Pose Estimation with Multiple Sources Using Evolutionary Algorithms

Claudio Rossi, Antonio Barrientos, Jaime del Cerro, Pedro Gutierrez, Alexander Martinez and Rodrigo San Martin
Departamento de Automática,
Ingeniería Electrónica e Informática Industrial
Universidad Politécnica de Madrid, Spain
Claudio.Rossi@upm.es

Abstract **– An important feature of heterogeneous multi-robots systems is the availability of sensing information coming from multiple sources or points of view. Team members can exchange localization information regarding a target, or provide extra sensor information to team-mates that lack of certain sensing capabilities. In this paper, we present two examples of multi-robot pose estimation in a team of robots composed by unmanned aerial and terrestrial vehicles. Information integration is achieved through an evolutionary-based vision algorithm. Experiments have been performed part on virtual-reality simulations and part in-field.**

I. Introduction

Field robots are meant to work in outdoor locations that are uncomfortable or dangerous for the human being, and find applications in agriculture, mining, search, rescue etc., in such different natural environments as sea, ground, air or even space. The possibility of using robots out of controlled environments such as workshops or industrial plants poses important challenges, and opens new scenarios of application and research issues.

The growing complexity of the scenarios of application and of the duties the robots are asked to perform, can be tackled by teams of robots that cooperate among them (and with humans). From the point of view of performances, robot teams can improve execution time of the task, provide a more reliable and fault-tolerant system, and can make the planning and execution of the task easier thanks to an adequate sharing of the duties [1], or by sharing the information gathered.

Localizing a common target or a team-mate is an important task that can take advantage of the cooperation between robots, since position data can be exchanged between team-mates that in this way receive additional information from a different point of view or with a different kind of sensor. Such information can be useful in order to improve the precision of the estimated localization of the target, and sometimes can be indispensable to make the localization possible.

In this paper we present two examples of information integration achieved through the extension of an evolutionary-based vision algorithm [2] that offers the possibility to include information coming from multiple sources in the search for the pose (position and orientation) of a target object. The use of multiple sources helps both the search process itself and the accuracy of the determined pose.

The effectiveness of the proposed approach is assessed with two experimental setups, involving unmanned aerial (UAV) and ground (UGV) vehicles. The first involves the team's supervising station and a small airship. The task is to compute the airship's pose. In this case, the main problem is the payload of the airship, which is extremely limited and does not allow it to be equipped with suitable instrumentation. The complete six coordinates localization information is reconstructed exchanging information with an external source capable of computing its position. In this case, high-level processed information is exchanged. The same strategy can be used in order to provide position information to a vehicle equipped with GPS hardware in case of temporary loss or degradation of the GPS signal.

In the second experiment two helicopters equipped with camera have to localize a ground vehicle. To this purpose, they exchange either low-level information (raw images), mid-level information (pre-processed image data) or high-level information (localization coordinates). In the first case the two UAVs send to each other the image they are capturing, and each UAV can use it in order to compute the pose of the target object. In the second case the two UAVs send to each other the feature points detected after a pre-processing of the image. In this way each helicopter can search for the target in two sets of image points taken from different viewpoints. In the last case, the UAVs exchange high-level information, i.e. the estimated position, and each vehicle incorporates such information in its search algorithm as an external hint.

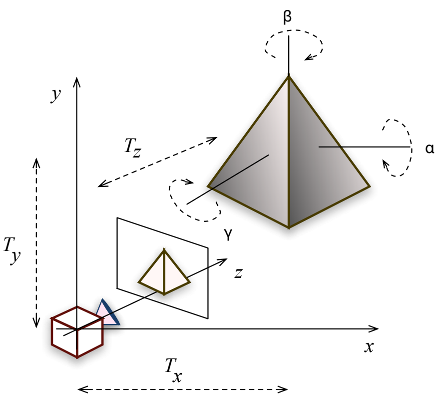

Fig. 1. Geometry of the problem. The camera is positioned in the center of the coordinate system

