2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Self-localization of Anonymous Mobile Robots from Aerial Images

Olivier Poulet[1] and François Guérin[2] and Frédéric Guinand[1]

Abstract— **This paper presents three methods for anonymous mobile robot localization within a global frame. An aerial camera takes, at regular time intervals, pictures of the area in which robots are moving. The camera determines the coordinates of each robot. Each robot receives the whole set of coordinates extracted from each picture. Mobile Robots are all identical, they do not have any identifier and they can neither communicate with each other nor they can detect themselves. The first localization method is based on the analysis of the angular variation between two images. The second method relies on the analysis of the distances stemmed from three successive pictures. The last one determines if there exists an orientation allowing a specific robot to travel the path between two successive positions. A simulation plateform using augmented reality and the multi-robot software Player-Stage are presented. This plateform is used for validating the different localization methods. Tests and results are presented and compared.**

I. Introduction

Within the mobile robotics domain, the generic term of localization may have different meanings according to the context.

Individual localization concerns the ability of a robot to localize itself alone. Several methods are able to perform individual localization. The *dead reckoning* method [1] allows a robot, knowing its initial position and its own movements, thanks to proprioceptive sensors, to compute its new relative position at any moment. However, the lack of accuracy limits the applicability of this method. Another method, called *global localization* and described in [2], relies on landmarks distributed in the environment of the robot. Using lidars and 3D cameras, the robot is able to localize itself based on the landmarks positions. The last individual localization method is known as the *kidnapping* method that allows a robot to relocalize itself after beeing moved from a known position to an unknown one [3].

In a multi-robots setting the localization problem presents another difficulty since each robot has to identify itself on the picture. This problem increases in difficulty when different robots in the scene perform close and symmetric movements. Identification is however necessary for many applications like exploration [4] or transportation [5].

The methods of cooperative localization in a global frame let robots adjust their own position from the knowledge of relative positions and directions of the other robots of the group. In [6], the authors use a Monte-Carlo method that provides an accuracy much better than global localization methods for one robot. Mutual Localization (ML) is used for localizing neighbor robots with respect to fixed or mobile landmarks attached to the robots [7].

Very few works address the ML problem in the context of anonymous robots. To the best of our knowledge, the researches of Franchi et al. [7] are among the most advanced ones. Robots are assumed to be identical, they cannot be distinguished by their shape or any other characteristics. Each robot considered in [7] is equipped with some material allowing the self-estimation of its pose (position and orientation) in its own fixed frame. In addition, each robot has the ability to detect other robots located in its neighborhood and to communicate with them. In that context, the authors propose a two-stages method for solving the problem. In the first phase, each robot sends to its neighbors its index, its estimated pose and the relative pose measures of its neighbors (without associating the measures with indices) produced by its own detector, and receives similar information from the neighbors. These information are used for computing all possible relative positions of the robots. Then, during the second phase the best possible solution is estimated thanks to an extended Kalman filter. The algorithm takes into account false positive, i.e. objects looking like robots but that are not robots, and false negative corresponding to non detected robots because of obstacles that may hide them. Anonymity entails some ambiguous solutions for estimating the pose of other robots, especially when their configuration has some symmetric properties. For addressing these issues, Franchi et al. proposed a new method by substituing the Kalman filter by a particle filter [8].

Cognetti et al. have extended the results for the problem of anonymous localization in three dimensions in [9]. Finally in [10], the authors describe algorithms for anonymous localization for different scenarii: in two and three dimensions with linear and angular speeds measures, and in three dimensions with speed-up and angular speed measures. The initial algorithms, based on geometric computations, gather the data of each robot and data received from other robots for providing different possible figures.

In all these works, communications between robots are anonymous but use indices. The use of indices aims at avoiding an over-representation of data. Indeed, if the same data is taken into account many times, the process might converge towards a wrong position as it was outlined in [11].

