Intelligent Automation And Soft Computing, 2018
Copyright © 2018, TSI® Press
Vol.24,no.2,431-442

Robot Pose Estimation Based on Visual Information and Particle Swarm Optimization

Carlos Lopez-Franco[1], Javier Gomez-Avila[2], Nancy Arana-Daniel[3], Alma Y. Alanis

Computer Science Department, CUCEI, University of Guadalajara. Blvd. Marcelino Garcia Barragan 1421, Col. Olimpica, Guadalajara, Jalisco, Mexico, C. P. 44430
Email: [1]carlos.lopez@cucei.udg.mx, [2]jegomezavila@gmail.com, [3]nancy.arana@cucei.udg.mx

ABSTRACT
This paper presents a method for 3D pose estimation using visual information and a soft-computing algorithm. The algorithm uses quaternions to represent rotations, and Particle Swarm Optimization to estimate such quaternion. The rotation estimation problem is cast as a minimization problem, which finds the best quaternion for the given data using the PSO algorithm. With this technique, the algorithm always returns a valid quaternion, and therefore a valid rotation. During the estimation process, the algorithm is able to detect and reject outliers. The simulations and experimental results show the robustness of algorithm against noise and outliers.

KEY WORDS: 3D pose estimation, particle swarm optimization.

1 INTRODUCTION

POSE estimation is an essential problem in robotics applications, it can be defined as an optimization problem. Given two consecutive data measurements acquired in different robot poses, the objective is to estimate the rigid transformation between them. An accurate pose estimation is fundamental in the development of an autonomous robot. This work focuses on mobile robots that require a precise position estimation in a GPS denial area. There are many environments where the GPS is not available in most indoor environments, or environments where the GPS signal presents errors due to signal arrival time measurements, atmospheric effects, etc.

The pose estimation of a mobile robot with wheels can be performed with the use of encoders, which provide a measurement of wheels rotations. However, these estimations suffer from systematic errors due to kinematics imperfections of the mobile robot, errors due to wheel slip, or to the environment itself. These errors will accumulate and the estimation of the pose will not be accurate. In addition, the estimation of the odometry from encoders is totally useless when dealing with not wheeled robots, like bipedal robots or flying unmanned vehicles. To avoid these problems, the pose estimation based on visual information is proposed, in this way, the relative pose is estimated based on what the system is watching at every step and thus no measurement errors will accumulate (Junmin, Jinge, Wentian, Shiwei, and Zhen, 2013).

Vision sensors are sensitive to changes in the light intensity, it is common to deal with outliers which are strange data values that do not belong to the datasets. These outliers can be detected and rejected with an optimization algorithm such as PSO (Bakar, Hamdan and Nazri, 2010). The PSO algorithm was developed by (Kennedy and Eberhart, 1995) and is inspired by social behavior of bird flocking. The PSO is an algorithm that is often used in vision tasks such as image classification (Jain, Kasturi, and Schunck, 1995), trajectory planning (Jin and Wu, 2013), electric power (Alanis, Rangel, Rivera and Lopez-Franco, 2013), image processing (López-Franco, Villavicencio, Arana-Daniel, and Alanis, 2014), among others optimization problems (Tewolde, 2013), (Eiben and Smith, 2008), (Goldberg and Holland, 1988).

CONTACT Alma Y. Alanis almayalanis@gmail.com

1.1 Related work

In (Welch and Foxlin, 2002), the authors use encoders to obtain motion information of a system. However, this kind of sensors is not useful in outdoors environments where the floor is not uniform and the contact of the wheel with the surface is not guaranteed. In (Welch and Foxlin, 2002) and (Choi, Suh, and Park, 2006) the authors use inertial sensors such as gyroscopes and accelerometers, and its measurements are fused to estimate the rotation of the system. Furthermore, they cannot compute its position and they need a specific pattern.

In (Yang, Dong, Wang, and Zhang, 2002) and (Yang, Yu, Wang, and Zhang, 2004) the authors propose the use of a Light Detection and Ranging (LIDAR) sensor. However these approaches present disadvantages when they are compared with computer vision approaches, especially when the robot is limited in weight load and power consumption, like in (Tong, Liu, and Li, 2012) where a monocular model-based approach is used for pose tracking or (Kaempchen, Franke, and Ott, 2002) and (Junmin, Jinge, Wentian, Shiwei, and Zhen, 2013) where stereo vision approach is proposed. These approaches work in an indoor environment only, and they do not present a method to handle outliers, i.e. when the matching algorithm has incorrect matches.

In our case, we use a stereo vision sensor due to the large amount of information that these sensors can provide, in addition to its light weight and low power consumption. Visual features are detected using Speeded-Up Robust Features (SURF) (Steder, Grisetti, Grzonka, Stachniss, Rottmann, and Burgard, 2007). It is important to note that visual sensors are noisy and susceptible to changes in illumination, this can cause mismatching errors and therefore errors in the pose estimation.

1.2 Main contribution

IN this work, we propose an algorithm to solve the pose estimation problem based on visual data. The proposed approach uses quaternions to represent rotation due to their advantages over rotation matrices (Salamin, 1979).

To estimate the quaternion we propose to use a soft-computing algorithm, in particular, the Particle Swarm Optimization (PSO) algorithm. The advantages of our approach are that the solution is always a quaternion, and therefore a valid rotation. In contrast, the SVD approach (Arun et al., 1987) can obtain solutions that minimize the objective function, but the solution matrix may not be a valid rotation matrix, i.e. det(R)=1.

