1 = x_2^T F x_1 \tag{2.4}$$

Structure from Motion (SfM), as we already mentioned, are a family of algorithms whose goal is to find the *Fundamental matrix* or the *Essential matrix* from a set of correlated points (or structures). The way of computing these matrices are the main difference between the algorithms. In the first instance, consider two-frame SfM, where F can be understood as a bifocal tensor of a weak perspective. There are several implementations, for example, we can use the eight-points algorithm or RANSAC as is described in Hartley & Zisserman (2003) and (Ma et al., 2004), with an excellent runtime, but not very good accuracy as is shown in Arana-Daniel et al. (2014).

Other variants use a factorization method or constrictions based in lines and planes (the last one is called Constrained SfM) (Szeliski, 2010), where line-based techniques need a trifocal tensor in order to do 3D reconstruction. Plane-based techniques often perform poorly since the algebraic error does not correspond to the meaningful re-projection errors. In a second instance, we can also find Three-frame and n-frame SfM (Hartley & Zisserman, 2003), that use trifocal and n-focal tensors respectively, this improves the accuracy, but with a high computational cost. Finally other variants use iterative algorithms as Bundle Adjustment (Szeliski, 2010) or Extended Kalman Filter, (Civera, Davidson, & Martínez Montiel, 2011), or Sequential Monte Carlo methods (Qian & Chellappa, 2004), whose performances are in function to their computational cost.

In the present work, we propose a SfM algorithm based in the object mapping paradigm (Thrun, 2002), i.e. an algorithm of SfM that can work with maps where the objects are represented by geometric entities. In order to compare the proposal and its advantages, we present a classic SfM algorithm in Algorithm 1, which is used in Section 6 to compare accuracy and runtime with our proposal.

Algorithm 1: Classic Structure from Motion Algorithm
Require: Correspondences of images, Calibration matrix.
Ensure: Rotation and translation between images.
1: Use correspondences to calculate the *Fundamental matrix F* (using RANSAC).
2: Calculate the *Essential matrix* with $E = K^T F K$.
3: Discompose $E = \hat{T}R$ on its singular values $E = U\Sigma V^T$ (to extract the movement).
4: Use $W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ or W^Tdefined on (Hartley & Zisserman, 2003) to extract the movement with $\hat{T} = VW\Sigma V^T$ and $R = UW^{-1}V^T$.
5: Select the correct movement from the forth possible answers (Due to the use of W or W^T as is shown on (Hartley & Zisserman, 2003)).

3. Conformal Geometric Algebra

Geometric Algebra (GA) represents a family of associative algebras constructed over a quadratic space with a special product called Geometric Product. GA is a mathematical framework where we can find embedded concepts from linear, tensor, and quaternion algebras and other precious tools (for a deeper introduction we recommend Perwass (2009)). We use Geometric Algebra of type $\mathbb{G}_{p,q}$, which has an algebraic signature (p, q).

Conformal Geometric Algebra is a pseudoeuclidian GA that use a conformal transformation to span the euclidian space, this is denoted by $\mathbb{R}^n \to \mathbb{R}^{n+1,1}$. In order to apply it to $\mathbb{R}^3$ with basis $\{e_1, e_2, e_3\}$, consider the conformal projection over Minkowski plane with basis $\{e_+, e_-\}$ that possess the properties described on (3.1). We can also define a new basis called null basis with (3.2), and finally describe the representation of a tridimensional euclidian point x_e in CGA constructed over $\mathbb{R}^{4,1}$ denoted $\mathbb{G}_{4,1}$, in (3.3).

$$\begin{aligned} e_+^2 &= 1, \\ e_-^2 &= 1, \\ e_+ \cdot e_- &= 0 \end{aligned} \tag{3.1}$$

$$e_0 = \frac{1}{2}(e_- - e_+), \; e_\infty = e_- + e_+, \tag{3.2}$$

Table 1. Geometric Entities of CGA.