The problem addressed in the current work is the global localization of anonymous and identical robots without communication between them. It is assumed that some pictures

[1] Olivier Poulet and Frédéric Guinand are with Normandie Univ. UNIHAVRE, LITIS Laboratory, 76600 Le Havre (France) pouletolivier@yahoo.fr, frederic.guinand@univ-lehavre.fr

[2] François Guérin is with Normandie Univ. UNIHAVRE, GREAH Laboratory, 76600 Le Havre (France) francois.guerin@univ-lehavre.fr

978-3-9524-2699-9 ©2018 EUCA

are taken, from a camera, at regular time intervals and that the coordinates, in the camera frame, of all the robots can be extracted from the pictures. Another assumption is that all the robots are on each picture. The set of anonymous coordinates is sent to all the robots. Using these anonymous sets illustrated in Figure 1, each robot has to determine its own position, without communicating with the other robots. The localization is made in a global frame, the one of the camera. Considering our set of assumptions, in particular the anonymity and the absence of communication between robots, the current work is, to the best of our knowledge, the first one addressing this problem.

In Section II, three localization methods addressing the problem are described. The first method compares angular variations of robots between consecutive pictures, while linear speed variations are at the center of the second method. The last method looks for possible orientations for the robots based on the set of coordinates received from each picture. Experiments and results will be presented in Section III and will be followed by the conclusion.

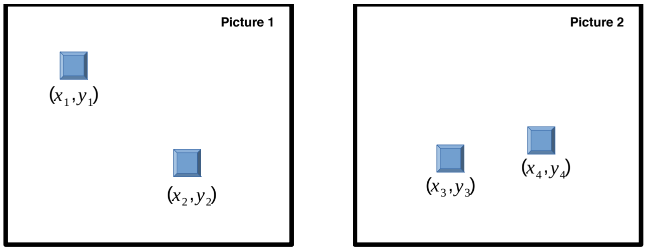

Fig. 1. The two robots receive, at regular interval, the coordinates of all the robots on the picture. Robots have to determine who they are.

II. Localization Methods

A. Angular Variation

This method relies on the orientation of robots with respect to the frame of the camera. Upon reception of picture i, each robot records all the angles ($\theta_i(p)$ within this picture between the orientation of the robot (p) and the frame of the camera as illustred on Figure 8. Between the reception of two consecutive pictures, each robot performs a rotation r (r is a random rotation speed, $r.\Delta t < 2\pi$ and Δt is the time between two pictures). Upon reception of the new picture it computes the new angles ($\theta_j(p)$) for comparing them with the previous ones ($\theta(p) = \theta_j(p) - \theta_i(p)$) . The difference between measured angles on the pictures and the estimated rotation angles gives some clues for answering the question: which robot am i on the picture? We assume the linear speed of robots equal to zero between two pictures. To determine its self-localization, a robot needs to compare its own rotation, obtained from the odometry, with the angles provided by the camera ($|r\Delta t - (\theta(p)|$). The smallest difference defines the position of the robot on the pictures. (see algorithm 1)

B. Linear Speed Variation

In the pinhole model [13], the camera has its own Cartesian coordinate system in three dimensions (X, Y, Z). The origin moves on a plane and the projection of the image plan is denoted by M and m as illustrated on Figure 2.

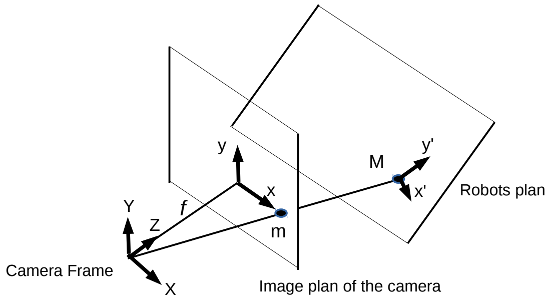

Fig. 2. The considered pinhole model.

Algorithm 1 analysis of the angular variation

```
$min \leftarrow 2\pi$
$numPicture \leftarrow 1$
while reception of a picture do
    if $numPicture = 1$ then
        for p=1:n do
            $\theta_1(p) \leftarrow \theta(p)$
        end for
        $r \leftarrow$ random value ($r.\Delta_t < 2\pi$)
    else
        for p=1:n do
            $\theta_{numPicture}(p) \leftarrow \theta(p)$
            $\Delta\theta(p) \leftarrow |\theta_{numPicture-1}(p) - \theta_{numPicture}(p)|$
        end for
    end if
    for p=1:n do
        if $|r.\Delta_t - \Delta\theta(p)| < min$ then
            $num \leftarrow p$
            $min \leftarrow |r.t_1 - \Delta\theta(p)|$
        end if
    end for
    $numPicture++$
end while
print : I am the robot $num$ on the picture
```