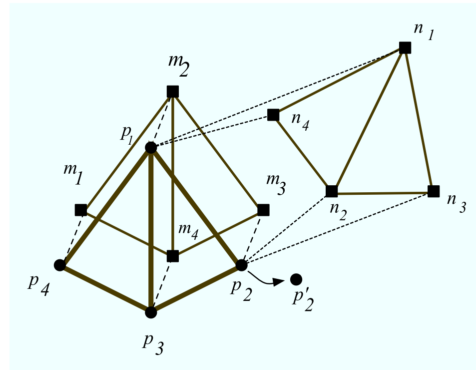

Fig. 2. Distance and matching points. Points $\{n_i\}$ and $\{m_i\}$ are the features of the model projected according to two set of pose parameters. Points $\{p_i\}$ are feature points extracted form the image. Feature correspondence is shown with dashed lines

The problem of pose estimation is central in computer vision, and has been approached with a vast variety of methods, like neural networks [3], linear programming [4], probabilistic model of appearance [5] and many others. One of the most complete pose estimation algorithms is SoftPOSIT [6]. For a comprehensive survey of different techniques of 3D object modeling, correspondence and pose estimation, [7] and [8] are recommended.

In [9], a probabilistic approach for collaborative multi-robot localization based on Markov localization is proposed. A distributed Extended Kalman Filter-based algorithm for the localization of a team of robots operating on outdoor terrain is developed in [10]. In the same paper, an excellent review of papers on cooperative localization can be found, with the observation that cooperative robotic navigation research primarily concentrates on indoor environments.

Other common sensor fusion approaches rely on statistics methods as maximum likelihood estimation, Kalman filtering, Bayesian theory and Dempster-shafer theory, or soft-computing techniques like neural networks and fuzzy sets. A brief review of the above mentioned techniques can be found in [11].

II. Evolutionary pose estimation

Given a bi-dimensional camera image of an object, the *pose estimation* problem consists in finding the objects position and orientation in space. Model-based methods make use of a model of the object of interest, and attempt to match some of its key features such as edges, marks, vertices, etc. to the contents of the image. A common approach is to divide the search in two steps: first, features are extracted from the image and second, a matching is looked for.

A. *The* EvoPose *Algorithm*

The pose estimation problem can be formulated as an optimization problem considering the distance of each feature in the image with the corresponding feature in the model, projected in the image plane according to six position and orientation parameters $s = (T_x, T_y, T_z, \alpha, \beta, \gamma)$(see Fig. 1). The matching problem is then turned into the optimization problem of finding the values of the array s such that the total distance is minimum. Since the correspondence between points is not known in advance, a projected model feature is considered to match the closest image feature. Thus, the distance measure is:

$$d = \sum_i min_j(|p_i - m_j(s)|) \tag{1}$$

where p_i is a feature point in the image, $m_i(s)$ is the model point projected according to the set of parameters $s = (T_x, T_y, T_z, \alpha, \beta, \gamma)$ and $|\cdot|$ is the distance between two-dimensional points (see Fig. 2).

If some of the points are noisy, missing or are added (false positives) the minimum total distance d will not be zero. This fact does not affect the minimum distance, that in most cases still relates to a good point matching, as the perturbation affects all candidate solutions. This is shown in Fig. 2, where point p_2 is moved to p'_2: although the total distance is changed, set $\{m_i\}$ still matches better than set $\{n_i\}$. This makes the matching algorithm robust to noisy images, poor image quality, partial occlusions and/or to faults in the feature-extraction process.

Nevertheless, especially with highly noisy images, the objective function presents local minima, which means that there are combinations of parameters that lead to a small total distance, while the projected model points poorly matches the image points. For this reason it is important to rely on search methods such as evolutionary algorithms (EAs) [12], where search is done sampling the solutions space and has the capacity of overcoming local optima[1]. In the *EvoPose* algorithm, the array s is a

[1] Evolutionary (or genetic) algorithms are a class of stochastic parallel search algorithms based on the principles of Darwins Evolution Theory. Evolutionary algorithms maintain a set of individuals, called *population* where each individual encodes a candidate solution of the problem at hand, and is called *chromosome*. Each chromosome has associated a measure of its goodness with respect to the problem, which is called its *fitness*. At each step, called *generation*, a new population is generated applying the *genetic operators* of *selection, crossover* and *mutation*. Generation after generation, good solutions emerge (evolve) towards optimality. EAs have shown their power as search procedures in several applications, especially in presence of noisy input data, and when problem-specific knowledge can be formulated as a cost function, as in the case of the problem under study.

candidate solution, encoded as a six-dimensional array of real numbers, and a good solution is sought with an evolutionary algorithm, which maintains a population of candidate solutions and evolves them toward optimality by means of the genetic operators.