Entity	IPNS	OPNS
Point	$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_0$	$x_c^* = s_1 \wedge s_2 \wedge s_3 \wedge s_4$
Sphere	$s = x_e + \frac{1}{2}(x_e - r)^2 e_\infty + e_0$	$s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4$
Plane	$P = n + de_\infty$	$P^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty$
Line	$L = \boldsymbol{n}l_e - e_\infty \boldsymbol{m}l_e$	$L^* = x_1 \wedge x_2 \wedge e_\infty$
Circle	$C = s_1 \wedge s_2$	$C^* = x_1 \wedge x_2 \wedge x_3$

$$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_0 \tag{3.3}$$

The elements of GA are called multivectors, and for every instance we find two representations; the Inner Product Null Space (IPNS) representation and the Outer Product Null Space (OPNS) representation. The basic geometric entities of CGA are listed in Table 1, where r represents the radius of a sphere, n and d represent the parameters of the Hesse normal form, $\boldsymbol{n}$ and $\boldsymbol{m}$ are the Plücker coordinates of a line, s_1 to s_4 are spheres and x_1 to x_4 are conformal points.

One of the advantages of CGA is that we are able to use operators over all the geometric objects in CGA. In general, every transformation can be expressed like (3.4), where $X, X', H \in \mathbb{G}_{4,1}$ and σ is a scalar parameter to ensure the bijection between the null cone and $\mathbb{R}^3$.

$$\sigma X' = HX\tilde{H} \tag{3.4}$$

In rigid transformations, the parameter $\sigma = 1$. Rotations are represented by the so-called *rotors* defined in (3.5), where n represents the dual entity of the rotation axis and θ is the angle of rotation. Translations are represented with *translators* described in (3.6), where t is the euclidian translation. Finally we can describe how these operators are used in (3.7).

$$R = e^{-\frac{\theta}{2}n} \tag{3.5}$$

$$T = 1 + \frac{1}{2}te_\infty \tag{3.6}$$

$$X' = RX\tilde{R},\ X' = TX\tilde{T} \tag{3.7}$$

We can use both equations into a single operator called *motor*, we denote such operator in (3.8). This final equation represents the rigid movement and can be applied on every entity of the algebra.

$$M = RT,\ \ X' = MX\tilde{M} \tag{3.8}$$

4. Bio-Inspired Intelligence Algorithm

Bio-inspired intelligence algorithms have been a very important tool for complex problems resolution. Particle Swarm Optimization (PSO) (Eberhart & Shi, 2001) is an evolutionary algorithm based in the observation of bird flocks and insect swarm behaviours that provides high convergence speed. The PSO algorithm consists of an interactive adaptation of a set of multidimensional vectors called particles. These particles communicate information with each other to find a better candidate solution for an objective function.

Nowadays, we can find many variants of the PSO algorithm that intent to alleviate the premature convergence problem in the original PSO. In this work, this new proposed algorithm called Bio-inspired Aging Model Particle Swarm Optimization (BAM-PSO) (Hernandez-Vargas et al., 2014) has been proven to solve the mentioned problem and improve the accuracy results obtained with other PSO variants.

The BAM-PSO algorithm is an evolutionary computation algorithm based in the Aging Leader and Challengers (ALC-PSO) algorithm (Chen et al., 2013). The basic idea in both algorithms is to include an aging process, in the case of BAM-PSO this is extended to each particle in the swarm. ALC-PSO uses a linear aging model. BAM-PSO proposes to use a bio-inspired aging model.