In addition, the proposed approach is able to detect and remove outliers. We propose the use of support features for each possible quaternion and select the solution with more supporting features.

This paper is organized as follows: In section 2, the pose estimation problem is introduced. Then, in section 3, the PSO paradigm is briefly discussed. Later, in section 4 the proposed approach for pose estimation is presented. The simulations and experimental results are presented in section 5 and 6. Finally, the conclusions are given in section 7.

2 POSE ESTIMATION PROBLEM

THE pose estimation problem consists in determining the rotation and translation of an agent. To estimate the pose of the agent we need a sensor attached to it, for example, a range sensor. In this work, we use an image sensor, which can return 3D information like a stereo vision sensor or an RGB-D sensor.

The pose estimation problem can be defined as the estimation of the translation and rotation of an agent moving through the environment by using images taken at discrete time instants, see Fig. 1. Given two camera poses at adjacent time instants *i-1* and *i* are related by a rigid body transformation matrix of the form

$$ {}^{i-1}\mathrm{T}_i = \begin{bmatrix} {}^{i-1}\mathrm{R}_i & {}^{i-1}\mathrm{t}_i \\ 0 & 1 \end{bmatrix} \tag{1} $$

where ${}^{i-1}R_i \in SO(3)$ is a rotation matrix and ${}^{i-1}t_i$ is a translation vector. The set $\{{}^{0}T_1, {}^{1}T_2, \ldots, {}^{n-1}T_n\}$ represents all the motions of the camera. The main goal of pose estimation is to compute the relative transformation ${}^{i-1}T_\mathrm{i}$ using visual information. The current camera pose at the instant m, can be computed with the concatenation of the transformations $\mathrm{T}_i(i=0\ldots m)$, that is

$$ \begin{aligned} C_m &= {}^{0}\,\mathrm{T}_1{}^{1}\mathrm{T}_2\ldots{}^{m-1}\mathrm{T}_m \\ &= C_{m-1}{}^{m-1}\mathrm{T}_m \end{aligned} \tag{2} $$

In (Arun, Huang, and Blostein, 1987) the authors present an algorithm for finding the least-squares solution R and t that minimize (3). They propose to compute the translation part as the difference of the centroids of the 3D features and the rotation part using Singular Value Decomposition (SVD).

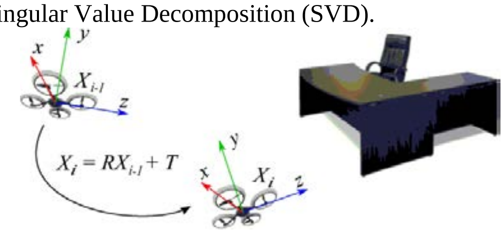

Figure 1. Pose estimation problem.

In this work, we focus on 3D features, and thus the problem can be described as finding the ${}^{i-1}T_{\mathrm{i}}$ that minimizes

$$\arg\min_{\mathrm{T}_i} \sum_{j=1}^{N} \left\| {}^{i-1}\mathrm{X}_j - {}^{i-1}\mathrm{T}_i {}^{i}\mathrm{X}_j \right\|^2 \tag{3}$$

where the subscript j denotes the j feature, and the superscript denotes the frame where the point is defined. The points with uppercase letters denote that they are expressed in homogeneous coordinates of 3D points i.e. $\mathrm{P}=\left[x, y, z, 1\right]^{\bullet}$

3 PARTICLE SWARM OPTIMIZATIONA

SOFT-computing algorithms are stochastic search methods inspired by biological behavior. In (Elbeltagi, Hegazy, and Grierson, 2005), the authors present a comparison study between: genetic algorithms (Holland, 1975), memetic algorithms (Merz, and Freisleben, 1997), particle swarm optimization (Kennedy and Eberhart, 1995), ant-colony optimization (Dorigo, Maniezzo, and Colorni, 1996), and shuffled frog leaping algorithm (Eusuff, and Lansey, 2003); they conclude that the Particle Swarm Optimization algorithm performs better in general, with respect to the quality of the solution and the success rate.

Particle swarm optimization (PSO) is a population-based optimization technique inspired by the social behavior of bird flock (Kennedy and Eberhart, 1995). The PSO algorithm starts with a population of particles whose positions represent the potential solutions for the given problem. The optimal solution is found by updating the particles positions in each generation, with

$$x_i\left(k\right) = x_i\left(k-1\right) + v_i(k) \tag{4}$$

where the velocities of the particles are

$$\begin{aligned} v_i\left(k\right) &= \alpha\phi_0 v_i\left(k-1\right) + c\phi_1\left(p_i - x\left(k-1\right)\right) \\ &+ c\phi_2\left(p_g - x_i\left(k-1\right)\right) \end{aligned} \tag{5}$$

where k is the number of iteration, x_i is each particle, v_i the velocity of the particle, α is an inertia factor, it makes the particle to keep its direction if this value is big. The terms ϕ_0, ϕ_1, ϕ_2 represent random values. The values p_i and p_g represent the best position of the particle and the best global position, respectively; these values are determined by the evaluation of some defined fitness measure. A review of PSO algorithm and its modification can be found in (Yuhui 2004).

Figure 2 shows an example of the algorithm evolution. As it can be seen, the particles, which represent solutions of the fitness function, converge to the desired minimum.

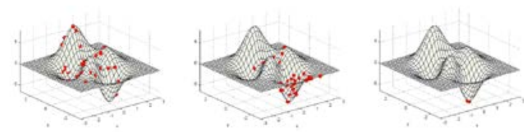

Figure 2. PSO evolution. Particles represent solutions of a particular fitness function.