The coordinates of point M in the camera frame are:

$$M = \begin{bmatrix} X_M \\ Y_M \\ Z_M \end{bmatrix} \tag{1}$$

The coordinates of the projection of the robot on image plan are:

$$m = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \lambda \begin{bmatrix} X_M \\ Y_M \\ Z_M \end{bmatrix} \tag{2}$$

with $\lambda = \frac{f}{Z_M}$ where f is the focal length of the camera.

If the robot moves from M to another M' position ($M' = (X'_M, Y'_M, Z'_M)$), then:

$$m' = \begin{bmatrix} x_{m'} \\ y_{m'} \\ z_{m'} \end{bmatrix} = \lambda' \begin{bmatrix} X_{M'} \\ Y_{M'} \\ Z_{M'} \end{bmatrix} \tag{3}$$

with $\lambda' = \frac{f}{Z_{M'}}$. The distance from M to M' is:

$$d_0 = \sqrt{(X_M - X_{M'})^2 + (Y_M - Y_{M'})^2 + (Z_M - Z_{M'})^2} \tag{4}$$

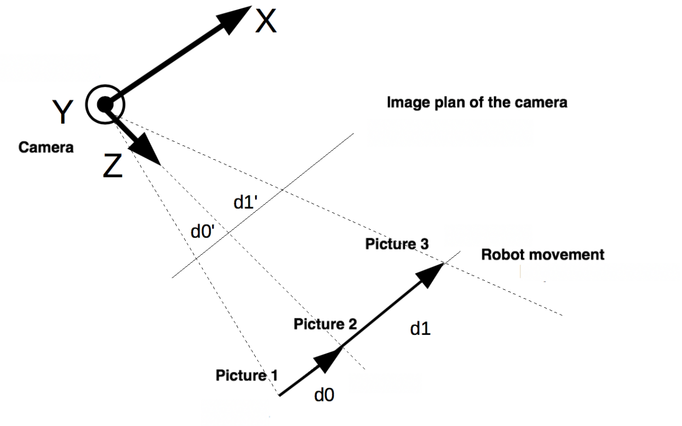

Fig. 3. The moving distances, d_0 and d_1 are projected on the image plan of the camera. d'_0 and d'_1 are measured and recorded. For sake of clarity, we suppose that both plans are parallel, but it has no impact on the method.

and on the image plan we have:

$$d'_0 = \sqrt{(\lambda X_M - \lambda' X_{M'})^2 + (\lambda Y_M - \lambda' Y_{M'})^2 + (\lambda Z_M - \lambda' Z_{M'})^2} \tag{5}$$

If both plans (image plan and robots plan) are parallel, then $d'_0 = \lambda\sqrt{(X_M - X_{M'})^2 + (Y_M - Y_{M'})^2}$ and thus,

$$d'_0 = \sqrt{(x_m - x_{m'})^2 + (y_m - y_{m'})^2} \tag{6}$$

The moving distances between three pictures, corresponding to dates t_0, t_1 and t_2, can be computed according to the Thalès theorem (as illustrated on Figure 3), under the assumption that speeds are linear and constant between two consecutive dates (v_0 between t_0 and t_1 and v_1 for the interval $[t_1, t_2]$):

$$a = \frac{d_0}{d_1} = \frac{d'_0}{d'_1} = \frac{v_0}{v_1} \tag{7}$$

Each time the robot moves a distance d, the camera measures the distance $a \times d$ on the image plan of the camera.

For its own localization, a robot, when receiving a picture, moves linearly at a given speed. Upon reception of another picture, it modifies its speed and moves linearly again until reception of a new picture. By recording its speeds between pictures i and $i+1$ and between pictures $i+1$ and $i+2$, it can compute the value of a. Finally, by comparing the set of ratios stemmed from all the coordinates for all the robots sent by the camera, it can deduce its own position on each picture. The computational complexity depends on the number of robots, for n robots, this complexity, in number of tests, is $O(n^3)$.