Note that the computational efforts for computing a solution only increases polynomially with the number of points under examination. This makes the scale-up behavior of the algorithm suitable for real images, where hundreds of points may need to be considered. In earlier work [2], *EvoPose* has demonstrated to be competitive with the well known SoftPOSIT algorithm [6], one of the most complete algorithms that deal with the correspondence and pose estimation simultaneously.

In case of moving objects, the problem consists in determining the pose of the object in a sequence of images, where the position and/or orientation of the object vary in time. In this case, the set of image points $P = \{p_i\}$ changes in time, becoming $P(t)$. Thus, distance function (1) becomes a function of time:

$$d(t) = \sum_i min_j(|p_i(t) - m_j(s)|) \tag{2}$$

Once the pose of the object has been determined for an image, this information can be used in order to determine the position of the object in the following image, without having to solve from scratch the pose problem for the new image. The efforts needed to compute a new pose are then reduced considerably, depending on the rate at which the pose changes. As in EAs new candidate solutions are generated searching in the neighborhood of the best existing individuals in the population, the information about the previous position is actually used in order to find the new one. Working in sequences of images turns out to help the search process, since noise in the image tends to vary. Thus, bad local minima due to noise may appear and disappear, while object features persist, although slightly moved, in subsequent frames. These correspond to a strong minimum, the correct pose.

III. *EvoPose* AND COOPERATIVE LOCALIZATION

The *EvoPose* algorithm can incorporate localization information coming from other sources, injecting the information into the candidate solutions. For example, if the distance from the object has been detected by a laser or infra-red sensor, then the value corresponding to the distance in the solution can be forced to be the known value, and the evolutionary algorithm will search for the remaining five. Let $k \in \{1 \ldots 6\}$ be the index of the known value and C its value. Then equation (1) becomes:

$$d = \sum_i min_j(|p_i - m_j(s)|) \quad / \quad s_k = C \tag{3}$$

Note that in this case the value is known and considered to be fully reliable, and the search space is reduced by one dimension. More in general, since the value C is known within a certain precision ε, a better strategy is to incorporate the uncertainty of the measurement by constraining the value of the corresponding parameter to the range $[C - \varepsilon, C + \varepsilon]$, reducing considerably the search space, but still allowing the search algorithm a certain degree of freedom.

The distance-based fitness function (1) can further be extended to include the information coming from another image, adding a term corresponding to the matching of the solution projected into the second image. In this case, there will be two sets of points $P^{(1)} = \{p_i^{(1)}\}$ and $P^{(2)} = \{p_i^{(2)}\}$, coming from the pre-processing of each image, and two sets of model points, $M^{(1)} = \{m_i^{(1)}\}$ and $M^{(2)} = \{m_i^{(2)}\}$, projected according to the orientation of each camera. The resulting distance function will be

$$\begin{aligned} d = W_1 \cdot d^{(1)} + W_1 \cdot d^{(2)} = \\ = W_1 \sum_i min_j(|p_i^{(1)} - m_j^{(1)}(s)|) + W_2 \sum_i min_j(|p_i^{(2)} - m_j^{(2)}(s)|) \end{aligned} \tag{4}$$

which takes into account the matching of the candidate solution with the two images, and reaches its minimum values only when both terms are low, i.e. when the computed position and orientation is in accordance with both input images. The values W_1 and W_2 weight the two distance terms, and are used to tune the relative importance of the information coming from the two images. In general, they are computed according to some heuristic rule, in the simplest case a step function: if one term exceeds a given threshold it may be the case that the corresponding images is not reliable, or does not contain the object looked for. Then, it is possible to ignore it by setting its weight to 0.