4 POSE ESTIMATION USING PSO

IN this section, we explain the proposed approach to solving the pose estimation problem using PSO. The camera motion T_i can be computed by determining the aligning transformation of two 3D feature sets. Let $p_j = {}^{i}x_j$ denote the j-th feature defined at pose i, and let $p'_j = {}^{i-1}x_j$ denote the j-th feature defined at pose i-1.

The transformation between the features p_j and p'_j can be defined as

$$\arg\min_{\mathrm{R,t}} \sum_{j=1}^{N} \left\| \mathrm{p}'_j - \left(\mathrm{R}\mathrm{p}_j + \mathrm{t}\right) \right\|^2 \tag{6}$$

In (Arun et al., 1987), the authors showed that the rotation in (6) can be solved by translating the features p_j and p'_j with respect to their centroids and solve the following

$$\arg\min_{\mathrm{R}} \sum_{j=1}^{N} \left\| \mathrm{r}' - \left(\mathrm{R}\mathrm{r}_j\right) \right\|^2 \tag{7}$$

where $r_j = p_j - \bar{p}_j$, $r'_j = p'_j - \bar{p}'_j$, and where $\bar{p}_j, \bar{p}'_j$ denote the centroid of the features p_j and p'_j respectively.

The translation part can be computed as [3], that is

$$t = \overline{\mathrm{p}}_j' - R\overline{\mathrm{p}}_j \tag{8}$$

To solve this problem we use quaternions to represent rotations (Hu, Dixon, Gupta, and Fitz-Coy, 2006). Unit quaternions can be defined as

$$\mathbf{q} \triangleq [q_0 \;\; \mathbf{q}_v] \tag{9}$$

where

$$\begin{bmatrix} q_0\left(t\right) \\ q_v\left(t\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\dfrac{\theta\left(t\right)}{2}\right) \\ k\left(t\right)\sin\left(\dfrac{\theta\left(t\right)}{2}\right) \end{bmatrix} \tag{10}$$

and where $\mathrm{q}_v(t) \triangleq \left[q_1(t) \quad q_2(t) \quad q_3(t)\right]^{\bullet}$, $q_i(t) \in \mathbb{R}\ \forall_i = 0,\ldots,3$ and the following constraint must be satisfied

$$\mathbf{q}^T\mathbf{q} = 1 \tag{11}$$

Given a unit vector *k(t)* and the angle *θ(t)*, the rotation matrix R*(t)* can be calculated with the Rodrigues formula (Ma, 2004) and can be expressed as

$$\mathrm{R}(\mathrm{q}) = \begin{bmatrix} q_0^2+q_{v1}^2-q_{v2}^2-q_{v3}^2 & 2(q_{v1}q_{v2}-q_{v3}q_0) & 2(q_{v1}q_{v3}+q_{v2}q_0) \\ 2(q_{v1}q_{v2}+q_{v3}q_0) & q_0^2-q_{v1}^2+q_{v2}^2-q_{v3}^2 & 2(q_{v2}q_{v3}-q_{v1}q_0) \\ 2(q_{v1}q_{v3}-q_{v2}q_0) & 2(q_{v2}q_{v3}+q_{v1}q_0) & q_0^2-q_{v1}^2-q_{v2}^2+q_{v3}^2 \end{bmatrix} \quad (12)$$

The PSO fitness function is defined as follows

$$f(q) = \left\| r_j' - (q r_j q^{-1}) \right\|^2 \quad (13)$$

The expression $q r_j q^{-1}$, can be evaluated as suggested by (Jia, 2008) and (Salamin, 1979), that is

$$q r_j q^{-1} = \cos(\theta) r_j + \sin(\theta) k \times r_j + (1-\cos(\theta)) k (k \cdot r_j) \quad (14)$$

where $k = [x \quad y \quad z]^{\bullet}$ and θ are the arguments of the quaternion.

It is important to note that each quaternion has three degrees of freedom due to the constraint (11). Therefore, since each PSO particle represents a feasible quaternion it should minimize a vector of dimension three.

The advantage of the PSO approach with respect to the pure SVD (Arun et al., 1987), is that the former always generates a valid quaternion and thus a valid rotation, whereas the later can obtain a solution that minimizes the fitness function, but the resulting matrix is not a valid rotation matrix, i.e. det(R)=1.

A diagram of the proposed approach is shown in Figure 3. The steps are (a) image capture, (b) feature extraction, c) the robots moves to another position, d) the disparity image can be computed, e) with the two sets of points the outliers are detected and removed using the PSO. Once the outliers are removed the rigid transformation can be computed.

5 OUTLIER REJECTION WITH PSO

THE SVD algorithm (Arun et al., 1987) is not robust to the presence of outliers. To overcome this problem we introduce the concept of supporting features in our PSO approach.