C. Localization During a Random Motion

The previous methods are based on the reception of pictures from a camera that thus induces a synchronization of the process. In this third method, the behavior of each robot is not related to the reception frequency of pictures. Robots are characterized by a linear and an angular speeds (v, r). The well known unicycle-like model gives for the robot plan:

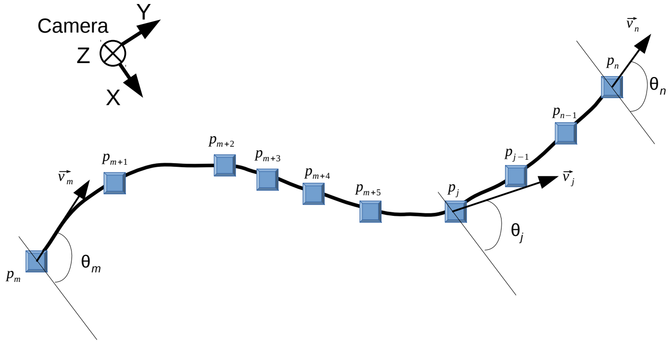

Fig. 4. Some positions of a robot along a trajectory during the time interval $[m, n]$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} v \\ r \end{pmatrix} \tag{8}$$

and for the image plan of the camera:

$$\begin{pmatrix} \frac{1}{\lambda}.\dot{x} \\ \frac{1}{\lambda}.\dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} v \\ r \end{pmatrix} \tag{9}$$

The robot moves along a trajectory as illustrated on Figure 4. The first picture is taken at time m and the position is denoted by p_m. The second picture is taken at time n at position p_n. Both the image plan of the camera and the robots plan are supposed to be parallel (λ is invariant).

If f is a function of classes C^n over I, then, from the Taylor theorem:

$$f(x_0 - h) = f(x_0) - hf'(x_0) + h\epsilon(h) \quad \forall h \in \mathbb{R}$$

with $x_0 \in I$ and such that

$$\lim_{h \to 0} \epsilon(h) = 0$$

By setting $x = t$ and $h = \delta t$, and $x(t_j) = x_j$, $x(t_j - \delta t) = x_{j-1}$, $y(t_j) = y_j$, $y(t_j - \delta t) = y_{j+1}$ and $\theta(t_j) = \theta_j$, $\theta(t_j - \delta t) = \theta_{j-1}$, we obtain at point n:

$$\begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \theta_{n-1} \end{pmatrix} + \delta t \begin{pmatrix} \lambda.\cos(\theta_n) & 0 \\ \lambda.\sin(\theta_n) & 0 \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} v_n \\ r_n \end{pmatrix} \tag{10}$$

By applying the same last formula at x_{n-1}, we obtain:

$$\begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \theta_{n-1} \end{pmatrix} = \begin{pmatrix} x_{n-2} \\ y_{n-2} \\ \theta_{n-2} \end{pmatrix} + \delta t \begin{pmatrix} \lambda.\cos(\theta_{n-1}) & 0 \\ \lambda.\sin(\theta_{n-1}) & 0 \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} v_{n-1} \\ r_{n-1} \end{pmatrix} \tag{11}$$

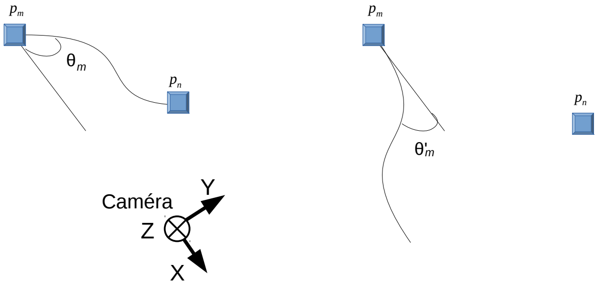

Fig. 5. A robot in its initial and final positions obtained from the two pictures. If its initial angle θ_m is not the good one, starting from the initial position could not lead to the final position (right-hand side of the figure).