Each particle in the swarm has a velocity v_{ij} and a position x_{ij}, and they are defined respectively, in Equations (4.1) and (4.2), where i is the ith-particle of the swarm, j is the jth-element of dimension problem, t is the iteration counter, w represents the inertial weight factor (constant), and R_2 are random, normalized and uniformly distributed values, $c_1$and c_2 represent the social and cognitive parameter respectively, $x_{ij}(t)$ is the particle ij-position for the iteration t; $x_{ij}(t+1)$ is the particle ij-position for $t+1$ iteration, $v_{ij}(t)$ is the particle ij-velocity for t iteration; $p_{ij}(t)$ represents the local best position for ij-particle in iteration t, and L is the leader that holds the best solution found by the swarm at iteration t

$$v_{ij}(t+1) = wv_{ij}(t) + c_1R_1\left(p_{ij}(t) - x_{ij}(t)\right) + c_2R_2\left(L_j - x_{ij}(t)\right) \tag{4.1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{4.2}$$

Finally, BAM-PSO includes a measure of premature convergence k_j in jth-dimension defined in (4.3), where D is the number of the dimension of the problem, $k_{\min}$ is the minimum deviation allowed, and $\bar{p}_j$ is the mean of all particles in jth-dimension.

$$k_j = \frac{k_{\min}}{\sqrt{\frac{\sum_{j=1}^{D}(p_{ij} - \bar{p}_j)}{D-1}}} \tag{4.3}$$

5. Structure from Motion Usign BAM-PSO and CGA

The SfM algorithms based in a bifocal tensor present low accuracy and they are strongly dependent of the exact image correlation. The object mapping approach is the idea of representing objects with a set of geometric entities, and allows compacts and rich information maps. CGA is a suitable framework to represent geometric entities, and recently, mapping algorithms have been developed in this mathematical framework (López-Gonzáles, Arana-Daniel, & Bayro-Corrochano, 2013).

Using the CGA mathematical framework combined with a modern technique of evolutionary computation (BAM-PSO), it is possible to find a suitable SfM algorithm that works with object maps. In order to achieve that, we have to develop the objective function that allows us to find the rigid transformation of an object, represented with various entities of CGA.

To establish a search space for BAM-PSO, we need to define first the parameters in the rigid transformation represented with the M versor described on (5.1), where T stands for the *translator*, and R_x, R_y and R_z are *rotors* that represent the rotation in the x, y and z axis respectively.

$$M = TR_xR_yR_z \tag{5.1}$$

In order to operate M, let us define θ_x, θ_y and θ_z as the rotation angles of R_x, R_y and R_z respectively; and t_x, t_y and t_z as the euclidian translation on T. Therefore, the search space is defined in (5.2).

$$\chi = \left\{\theta_x, \theta_y, \theta_z, t_x, t_y, t_z\right\} \tag{5.2}$$

Table 2. Parameters Extraction in CGA.

Entity	$p(\cdot):\Omega \in \mathbb{G}_{4,1} \rightarrow \mathbb{R}^h$
Point	$x_c \rightarrow \{x_e\}$
Sphere	$s \rightarrow \{x_e, r\}$
Plane	$P \rightarrow \{n, d\}$
Line	$L \rightarrow \{\boldsymbol{n}, \boldsymbol{m}\}$
Circle	$C \rightarrow \{c, r, n, d\}$

The search space will be bounded for the first three parameters by $[\pi, -\pi]$. Every element and operator of CGA can be represented in a computer with an array of 32 spaces; to get a better management of the computing resources, we can parameterize each geometric entity $\boldsymbol{\Omega}$. In this case, we can find another virtue of CGA, in the IPNS representation we can easily extract the parameter of almost all the geometric entities as shown in Table 2.

Table 2 shows that in the IPNS representation, the extraction of parameters is easily performed. The case of the point and sphere, is almost trivial extracting a euclidian point and a radius in the case of the sphere. The plane is extracted using the Hesse normal form with four parameters, and for the line is used a normalized Plücker coordinates with six parameters.