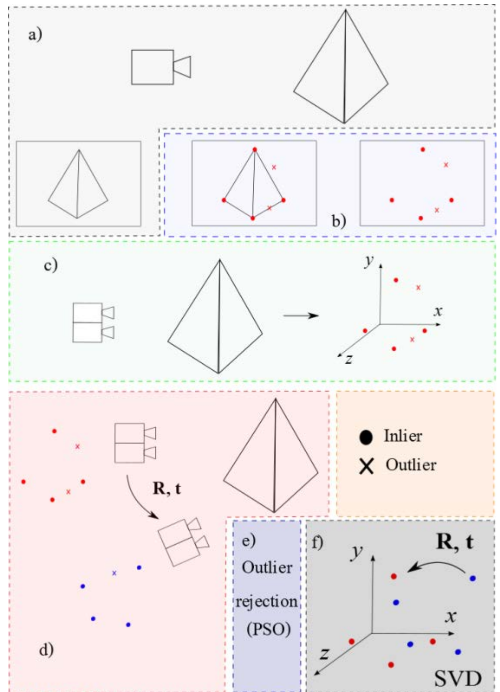

Figure 3. Diagram of the algorithm. a) Image capture. b) Feature extraction. c) 3D information obtained with stereo vision. d) Two sets of points, due to cameras movement. e) Outlier rejection with PSO. f) Pose estimation with SVD.

A pair of corresponding features r_j and r'_j will be considered a support vector for a given quaternion q using the following function

$$s(r_j, r_j') = \begin{cases} 1 & \textbf{if } \left\| r_j' - (q r_j q^{-1}) \right\|^2 < \delta \\ 0 & \textbf{otherwise} \end{cases} \quad (15)$$

where δ is a threshold value. This value should be stored by the PSO particles, and thus the best particle would be chosen as the particle with more support features which minimizes the fitness function. At the end of the PSO algorithm, all the features that do not support features are considered as outliers, and they are removed from the final refinement step. It is also important to see that the numbers of support features must be greater than three, otherwise, features will be collinear or coplanar and SVD could compute a reflection matrix instead of a rotation matrix. Four features are the minimum necessary to ensure a rotation matrix after SVD (Arun et al., 1987). The process is shown in Figure 4.

6 SIMULATIONS

IN the simulation experiments we generate a set of 3D points, then the camera was moved in the scene. Therefore, the points are not moving but the camera,

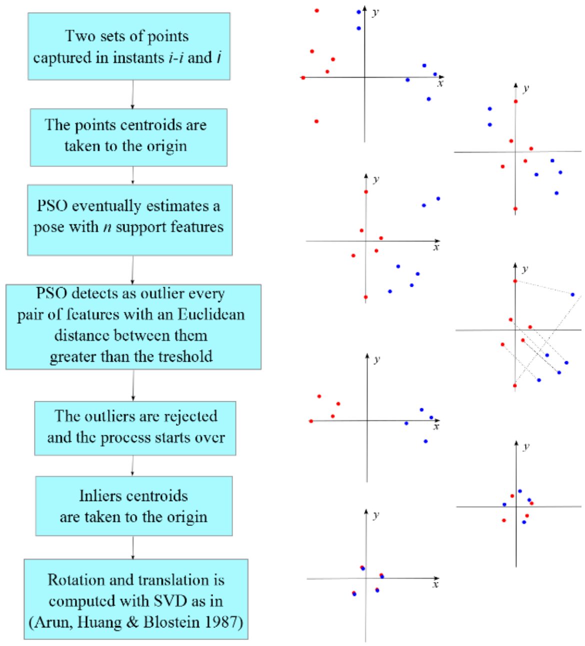

Figure 4. **Outlier rejection with PSO, n must be greater than 3, otherwise, SVD cannot ensure a rotation matrix as described in section 4.1.**

and thus the camera motion will produce the two sets of points p_j and $\bar{p}_j$. Since the pose of the camera is known we can project the points into their respective coordinate frame, and this will provide the two sets of 3D features as depicted in Figure 5.

6.1 Simulation setup

In Figure 5, we can see vector *t*, which can be computed with (8). To find *R*, both data sets will be taken to the origin and the SVD algorithm will be used to compute rotation only. If the algorithm fails, PSO will compute the rotation matrix.

Additionally, noise will be randomly added to the second set of data and the camera projection matrix *P* will be applied with (16) to both sets such that only the points that are visible will be part of the minimization. A point *X* in 3D space will be projected with

$$\mathrm{x} = \mathrm{PX} \tag{17}$$

where

$$\mathrm{P} = \mathrm{K}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{18}$$

and where

$$\mathrm{K} = \begin{bmatrix} \alpha_x & 0 & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

represents the calibration matrix with α_x and α_y the focal length of the camera and c_x and c_y the image center.

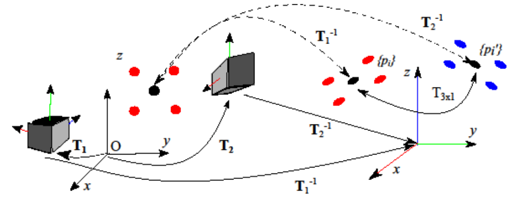

Figure 5. **Inverse transformations applied to cameras and sets of points. T1 and T2 are transformations of the form Rx+t where R is a 3x3 matrix and t is a 3x1 vector.**

In the simulation, we test if the 3D features are projected into the boundaries of the image, this allows us to test if a feature is visible.

6.2 Simulation results

In the first simulation, each algorithm runs 50 times. The combination PSO-SVD was compared with the SVD and de M-estimator Sample Consensus (MSAC) algorithm for a different number of outliers. MSAC is a variation of the Random Sample Consensus (RANSAC) and According to (Torr and Zisserman, 2000); the implementation of this new method yields a benefit to all robust estimations with no computational burden and hence, there is no reason to use RANSAC in preference to this method.

The error in position (in meters) for a different number of outliers is plotted and shown in figure 6 with a different noise level for the three algorithms. In Figure 7, the processing time for PSO/SVD and MSAC is shown for a different number of outliers and with a different noise level for both algorithms.