By iteration, this leads to:

$$\begin{pmatrix}x_n\\ y_n\\ \theta_n\end{pmatrix} = \begin{pmatrix}x_m\\ y_m\\ \theta_m\end{pmatrix} + \delta t \sum_{j=m+1}^{n} \begin{pmatrix}\lambda.\cos(\theta_j) & 0\\ \lambda.\sin(\theta_j) & 0\\ 0 & 1\end{pmatrix} . \begin{pmatrix}v_j\\ r_j\end{pmatrix} \tag{12}$$

Then, focusing only on the last line:

$$\theta_n = \theta_m + \delta t \sum_{j=m+1}^{n} r_j \quad \text{or} \quad \theta_j = \theta_m + \delta t \sum_{k=m+1}^{j} r_k \tag{13}$$

and by reinjecting this into equation (12), we obtain:

$$\begin{pmatrix}x_n\\ y_n\end{pmatrix} = \begin{pmatrix}x_m\\ y_m\end{pmatrix} + \delta t \sum_{j=m+1}^{n} \begin{pmatrix}\lambda\cos(\theta_m + \delta t \sum\limits_{k=m+1}^{j} r_k)\\ \lambda\sin(\theta_m + \delta t \sum\limits_{k=m+1}^{j} r_k)\end{pmatrix} .v_j \tag{14}$$

If λ is known, only θ_m, the initial angle of the robot on the first picture is unknown. r_j and v_j are the rotation and the linear speed commands, they are known by each robot at any moment. Finding the position of a robot on every picture is equivalent to determine if there exist one and only one angle between two sets of coordinates leading this robot from the position $p_m = (x_m, y_m)$ to $p_n = (x_n, y_n)$ as illustrated on Figure 5.

The first line of the system (14) is:

$$x_n = x_m + \delta t \lambda \sum_{j=m+1}^{n} (\cos(\theta_m + \delta t \sum_{k=m+1}^{j} r_k) v_j) \tag{15}$$

It can be solved in the following way. Let:

$$A = \sum_{j=m+1}^{n} (\cos(\delta t \sum_{k=m+1}^{j} r_k) v_j) \tag{16}$$

$$B = \sum_{j=m+1}^{n} (\sin(\delta t \sum_{k=m+1}^{j} r_k) v_j) \tag{17}$$

$$C = \frac{x_n - x_m}{\lambda \delta t} \tag{18}$$

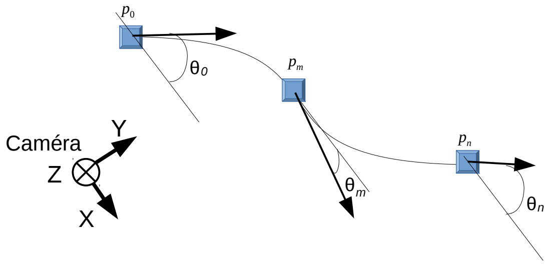

Fig. 6. The different positions of a robot along a trajectory starting at p_0, ending as p_n and crossing p_m

$$\begin{cases} \rho = \sqrt{A^2 + B^2}\\ \alpha = \arctan(\frac{B}{A}) \end{cases} \tag{19}$$

Which leads to:

$$\begin{cases} \text{if } \rho = 0 \text{ there is no solution}\\ \text{if } \mid \frac{C}{\rho} \mid > 1 \text{ there is no solution}\\ \text{else } \theta_m = \arccos(\frac{C}{\rho}) - \alpha\\ \theta_m \in [0; \pi] \end{cases} \tag{20}$$

The second line of the system (14) is:

$$y_n = y_m + \delta t \lambda \sum_{j=m+1}^{n} (\sin(\theta_m + \delta t \sum_{k=m+1}^{j} r_k) v_j) \tag{21}$$

That can be solved in the following way. Let $D = \frac{y_n - y_m}{\lambda \delta t}$, the equation to be solved is:

$$\begin{cases} \text{if } \rho = 0 \text{ there is no solution}\\ \text{if } \mid \frac{D}{\rho} \mid > 1 \text{ there is no solution}\\ \text{else } \theta_m = \arcsin(\frac{D}{\rho}) - \alpha\\ \theta_m \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right] \end{cases} \tag{22}$$

The following cases have to be considered:

$$\begin{cases} \cos(-\theta_m) = \cos(\theta_m)\\ \sin(\pi - \theta_m) = \sin(\theta_m) \end{cases} \tag{23}$$

When a robot has to localize itself, if λ is known, it compares pairs of positions on the two pictures (one point of the first picture and the second point on the second picture), and attempts to compute a unique initial angle compatible for a motion from $p_m = (x_m, y_m)$ to $p_n = (x_n, y_n)$. If such an angle could be determined then the coordinates correspond to the positions of the robot of both images.