In case the information coming from the external source is a full solution, a third extension can be used. Complete external localization information can be incorporated as an "hint" in the search algorithm in form of new individuals, that will mix with the existing population of candidate solutions, spreading the information they carry. Since the selection mechanism of the EAs selects candidate solutions with a probability proportional to its goodness, good hints will be taken into account with high probability. Hence, in case the information received is bad or incorrect, in the sense that it does not fit well own image data, the selection mechanism of the EA will discard the received individuals, and the hint will not be trusted.

IV. EXPERIMENTS

In the experiments, the features (marks) to be matched are extracted in a pre-processing step consisting in the application of standard filtering and corner detection algorithms [13]. Color marks have been used as features.

A. *Experiment 1*

In experiment 1, an airship of 1.85 meters of length, equipped with a light-weight inertial unit (*XSens MT9*) providing its attitude, and bi-directional communication link has been used. The distance of the airship form the camera was about 5 meters. Its position was computed on ground by a fixed station, equipped with a camera (*Canon VC-C4*, manually calibrated) and running

TABLE I.

RESULTS OF EXPERIMENT 1. POSITION AND ATTITUDE VALUE ARE AVERAGE OVER 10 RUNS.

Algorithm	Orientation error (deg.) Roll (σ_{Roll})	Pitch (σ_{Pitch})	Yaw (σ_{Yaw})	Position error (cm) X (σ_X)	Y (σ_Y)	Z (σ_Z)	Time (sec)
EvoPose	20.6 (15.2)	1.8 (2.4)	2.9 (3.6)	5 (4.6)	9 (6.9)	16 (15)	3.2
EvoPose^{+IMU}	0.1 (0.1)	0.1 (0.1)	0.1 (0.1)	6 (4.2)	9 (7.1)	15 (16)	2.3

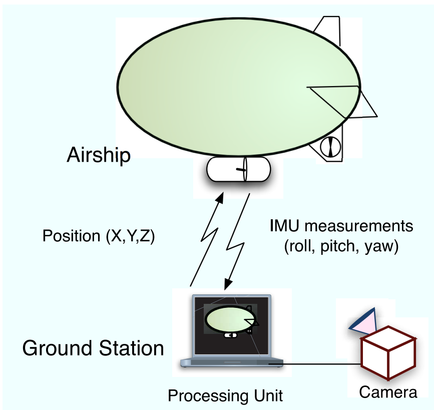

Fig. 3. SETUP OF EXPERIMENT 1

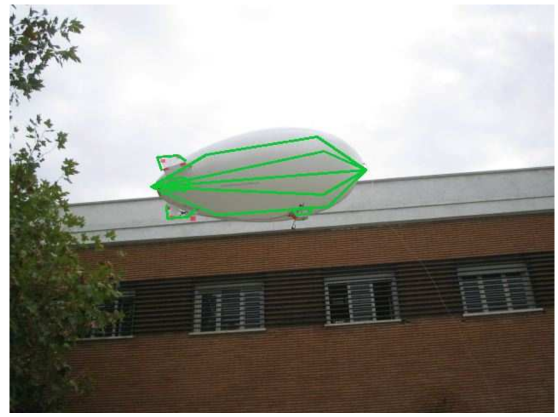

Fig. 4. EXPERIMENT 1: EXAMPLE OF MATCHING POSE

the *EvoPose* algorithm, extended to include external information (see Eq. 3). After the pose calculation, the computed position was sent back to the airship. In this way, the position and orientation of the airship could be known both by the ground station and the airship itself (and to any other team members if needed). Fig. 3 depicts the experiment.

The experiment was aimed at testing the possibilities of the *EvoPose* algorithm when external information is included. As stated before, in video sequences of a moving object, it is not necessary to re-compute the pose in each frame, since information regarding the previous one can be effectively used to compute the new one. For this reason, the experiment has been restricted to the finding of the first pose, which is the most difficult and expensive stage of the search. When a good initial pose has been found, *EvoPose* has been shown to be able to track its changes in real time. The algorithm was run on a fixed image (snapshot) taken at a give time, using the orientation information sent to ground by the airship.