7 EXPERIMENTS

FOR the real tests, a set of consecutive images were taken with the Bumblebee XB3® stereo camera, see Fig. 8. In the first experiment, a smooth translation in the *z*-axis was performed with no rotation.

The feature extraction was performed with intensity data. There are many feature extraction algorithms, one of the most used methods is the computation of gradients in order to locate the points in the image with high textural derivatives (Gedik and Alatan, 2013). In our case, the features detection and description were performed with the SURF algorithm (Bay, Ess, Tuytelaars, and Van Gool, 2008; Bay, Tuytelaars, and Van Gool, 2006), and the result of applying SURF for two consecutive images is shown in Figure 9. The location *x* and *y* of every corner detected with the algorithm will have a value in *z* provided by the stereo camera.

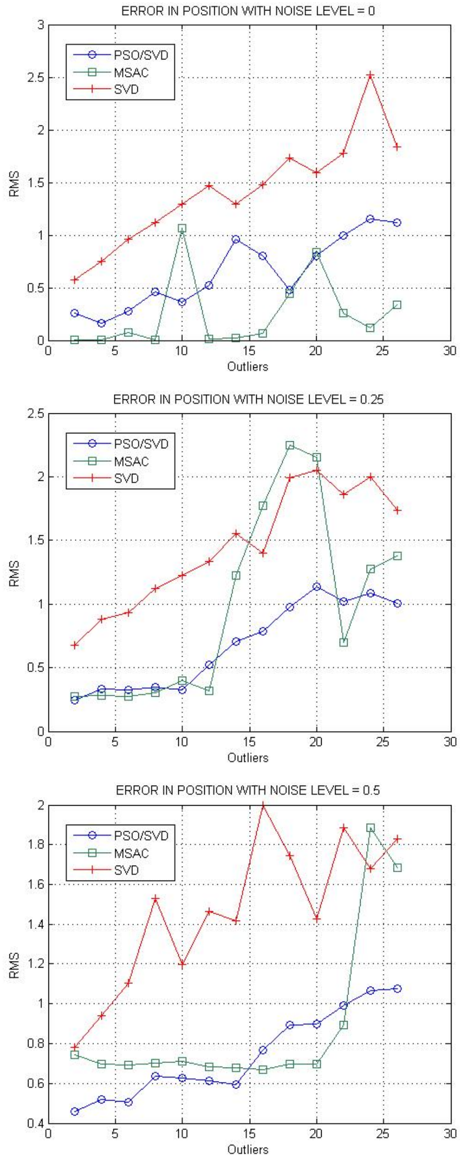

Figure 6. **Error in position with different noise levels.**

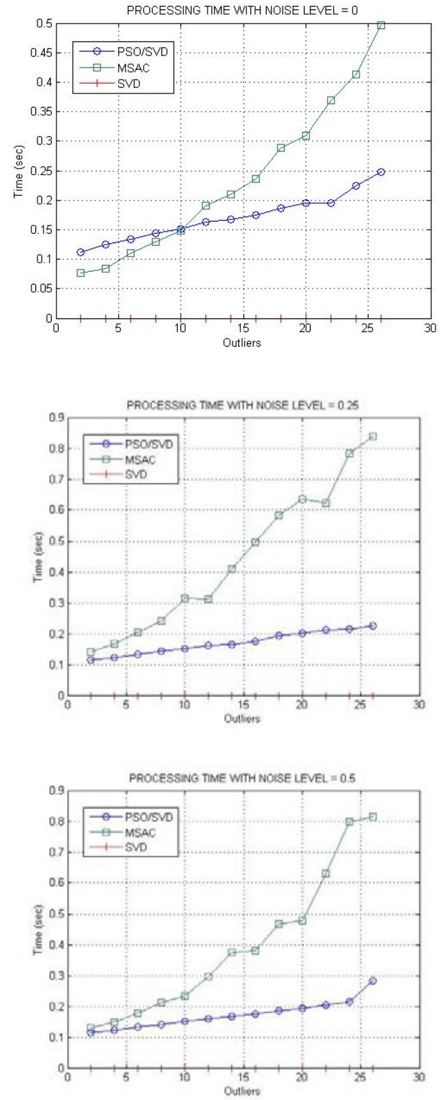

Figure 7. **Processing time with different noise levels.**

It can be seen in Fig. 9 and Fig. 10 that the SURF algorithm provides some incorrect matches among the points. Both MSAC (Figure 11) and PSO (Figure 12) algorithms were implemented and compared with these images in order to test their ability to reject outliers.

In the second experiment, a smooth rotation was applied to get 31 different orientations; some of them are shown in Figure 13. In this case, the orientation provided by MSAC and PSO/SVD algorithms will be compared with an *Inertial Measurement Unit* (IMU) because of its high precision. The IMU used in this experiment is the Xsens® MTi-G-700-2A5G4.

The error in the experiments was computed with

$$\left\| I - R_{IMU} R^{T} \right\|_{F} \tag{20}$$

where I represent the identity matrix of 3x3, R_{IMU} the rotation provided by the IMU and R^T the rotation matrix calculated with either MSAC or PSO/SVD.

Figure 14 shows five rows of images, every row shows a subset of images that correspond to different experiments, including the experiments in Figure 8 and 13. In Table, 1, RMS errors of each experiment are shown.