In order to extract the parameters of a 3D circle, let us define the circle with a euclidian point c that represents the circle centre, r is a radius and for the inclination we can use the parameters of the plane that contain the centre of the circle. In this way, we can obtain $C \rightarrow \{c, r, n, d\}$ with eight parameters. To get these parameters we use (5.3) published in Hitzer (2005), constructed with three points contained in the circle, these points are solutions to the Equation (5.4), for the plane we expand the circle with $C^* \wedge e_\infty$. An alternative to use these parameters, we can represent the circle like a multi-entity object with three points, this is not a unique representation of the circle, but for the goal to obtain the rigid transformation, it works faster.

$$c = -C(e_\infty \wedge e_0)I_c - \frac{1}{2}\left(\frac{C^2}{I_c^2}\right)e_\infty I_c = (p_1 - p_2) \wedge (p_2 - p_3) \tag{5.3}$$

$$C \Lambda X = 0 \tag{5.4}$$

The circle entity carries strong information of its motion, i.e. the rigid transformation between two circles (local rotation and translation) has just two possible solutions. To understand this, consider first the case of the point. Between two points there is only one possible translation, but the point itself is ambiguous to rotations, then, for a general rigid transformation between two points, we have infinite solutions. This does not mean that the points are useless for motion estimation, but we need more of them to have a better understanding of the motion. We mean with more points we have a stronger assumption.

The case of the line is similar to the one of the point, for example, between two lines we can always calculate a rotation (with just two solutions), but if we include translations in a general rigid motion, we get infinite possible rigid transformations between them, i.e. the line is ambiguous to translations. The case of spheres is the same of the point, ambiguous with rotations, and the plane is ambiguous to translations.

As we can see, points, spheres, lines and planes have ambiguities for general rigid transformations and if we try to calculate it, we will get infinite solutions. For the BIA, it means that we get infinite equal local minima. This issue is a strong restriction for the proposal, we cannot parallelize it to the entity level, another way to say it is that we cannot optimize the movement of an object with any of these entities independently.

Then, for an object represented with multiple entities, we can argue that with more entities, we get better accuracy. Using points and spheres we are giving strong information about translation and with lines and planes, information about rotation.

Consider again the circle, between two circles, we can easily find the translation between their centres, and for the rotation we have exactly the same solution of the planes that contain those circles. Then, we have two solutions for a general rigid transformation. This part is important for the proposal because we have an entity that contributes with strong information about translation and rotation.

Finally, using the Root of the Mean Squared Error (RMSE), we can obtain a suitable objective function described on (5.5).

$$f_o = \min_{\chi} \sum_{k=1}^{q} \sqrt{\frac{\sum_{l=1}^{h}\left[p_l\left(\Omega_t^k\right) - p_l\left(M\Omega_{t-1}^k\tilde{M}\right)\right]^2}{h}} \tag{5.5}$$

where Ω_t^k represents the kth-entity of an object in the time t. Then the objective function optimizes M constructed in χ search space, with the parametric difference between the actual entity and an estimated one, for an object represented for q entities, each one with a different number of h parameters.

6. Experiments and Results

In order to proof the proposal and to extend the previous results including circles between the geometric entities used, we present two representative experiments that show the quality of the applications that can be developed. In the first instance, we summarize some important particularities of the implementation. We are going to compare the proposed SfM algorithm with a classic two-frame SfM algorithm, both of them need previous data that depends of other algorithms; in the first place, the proposal needs a previous mapping, in the second, Classic SfM needs an exact correspondence algorithm. For a fair comparison, these two processes were performed in a supervised way.

A previous calibration of the camera has been done with Zhang method (Zhang, 2000). The map for the proposed method was constructed using a stereo camera system Bumblebee 2 BB2–08S2 from Point Grey through Triclops Stereo SDK. Both algorithms were programmed in MATLAB platform although it is not the fastest language; we assume the time of comparison is useful. The first experiment was done in an Intel i7-2,600 processor with 8 GB of RAM and the second one in an Intel i7-4,770 processor with 16 GB of RAM. Finally, let us define an error to compare both algorithms in (6.1), where x_t represents an euclidian point that belongs to a rigid object in the time t and H is a homogeneous transformation constructed with the found movement on each algorithm.

$$E_H = ||x_t - Hx_{t-1}|| \tag{6.1}$$

6.1. First experiment

Consider the images in Figure 2, we have three different scenes (a), (b) and (c) with a human body, let's define five objects; 1: Head and trunk, 2: Right arm, 3: Left arm, 4: Right leg and 5: Left leg. In the (d) image we present the map of entities with all

Table 3. Results of the First experiment.