If λ is unknwon, three pictures are needed. In that case, the angle θ_m of the middle picture is common to both trajectories: from p_0 to p_m and from p_m to p_n as illustrated on Figure 6. In that situation, the system to be solved is:

$$\begin{pmatrix}x_m\\ y_m\end{pmatrix} = \begin{pmatrix}x_0\\ y_0\end{pmatrix} + \delta t \sum_{j=1}^{m} \begin{pmatrix}\lambda.\cos(\theta_m - \delta t \sum_{k=j+1}^{m} r_k)\\ \lambda.\sin(\theta_m - \delta t \sum_{k=j+1}^{m} r_k)\end{pmatrix} .v_j \tag{24}$$

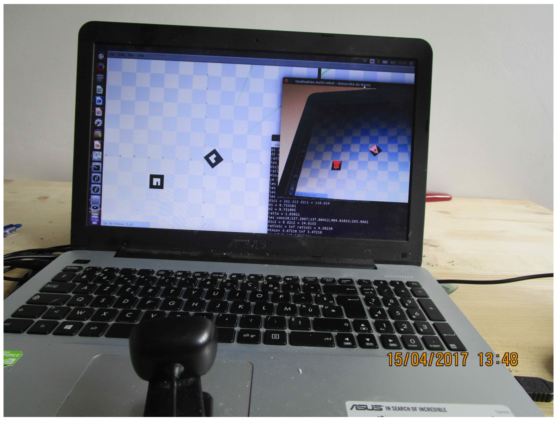

Fig. 7. Picture of the experimental platform. In the foreground of the picture, the webcam takes pictures of the screen of the computer on which the simulation is ran. On the screen of the computer, on the left part, a view of the Player-Stage simulator running our experiment (with two robots). On the right part, a window contains the image produced by Artoolkit with the detected patterns corresponding to the robots.

$$\begin{pmatrix}x_n\\ y_n\end{pmatrix} = \begin{pmatrix}x_m\\ y_m\end{pmatrix} + \delta t \sum_{j=m+1}^{n} \begin{pmatrix}\lambda.\cos(\theta_m + \delta t\sum_{k=m+1}^{j} r_k)\\ \lambda.\sin(\theta_m + \delta t\sum_{k=m+1}^{j} r_k)\end{pmatrix} .v_j$$

The system has two unknown: λ and θ_m. There are n^3 possibilities for n robots which entails an increase in the computational complexity.

For the experiments, λ is supposed to be known.

III. Experiments and Results

Experiments have been led on the multi-robot Player Stage simulator [14]. Robots in the simulator are modeled as simple squares. The simulator runs the movements of the robots that are captured by a webcam taking pictures of the screen of the computer on which the simulation is running, as it is illustrated by a screenshot of the experimental setup on Figure 7.

The coordinates of the robots have been determined using the Artoolkit Software Development Kit for Augmented Reality. This SDK [12], is able to detect some predefined patterns on a picture and to extract from these patterns a set of coordinates as illustrated on Figure 8.

The Artoolkit software running within Processing 3 detects the robots (basic square patterns) and computes the coordinates of specific points inside these squares (Figure 8). These coordinates are stored in a file which is sent to all the robots. Each of them, upon reception of the file, performs the previously described processing in order to localize itself.

The general principle of the experimental platform is described in Figure 9.

Tests have been realized with two robots. A new picture has been taken every 10 seconds. The camera was positioned approximately at 30cm of the screen. The size of the robots was about 7mm × 7mm on the screen. About the parallelism of the image plan of the camera and the plan of the robots, a linear constant speed movement has been simulated for a robot between two consecutive pictures. λ was obtained as the ratio between the known distance performed by the robot and the distance measured on the image plan of the camera. This value should have been the same everywhere on the screen, but the parallelism was experimentally not easy to guarantee and was not perfect. Thus, during our experiments, the λ values obtained for different points of the plan of the robots varied between 20.8 and 22.07. An average value was considered for performing the computations. It is worth mentioning that this approximation is a potential source of localization errors.