In the third experiment, the camera was mounted on a KUKA© youBot robot and the position computed by the algorithm was compared with the pose computed by the robot using its encoders. The configuration is shown in Figure 15. The images of this experiment are shown in Figure 16.

Figure 8. **Set of consecutive images. A smooth translation in the z-axis is performed with no rotation.**

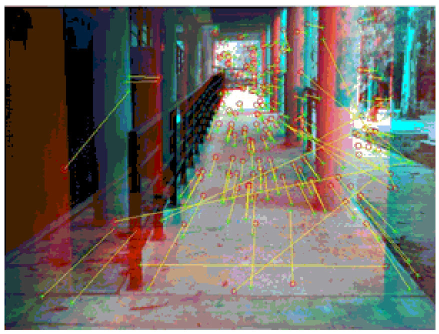

Figure 9. **SURF applied to consecutive images of Figure 5.**

Figure 17 shows the comparison between the algorithms. The lines represent the trajectory of the robot on the XY plane and each point represents the place where the image was captured.

8 CONCLUSIONS

IN this paper, the authors have presented a PSO approach for pose estimation using visual information. The proposed approach uses a quaternion in the fitness function to find the best rotation between two data sets. The concept of support features is introduced to select the PSO particle with more features that support the given quaternion, and therefore the approach is able to reject outliers. Finally the results show that the performance of the proposed approach is not affected by the outliers. It is also shown that even when the correspondences are correct in 2D is necessary to implement the PSO approach because of the inaccuracies present in stereo vision. Furthermore, the proposed approach always generates a valid quaternion and therefore a valid rotation.

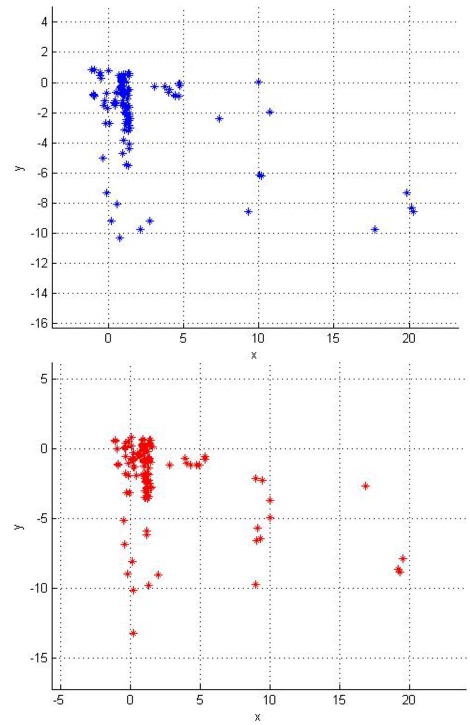

Figure 10. **a) Original and b) rotated data sets.**

9 ACKNOWLEDGMENTS

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

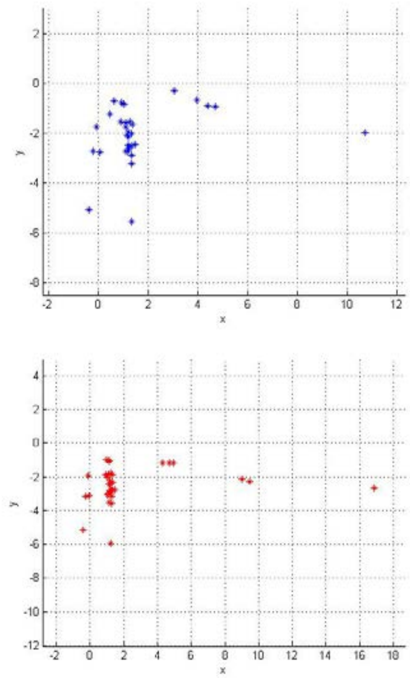

Figure 11. a) Original and b) rotated data sets after MSAC.

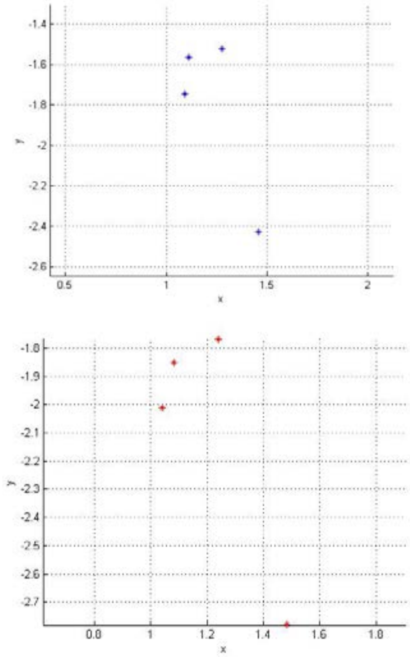

Figure 12. a) Original and b) rotated data sets after PSO.