	Between (a) and (b)				Between (b) and (c)			
	Proposed SfM		Classic SfM		Proposed SfM		Classic SfM	
Object	Time (s)	E_H (m)	Time (s)	E_H (m)	Time (s)	E_H (m)	Time (s)	E_H (m)
Head and truck	2.945	**0.0031**	**2.1512**	1.5112	4.1200	**0.0015**	**1.987**	1.561
Right arm	2.4531	**0.124**	**2.2113**	2.0021	2.513	**0.2415**	**1.912**	4.5124
Left arm	4.3458	**0.021**	**2.008**	3.1240	4.125	**0.1234**	**2.145**	3.1564
Right leg	4.6521	**0.124**	**1.984**	4.5121	2.453	**0.6124**	**2.004**	2.1356
Left leg	5.8412	**0.013**	**2.1542**	3.2135	3.412	**0.0130**	**1.815**	4.1254

Note: Bold values show the best results of time and error between the classic SfM algorithm and the proposed SfM algorithm.

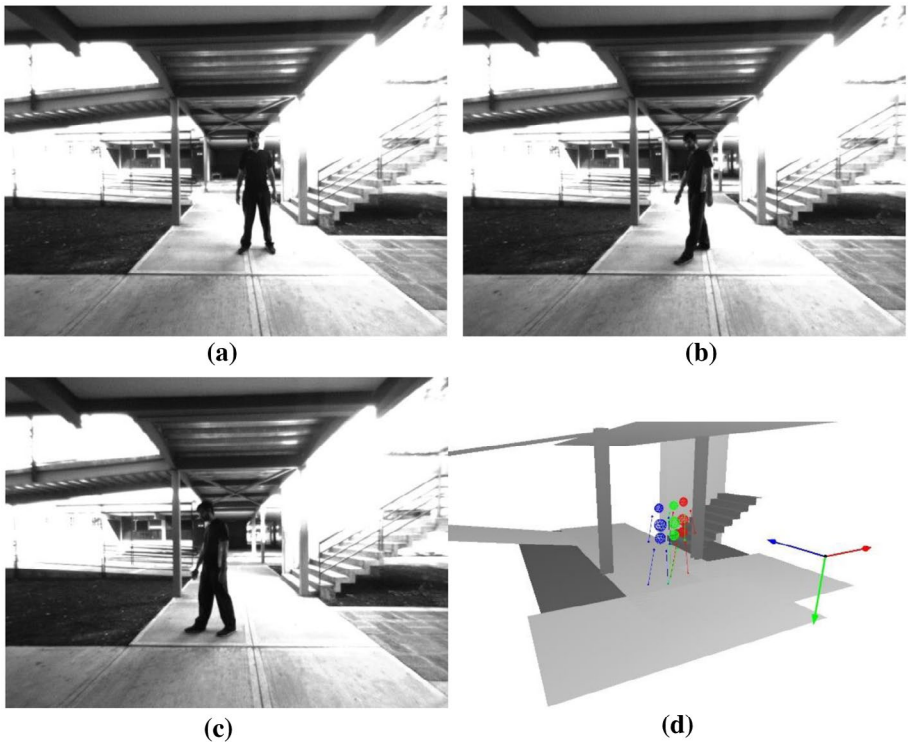


Figure 2. Input Images and Map for the First Experiment. Images (a), (b), and (c) are the Input Images for the Algorithms and (d) is the Map with Geometric Entities, which Represent the Objects on all the Input Images.