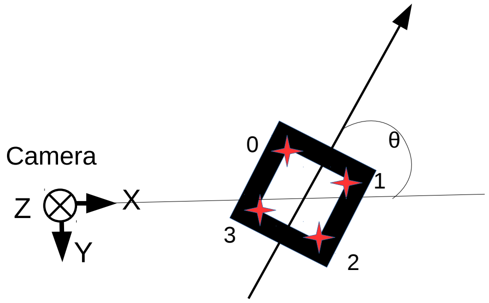

Fig. 8. The orientation of the robot is obtained from the pattern sticked on it by applying suitable methods included in the Artoolkit SDK.

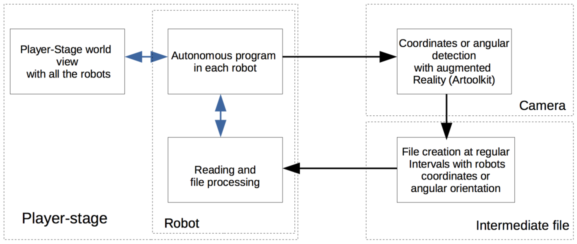
Player-Stage world view with all the robots
Autonomous program in each robot
Coordinates or angular detection with augmented Reality (Artoolkit)
Camera
Reading and file processing
File creation at regular Intervals with robots coordinates or angular orientation
Player-stage
Robot
Intermediate file

Fig. 9. Principle of the experimental platform.

For the test of angular variation, rotations of robots were randomly generated, started at the same moment, limited to 9 degrees/second and rotating counterclockwise.

For the test of localization by linear speed variation, the robots started at the same time using random linear speeds limited to 0.1 unit/second, one unit being defined within the Player-Stage simulator.

For the test of localization during a movement, the discretization time was $\delta t = 0.01$ second. Random trajectories have been programmed on each robot. The linear speed and rotation variations were triggered at random moment (between 0 and 3 seconds each) and bounded by 0.1 and 0.5 unit/second for the linear speed and between -20 and 20 degrees/second for the rotating speed. If a robot met an obstacle (wall, other robot), the movement was stopped and the robot rotated on itself at a random rotating speed.

Results are reported in the table below.

The high success rate for the localization by angular variation could be explained by the limited number of tests that have to be performed for the computation coupled with the efficiency of the Augmented Reality approach. The detection is rather simple and is not error-prone due to the efficiency of the augmented reality method since robots are not moving. However the method relies on the knowledge of the orientation angle, which was made possible by Artoolkit, and this is its main weakness, indeed, if the orientation is not detected the method fails.

The success rate of the method based on the linear speed variation is not as good as for the previous method, however, it remains at a good level. The number of tests needed to be done for the localization is n^3, where n is the number of robots.

Finally the last method is more error-prone, however it should be noticed that we made the experiments in the worst conditions by considering similar speeds for the robots that started almost at the same moment. The success rate is however comparable to the results obtained by the previous method for a computational complexity also comparable.

localization	angular	linear	movement
Number of runs	65	100	118
Success rate (%)	96.9	84	82.2
Average	1.6°(*)	0.823(**)	7.22°(***)
Standard Deviation	1.1°(*)	2.93(**)	7.3°(***)

(*) average deviation between measured angles and real ones, (**) average deviation between ratio measured and real ones, (***) average angle between x and y axes.

IV. Conclusion and Future Works

The aim of this work was to study the problem of the localization of anonymous and identical mobile robots in a multi-robots settings. Robots were not allowed to communicate with each other. Based on the pictures taken from above, for instance a drone taking pictures of several ground robots moving in a specific area, each robot should be able to identify itself in the scene. For that purpose we have proposed three different methods relying on the variation of linear or rotating speeds of the robot. The methods were tested experimentally through simulation using Player Stage simulator, the Artoolkit SDK for the detection of patterns for extracting the set of coordinates of the robots as well as their orientation angles, and Processing 3 for making everything work. The success rate was always greater than 82%, however this should be confirmed by testing the method with a growing number of robots.