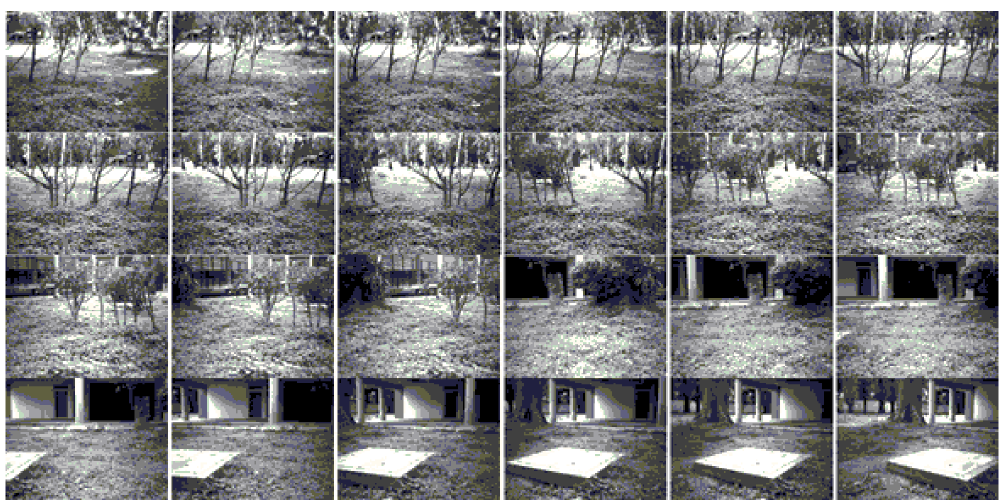

Figure 13. Set of consecutive images for experiment 5. Rotation only. The complete experiment consists of 31 images.

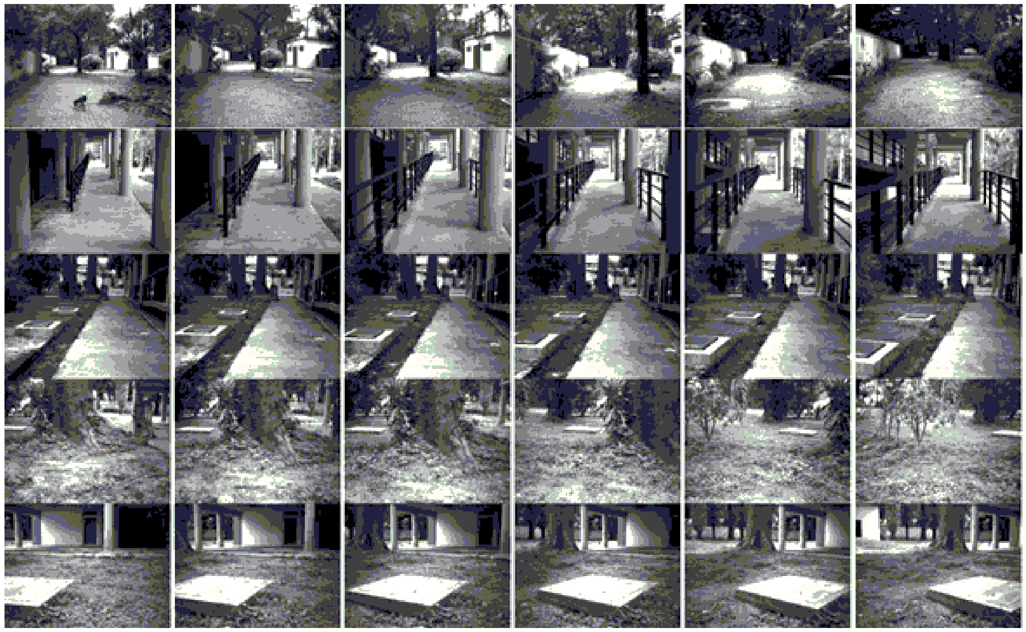

Figure 14. Subset of images for every experiment performed. Every row represent a different experiment.

Table I. Error comparison between algorithms for each experiment

Experiment	Images	RMS_{MSAC}	$RMS_{PSO/SVD}$
1	19	0.7092	0.1067
2	9	0.8644	0.1284
3	18	0.5501	0.0280
4	17	0.5739	0.1330
5	31	0.4475	0.1591

Figure 15. Bumblebee XB3 mounted on the KUKA© youBot for position experiments.

Figure 16. Set of images for the position experiment.

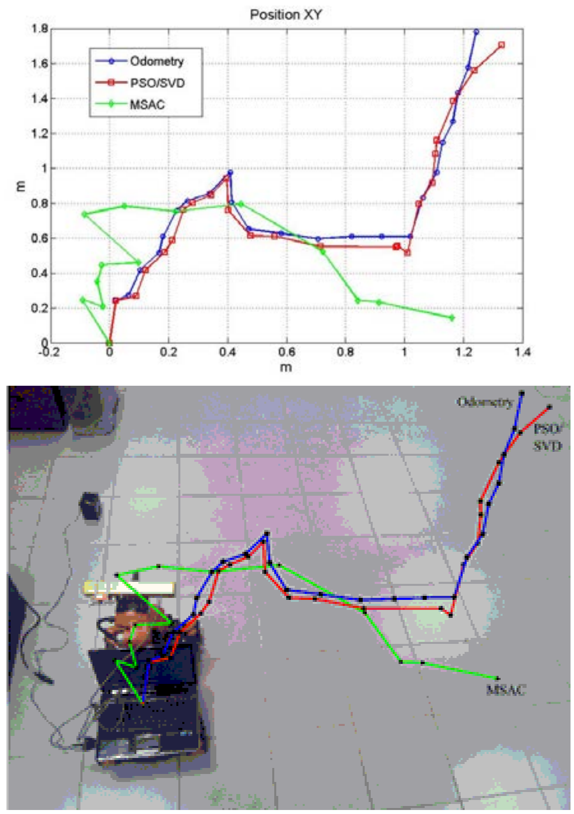

Figure 17. Trajectories computed by odometry and the two algorithms.

REFERENCES

A. Y. Alanis, E. Rangel, J. Rivera, N. Arana-Daniel, and C. Lopez-Franco. (2013). Particle swarm based approach of a real-time discrete neural identifier for linear induction motors. Mathematical problems in engineering, 2013.