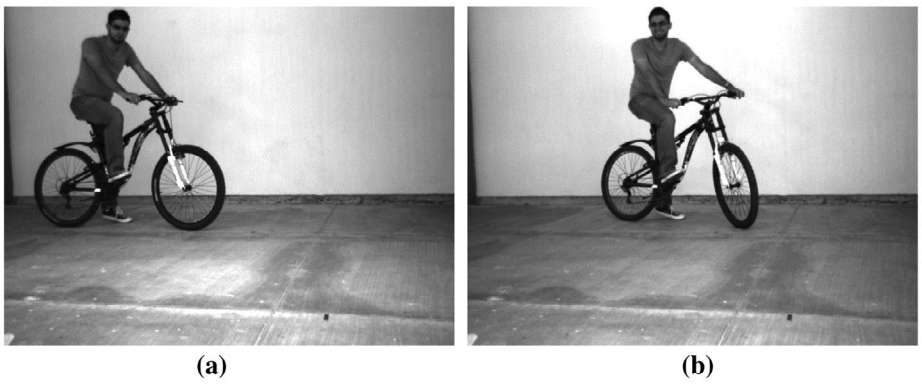


Figure 3. Input Images for the Second Experiment. Note the Presence of Circle Shapes in Several Objects.

the scenes included, where object 1 is represented with three spheres, objects 2 to 5 are represented with two points and one line. It is important to notice how geometric entities represent structures and offer more information than cloud points, but the map is still compact in memory. In Table 3, we show the results.

Note that for all the results on Table 3, the proposed SfM has a significant better accuracy, but the classic SfM wins always in runtime.

6.2. Second experiment

In this second experiment consider the images in Figure 3 as the input images to our algorithms. Let us define the objects like; 1: Head and trunk; 2: Right arm, 3: Left arm, 4: Thigh, 5: Shin, 6: Front wheel and 7: Back wheel. Figure 4 represents the map for the proposed algorithm, where object 1 is represented with three spheres, objects 2 to 5 are represented with two points and one line, and objects 6 and 7 are represented

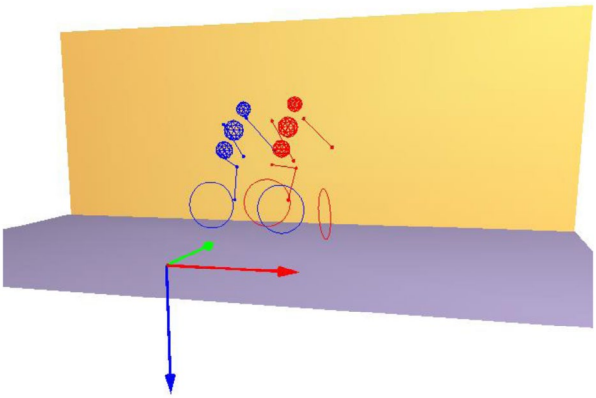


Figure 4. Object Map for the Second Experiment.

Table 4. Results of Second Experiment.

	Propose SfM		Classic SfM	
Object	Time (s)	E_H (m)	Time (s)	E_H (m)
Head and Truck	2.1765	**0.2450**	**1.5857**	1.2879
Right arm	3.6910	**0.1544**	**1.5333**	1.5482
Left arm	2.7572	**0.0565**	**1.6499**	2.1046
Thigh	2.8906	**0.0045**	**1.5291**	3.2145
Shin	3.1729	**0.1287**	**1.2430**	2.846
Front wheel	**0.9612**	**0.0476**	1.3456	0.5946
Back wheel	**0.5579**	**0.1151**	1.4691	0.7642

Note: Bold values show the best results of time and error between the classic SfM algorithm and the proposed SfM algorithm.

with one circle, in order to shown the strong assumption of using circles.

Table 4 shows the results of the second experiment. Finally, in Figure 5 we project the error in the second image, to present some intuition in the comparison.

In the same way like the first experiment, the proposed algorithm presents results with better accuracy, and in this case, two of the objects have a better runtime. Actually, these two objects contain a circle, with this we can argue experimentally that the circle has strong information of its motion. Moreover, the classic SfM in six of the cases has a better runtime.

In Figure 5, we have taken randomly one point in every object with the mark "*", with this we can calculate the result between two images and we re-project the resultant point with "o". Then the line in both images are the re-projection of the movement estimation error, where is easy to see how our proposal presents a smaller error in comparison than the classic approach.