Table I reports on the results of experiment 1, comparing the results of the plain *EvoPose* algorithm with the improved version using the attitude information coming from the airship's IMU (*EvoPose*$^{+IMU}$). The table reports the average absolute attitude error (roll, pitch, yaw, rounded to decimals), along with the respective standard deviations σ, w.r.t. the known attitude, and the average position error of the localization (rounded to units), along with the respective standard deviations σ. Since the position of the airship was unknown, such figures are relative to the mean values. The last column reports the average time needed to compute a good pose the first time, where "good" means that the fitness of the best solution was under a predetermined threshold, fixed during early testing of the algorithm. Position, attitude and time values in Table I are average over 10 runs.

The accuracy of the position is roughly the same in the two cases, although incorporating the external sensor data improves performances of the algorithm, since the time needed to find a solution is lower, indicating that it is easier for the algorithm to find the position of the target object if its orientation is known. As far as orientation is concerned, the detected values and their standard deviation coincide with the accuracy of the on-board IMU, since the search was constrained to the small interval around its precision.

The hardest values to find are the distance from the camera to the airship and its roll. These results are not surprising, since using only one camera displacements along the line of sight are estimated by the change in the dimensions of the target object, which are difficult to detect. Due to the geometry of the airship, changes in roll were also expected to be difficult to detect. Integrating the sensor data coming from the airship has overcame this problem.

B. *Experiment 2*

In the second experiment only computer vision was used. Two helicopters equipped with camera had to compute the pose of a ground vehicle. The target vehicle was an Activmedia Pioneer-2 DX robot (VRML model), with a dimension of $(44 \times 38 \times 22)$ cm, and the UAVs were placed at a distance (Z axis) of 30 and 20 meters, at varying altitudes (Y axis) and displacements along the X axis. Figure 5 depicts the experiment. The positions and attitude of the helicopters was known without any uncertainty, and the target object was not moving. The

TABLE II.
RESULTS OF EXPERIMENT 2. POSITION AND ATTITUDE VALUE ARE AVERAGE OVER 10 RUNS.

Cameras	Orientation error (deg.)			Position error (m)			Time
	Roll (σ_{Roll})	Pitch (σ_{Pitch})	Yaw (σ_{Yaw})	X (σ_X)	Y (σ_Y)	Z (σ_Z)	(sec)
H1	1.3 (1.7)	1.0 (2.1)	0.1 (0.3)	1.0 (0.4)	0.5 (0.2)	2.3 (1.8)	2.3
H2	1.1 (1.6)	1.0 (3.3)	0.5 (1.3)	1.0 (0.7)	1.2 (1.3)	2.2 (3.3)	> 5
(a)							
H1 + H2	1.1 (1.8)	0.1 (0.3)	0.0 (0.1)	0.1 (0.1)	0.2 (0.2)	0.9 (0.4)	1.6
(b)							
H1 + $hint$	0.6 (1.0)	1.0 (1.8)	0.0 (0.0)	0.9 (0.3)	0.3 (0.1)	1.6 (0.9)	0.6
H2 + $hint$	0.2 (0.1)	0.1 (0.5)	0.0 (0.1)	0.6 (0.2)	0.7 (0.2)	0.7 (0.4)	1.3

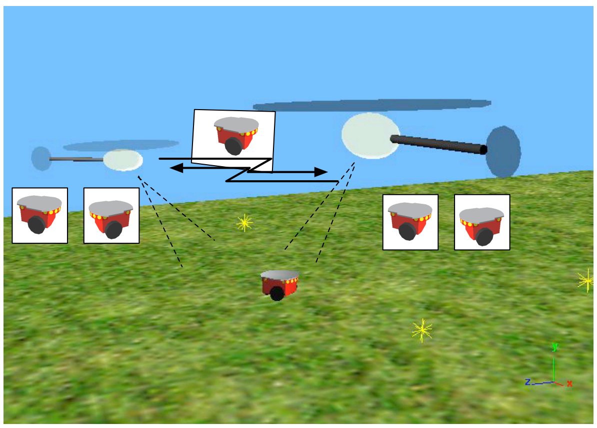

Fig. 5. EXPERIMENT 2, PART (A)

information exchange and matching algorithms were performed in real time.