In a short-term perspective, the quality of the localization provided by these methods will be evaluated for real experiments (pictures taken from a drone) and for a large number of robots. Then, using a drone for localizing ground robots could be used, but it should include a module for automatically correcting the camera tilt variations (gimbal) through an adaptation of the λ value. Then, a real application will be tested: The crossing of a tunnel (or all aeras without reception) with autonomous vehicles. This one can't know their localization inside the tunnel but can used pictures before and after the crossing to be situated. Then, the problem of the initial conditions must be studied so that the robots cannot generate movements not allowing the distinction between them. Then, a study of the estimation of the probability of sucess must be led to predict the sucess rate according to the context : number of robots, frequency of pictures, discretization time δt...

In a mid-term perspective, the potential developments for this work are manyfold. Localization problems under various situations involving a growing number of robots, or when robots enter into/leave the pictures or when the group behaves like a swarm (similar direction and speeds) are still to be investigated.

Aknowledgment

This work has been done in the context of the MADNESS project, co-financed by the European Union with the European regional development fund (ERDF) and Normandie Regional Council.

References

[1] J. Borenstein, H.R. Everett, L. Feng, *"Where am I?" Sensors and methods for mobile robot positioning*, Technical Report, University of Michigan, 1996.

[2] S. Se, D. Low, J. Little, *Local and Global Localization for Mobile Robots using Visual Landmarks*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 414–420, Maui, Hawaii, 2001.

[3] D. Fox, S. Thrun, W. Burgard, F. Dallaert, *Particle filters for mobile robot localization*, A. Doucet, N. de Freitas, and N. Gordon, editors, Sequential Monte Carlo Methods in Practice, pages 401–428. Springer-Verlag, Berlin, 2001.

[4] W. Burgard, M. Moors,D. Fox, R. Simmons, S. Thrun, *Collaborative Multi-Robot Exploration*, Processing of the Internationnal Conference on Robotics and Automation(ICRA), San Francisco, 2000.

[5] G. Habibi ,Z. Kingston, W. Xie,M. Jellins, J. McLurkin, *Distribued Centroid Estimation and Motion Controllers for Collective Transport by Multi-Robot Systems*, Robotics and Automation (ICRA), 2015 IEEE International Conference on.IEEE, May 2015.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, *A probabilistic approach to collaborative multi-robot localization*, Auton. Robots, vol. 8, no. 3,pp. 325-344, 2000.

[7] A. Franchi ,G. Oriolo and P. Stegagno, *Mutual Localization in a Multi-Robot System with Anonymous Relative Position Measures*, IEEE/RSJ Int.Conf. on Intelligent Robots and Systems, St. Louis, MO, pp 3974-3980, 2009.

[8] A. Franchi ,G. Oriolo and P. Stegagno, *Probabilistic Mutual Localization in Multi-Agent Systems from Anonymous Position Measures*, 49th IEEE Conf. on Décision and Control, Atlanta, GA, pp 6534-6540, 2010.

[9] M. Cognetti, P. Stegagno, A. Franchi, G. Oriolo, H.H. Bülthoff, *3-D mutual localization with anonymous bearing measurements*, IEEE Proc. Int. Conf. on Robotics and Automation, 2012.

[10] P. Stegagno,M. Cognetti,G. Oriolo, H.H. Bülthoff, A. Franchi, *Mutual Localization in a Multi-Robot System with Anonymous Relative Position Measures*,IEEE Trans. on Robotics,32(5), pp 1133-1151,2016.

[11] N. Karam, F. Chausse, R. Aufrere, R. Chapuis, *Localization of a group of communicating vehicles by state exchange*,In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 519–524, 2006.

[12] E.H.C. Harik, F. Guérin, F. Guinand, J.F. Brethé, H. Pelvillain, A. Zentout, *Vision Based Target Tracking Using An Unmanned Aerial Vehicle*, IEEE International Workshop on Advanced Robotics and its Social Impacts, ARSO, 2015.

[13] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.

[14] B.P. Gerkey, A. Vaughan, R.T. Herman, A. Howard, *The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems*, Proceedings of the 11th international conference on advanced robotics (pp.317–323). Los Alamitos: IEEE Computer Society Press, 2003.