7. Conclusion and Future Work

As we have shown in the experimentation, the proposed algorithm presents a novel method to use SfM with the object mapping paradigm. The use of bio-inspired intelligent algorithms over the suitable mathematical framework like CGA, allow us to accomplish an algorithm that overcomes the accuracy problem in Two-frame SfM algorithms. The proposed algorithm exhibits excellent results in the error, but it is slower than the classic one.

The principal improvement in comparison with Arana-Daniel et al. (2014), is the inclusion of the circle entity, as we already said, the circle carries strong information about its motion, because it has information about the change in orientation and translation, this property leads to a function with fewer local minima to optimize. Thanks to this property, we can obtain the result so accurate and fast for the front and back wheel in Table 4. Including this entity in a multi-entities object will improve the performance of the algorithm.

As future work, the present paper can be extended to other Geometric Algebras using the same objective function with new parametrization. We can also parallelize the BAM-PSO algorithm to get better times. In the case of the Classic SfM algorithm, we can also parallelize the singular value decomposition, as is shown in (Luk, 1985), and compare again the algorithms.

Acknowledgements

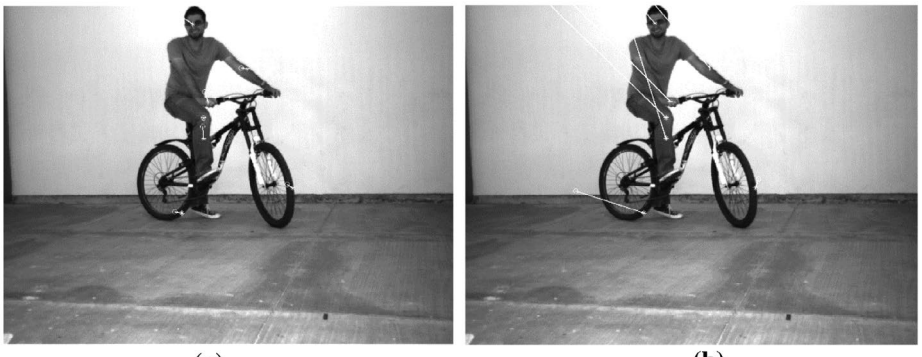
The authors thank the support of CONACYT México, through Projects CB256769 and CB258068 (Project supported by Fondo Sectorial de Investigación para la Educación).

Disclosure statement

No potential conflict of interest was reported by the authors.

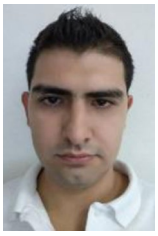
Notes on contributors

Nancy Arana-Daniel received in 2003 an M. S. degree in Computer Science and a Ph.D. in Computer Science in 2007, both from Center of Research and Advanced Studies, CINVESTAV, Unidad Guadalajara, México. She is currently a research fellow at the University of Guadalajara, in the Department of Computer Science México, where she is working in the Laboratory of Intelligent Systems and the Research Center on Control Systems and Artificial Intelligence. Her research interests focus in applications of geometric algebra, machine learning, optimization, pattern recognition and robot navigation.



(a) (b)

Figure 5. Re-projection of the Movement Estimation Error. (a) Proposed SfM, (b) Classic SfM. The "*" Mark Represents the Correct Point and "o" Mark Represents the Estimated Point with the Respect Algorithm. Note how the Proposal Represents Smaller Errors in Comparison to the Classic Approach.



Carlos Villaseñor is a Ph.D. student in the Computer Sciences Department, in the Centro Universitario de Ciencias Exactas e Ingeniería at the Universidad de Guadalajara in México. He received a Master's degree in Computer Science in 2014 and a B.Sc. degree in Industrial Engineer at the same university. His research focuses in bio-inspired algorithms and applications of geometric algebras.