We have tested two different strategies: in part (a) the robots sent to each other the features information for being matched independently by the partner, and in part (b) they exchanged the information about the estimated location. The frequency of updates was the highest possible: as soon as a new image or position was available, it was sent out. This corresponded to a frequency of about 10 frames per second in case (a) and to a frequency of over 100 Hz in case (b).

This experiment has been conducted on realistic simulation software [14].

Part (a)

In part (a), the two UAVs sent to each other the feature points extracted form the image, along with camera orientation for the correct interpretation of the image. In this way each helicopter could search for the target vehicle in the two images taken from different viewpoints, achieving a sort of stereo vision.

An alternative strategy would be to exchange the raw images. Sending the raw image has two main disadvantages: the high bandwidth required to transmit a video stream of acceptable quality, and the double pre-processing of each image, since each vehicle has to process it individually.

Sharing pre-processed information (points of interest) has the advantage that the computation is distributed among the two vehicles (which is one of the aims of a multi-robot system), since each image is pre-processed only once, while the information exchanged it still allows the stereo-like matching of the image features with the estimated pose. Moreover, the needed bandwidth to exchanged the set of points is lower than the one needed for exchanging a full-quality image. The number of points exchanged was of the order of tens. Each UAV was running the *EvoPose* algorithm, with formulation (4) for finding a good matching between the model and the two images. The two terms of the formula were equally weighted, i.e. $W_1 = W_2 = 0.5$.

Table II reports on the results of the experiment, comparing the results obtained in the localization of the ground vehicle by the two UAV in isolation (first two rows) with the results obtained exchanging the image data and running the algorithm to match the candidate pose on both sets (third row).

The experiments confirm that using both cameras makes the algorithm more robust thanks to the redundant information, and allow achieving a more precise localization with respect to using only one. Furthermore, the time needed to compute a good pose for the first time is lower.

This approach has the further advantage that, since the two robots use the same information and run the same algorithm, the computed localization will be the same.

Part (b)

In part (b) of the experiment, each UAV computed independently the pose of the target vehicle and sent it to the team-mate, that incorporated it as a new individual in its population. This approach requires low bandwidth (only 6×4 bytes are sent, assuming the size of a floating point number is 4) and lower processing, since the distance function is computed only on one set of feature points. However, the precision on the localization does not take full advantage of the stereo matching, and the computed localization can be different for the two robots, which is an undesirable situation, since it causes incoherence in the localization knowledge of the team members. As far as attitude is concerned, the differences reported in the simulations were negligible. Concerning the estimated position the differences were higher: for instance, in experiment 2(b), helicopter 1 estimates the target is almost one meter far from the position estimated by helicopter 2. The fourth and fifth rows of Table II report on the results of experiment 2(b). Again, the experiments confirm that exchanging the information coming from two sources makes the localization more precise w.r.t the localization

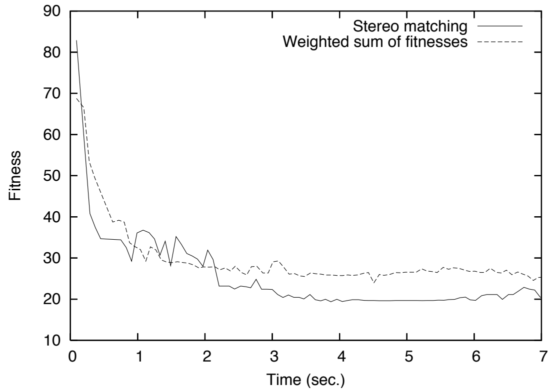

Fig. 6. FITNESS OF BEST INDIVIDUAL: COMPARISONS FOR STRATEGY (A)

computed by each robot in isolation. In particular, helicopter 2 improves notably its estimation, thanks to the information received from its team-mate.

In conclusion, the first approach resulted to be the most effective: its computational cost does not increase significantly with respect to the stand alone application by the two vehicles, and the bandwidth required is low (order of hundred of bytes). Additionally, it guarantees that the two vehicles have the same estimated position.