K. S. Arun, T. S. Huang, and S. D. Blostein. (1987). Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5), pp. 698–700.

A. A. Bakar, A. R. Hamdan, and M. Z. A. Nazri. (2010, November). A Rough set outlier detection based on Particle Swarm Optimization. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on* (pp. 1021-1025). IEEE.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

H. Bay, T. Tuytelaars, and L. Van Gool. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, (pp. 404–417). Springer Berlin Heidelberg.

W. N. Chen, J. Zhang, Y. Lin, N. Chen, Z. H.. Zhan, H. H. Chung, and Y. H. Shi. (2013). Particle swarm optimization with an aging leader and challengers. *Evolutionary Computation, IEEE Transactions on*, 17(2), 241-258.

Y. M. Choi, Y. S. Suh, and S. K. Park. (2006, October). Pose Estimation from Landmark-based vision and inertial sensors. In *SICE-ICASE International Joint Conference*, pp. 1668-1671.

M. Dorigo, V. Maniezzo, and A. Colorni. (1996). Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, *IEEE Transactions on*, 26(1), 29-41.

A. E. Eiben and J. E. Smith. (2008). Introduction to evolutionary computing (natural computing series).

E. Elbeltagi, T. Hegazy, and D. Grierson. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics*, 19(1), 43-53.

M. M. Eusuff and K. E. Lansey. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129(3), 210-225.

O. S. Gedik and A. A. Alatan. (2013, July). Fusing 2D and 3D clues for 3D tracking using visual and range data. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 1966-1973). IEEE.

H. Geng and Q. Hu. (2013). Feature-matching and extended kalman filter for stereo ego-motion estimation." In *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, pp. 242–246.

D. E. Goldberg and J. H. Holland. (1988). Genetic algorithms and machine learning. Machine learning, 3(2), 95-99.

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

J. H. Holland. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.

G. Hu, W. E. Dixon, S. Gupta, and N. Fitz-Coy. (2006, May). A quaternion formulation for homography-based visual servo control. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (pp. 2391-2396). IEEE.

R. Jain, R. Kasturi, and B. G. Schunck. (1995). Machine vision (Vol. 5). New York: McGraw-Hill.

Y.-B. Jia. (2008). Quaternions and rotations. Handout of the Problem Solving Techniques for Applied Computer Science Lecture at Iowa State University, 52.

M. Jin and E. Wu. (2013). Collision-free and energy-saving trajectory planning for large-scale redundant manipulator using improved pso. *Mathematical Problems in Engineering*, 2013.

L. Junmin, W. Jinge, Z. Wentian, J. Shiwei, and L. Zhen. (2013, December). Robot Pose Estimation and Accuracy Analysis Based on Stereo Vision. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on* (pp. 555-559). IEEE.

N. Kaempchen, U. Franke, and R. Ott. (2002, June). Stereo vision based pose estimation of parking lots using 3D vehicle models. In *Intelligent Vehicle Symposium*, 2002. IEEE (Vol. 2, pp. 459-464). IEEE.

J. Kenndy and R. C. Eberhart. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942-1948).

C. López-Franco, L. Villavicencio, N. Arana-Daniel, and A. Y. Alanis. (2014). Image classification using PSO-SVM and an RGB-D sensor. *Mathematical Problems in Engineering*, 2014.

Y. Ma. (Ed.). (2004). *An invitation to 3-d vision: from images to geometric models* (Vol. 26). Springer Science and Business Media.

M. Minoux. (1986). Mathematical programming: theory and algorithms. John Wiley & Sons.

E. Salamin. (1979). Application of quaternions to computation with rotations. Working Paper.

Y. Shi. (2004). Particle swarm optimization. *IEEE Connections*, *2*(1), 8-13.

B. Steder, G. Grisetti, S. Grzonka, C. Stachniss, A. Rottmann, and W. Burgard. (2007, October). Learning maps in 3d using attitude and noisy vision sensors. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on (pp. 644-649). IEEE.

G. S. Tewolde. (2013, May). Evolutionary learning for improving performance of robot navigation. In *Electro/Information Technology (EIT), 2013 IEEE International Conference on* (pp. 1-5). IEEE.

G. Tong, R. Liu, and H. Li. (2012, May). The monocular model-based 3D pose tracking. In *Control and Decision Conference (CCDC), 2012 24th Chinese* (pp. 980-985). IEEE.

P. H. Tor and A. Zisserman. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding, 78*(1), 138-156.

G. Welch and E. Foxlin. (2002). Motion tracking survey. *IEEE Computer graphics and Applications*, 24-38.

M. Yang, B. Dong, H. Wang, and B. Zhang. (2002, June). Laser radar based real-time ego-motion estimation for intelligent vehicles. In *Intelligent Vehicle Symposium, 2002*. IEEE (Vol. 1, pp. 44-51). IEEE.

NOTES OF CONTRIBUTORS

Carlos López-Franco received the 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.

Javier Gomez-Avila is a PhD student in Computer Science Department, at Centro Universitario de Ciencias Exactas e Ingeniería at University of Guadalajara, Mexico. He received his M. S. degree in computer science in 2014 at University of Guadalajara where he is currently working in the Intelligent Systems Research Group. His research interest include computer vision, visually guided robotics and artificial neural networks.

Nancy Arana-Daniel received in 2003 the M. S. degree in Computer Science and the 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.

Alma Y. Alanis, received the Ph.D. degrees 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.