Carlos López-Franco received a Ph.D. degree in Computer Science in 2007 from the Center of Research and Advanced Studies, CINVESTAV, Mexico. He is currently a professor at the University of Guadalajara, México, Computer Science Department, and member of the Intelligent Systems group. His research interests include geometric algebra, computer vision, robotics and intelligent systems.

Alma Y. Alanís, received a Ph.D. degree in Electrical Engineering from the CINVESTAV-IPN, Guadalajara Campus, Mexico, in 2007. Since 2008 she has been with University of Guadalajara, where she is currently a chair professor in the Department of Computer Science. She is also member of the Mexican National Research System (SNI-2) and member of the Mexican Academy of Sciences. She has published papers in recognized international journals and conferences, besides four international books. Her research interest centers on neural control, backstepping control, block control, and their applications to electrical machines, power systems and robotics.

Roberto Valencia-Murillo is a Ph.D. student in Computer Science Department, at the Centro Universitario de Ciencias Exactas e Ingeniería at Universidad de Guadalajara in México. He has received a Master's degree in Computer Science in 2013 and a B.Sc. degree in Computer Science at the University of Baja California. His research interests focus in machine learning, robot navigation and geometric methods for robotics

References

Arana-Daniel N., Villaseñor C., López-Franco C., & Alanís A. (2014). Bio-inspired aging model-particle swarm optimization and geometric algebra for structure from motion. In Eduardo Bayro-Corrochano and Edwin R. Hancock, *Progress in pattern recognition, image analysis, computer vision, and applications* (pp. 762–769). Puerto Vallarta, Mexico: Springer.

Chen, W.N., Zhang, J., Lin, Y., Chen, N., Zhan, Z.H., Chung, H.S., & Shi, Y.H. (2013). Particle swarm optimization with an aging leader and challengers. *Evolutionary Computation, IEEE Transactions on, 17*, 241–258.

Civera, J., Davidson, A.J., & Martínez Montiel, J. (2011). *Structure from motion using the extended Kalman filter*. Berlin/Heidelberg: Springer Science & Business Media.

Eberhart, R.C., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Evolutionary computation, 2001. proceedings of the 2001 congress on* (pp. 81–86). Seoul, Korea: IEEE.

Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge, UK: Cambridge University Press.

Hernandez-Vargas, E., Lopez-Franco, C., Rangel, E., Arana-Daniel, N., & Alanis, A. (2014). Particle swarm optimization with bio-inspiring aging model. Submitted to IEEE Trans Evolutionary Computation, 2014.

Hitzer, E.M. (2005). Euclidean geometric objects in the Clifford Geometric Algebra. *Bulletin of the Belgian Mathematical Society-Simon Stevin, 11*, 653–662.

Jiang, L.Y. (2009). Fast detection of multi-circle with randomized Hough transform. *Optoelectronics letters*, 5(5), 397–400.

López-Gonzáles, G., Arana-Daniel, N., & Bayro-Corrochano, E. (2013). Conformal hough transform for 2D and 3D cloud points. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 73–83.

Luk, F.T. (1985). A parallel method for computing the generalized singular value decomposition. *Journal of Parallel and Distributed Computing*, 2(3), 250–260.

Ma, Y., Soatto, S., Kosetska, J., & Sastry, S. (2004). *An invitation to 3-d vision: From images to geometric models*. Berlin/Heidelberg: Springer Science & Business Media.

Perwass, C. (2009). *Geometric algebra with applications in engineering*. Berlin Heidelberg: Springer-Verlag.

Qian, G., & Chellappa, R. (2004). Structure from motion using sequential monte carlo methods. *International Journal of Computer Vision*, 59(1), 5–31.

Semeikina, E., & Yurin, D. (2011). Circle detection based on multiscale edge curvature estimation. *Pattern Recognition and Image Analysis*, 21(2), 318–320.

Simon, D. (2013). *Evolutionary optimization algorithms*. New Jersey, NJ: John Wiley & sons.

Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Berlin/Heidelberg: Springer Science & Business Media.

Thrun, S.A. (2002). Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1–35.

Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11), 1330–1334.