C. Fitness Comparison

To conclude this section, let us analyze the behavior of the search process in experiment 2, part (a). Fig. 6 compares the fitness of the best solution found by the algorithm used in isolation on each image, and used to find a matching in both images simultaneously. In this case, the fitness of the stereo matching is lower than the weighted sum of the fitness obtained computing each term in isolation. In other words, each term of equation (4) reaches lower values when used in combination with the other one, meaning that a better solution is found thanks to the synergy with the other term. A similar behavior has been observed for strategy (b) when the information (hints) from the team-mate is used.

V. CONCLUSION AND FUTURE WORK

We have presented two examples of vision-based multi-robot cooperative localization, in which the position of a team-mate or of a target vehicle is computed by the joint efforts of two members of the team. The cooperation is achieved by generalizing an evolutionary-based vision algorithm in order to allow it to incorporate sensor information coming from external sources and to search in multiple images in a stereo-like fashion.

Experiments have confirmed the effectiveness of the proposed approach, showing that it can be used for the purpose of mixing information coming from multiple sources, and that the behavior of the algorithm and the quality of the solutions found are improved by such mixing.

An important feature of the extensions of the algorithm proposed, is that it is not limited to two vehicles, but can be applied to any number of vehicles, the only limitation being of computational and communication bandwidth nature, allowing the joint localization of a target by several vehicles.

The focus of this work was on testing the possibility of using the evolutionary algorithm for the purpose of multi-vehicle cooperative localization. A number of issues remain open, and will be object of future work.

EvoPose is a matching algorithm, and as such its main limitation relies in the dependence on a feature extraction process. The study of the impact of the feature-detection algorithm and of camera calibration on the quality of the points detected and fed to the *EvoPose* algorithm, and consequently on the performances of the matching process, was not in the aims of this paper. Also, what happens when the object looked for is not in the field of view, and how to detect false positives are questions that need to be answered prior to implantation in robots for in-field application.

The computation of the weighting terms in the case of the stereo matching extension will also be object of study, since it can incorporate valuable heuristic information such as confidence in the data and/or in the quality of the hardware producing it, quality of the point of view, processing resources of the sender etc.

REFERENCES

[1] Cao, Y. U., Fukunaga, A. S. & Kahng, A. (1997). Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4(1):7–27.

[2] Rossi, C., Abderrahim, M., and Diaz, J. (2005). An evolutionary algorithm for model-based object pose estimation and tracking. In *Proc. IEE Int. Conf. on Visual Information Engineering*, pages 78-83.

[3] Wunsch, P., Winkler, S., Hirzinger, G. (1997). Real-time pose estimation of 3-D objects from camera images using neural networks. In *Proc. ICRA 97*, pages 3232-3237.

[4] Lowe, D. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450.

[5] Pope, A.R., Lowe, D.G. (2000). Probabilistic models of appearance for 3-D object recognition. *Int. Journ. of Computer Vision* 40(2): 149167.

[6] Davids, L., DeMenthon, D., Duraiswani, R., and Samet, H. (2002). SoftPOSIT: Simultaneous pose and correspondence determination. In *Proc. 7th ECCV*, volume III, pages 698–703.

[7] O. Faugeras (1993). *Three-Dimensional Computer Vision - A Geometric Viewpoint*. MIT Press.

[8] Horaud, R., Dornaika, F., Lamiroy, B., and Christy, S. (1997). Object pose: the link between weak perspective, paraperspective, and full perspective. *Int. Journal of Computer Vision*, 22(2):173–189.

[9] D. Fox, W. Burgard, H. Kruppa, S. Thrun (2000). A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots* 8:3, pages 325-344.

[10] R. Madhavan, K. Fregene, L. Parker (2004). Distributed Cooperative Outdoor Multirobot Localization and Mapping. *Autonomous Robots* 17:, pp. 2339.

[11] J. Gu, M. Meng, A. Cook, P.X. Liu (2002). Sensor Fusion in Mobile Robot: Some Perspectives In *Proc. ICRA 2002*, pages 11941199.

[12] D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York, 1989.

[13] Jianbo Shi and Carlo Tomasi (1994). Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*.

[14] O. Michel (2007). Webots: Professional Mobile Robot Simulation. *J.Adv. Robotics Systems*,1(1):39-42.