Mobile robot localization in industrial environments using a ring of cameras and ArUco markers

Sara Roos-Hoefgeest
ISA
Universidad de Oviedo
Gijón, Spain
roossara@uniovi.es

Ignacio Alvarez Garcia
ISA
Universidad de Oviedo
Gijón, Spain
ialvarez@uniovi.es

Rafael C. Gonzalez
ISA
Universidad de Oviedo
Gijón, Spain
rcgonzalez@uniovi.es

Abstract— **Localization of mobile robots in industrial environments is key in an increasingly automated industry. Nowadays, the inspection and repair of heavy steel plates is performed by human workers. Repair work often requires long hours in uncomfortable postures that can cause problems for the worker. We propose a mobile robot placed on top of a steel plate that must move along the plate to inspect and repair it, without leaving the sheet. Robot localization on the plate is key to generate the inspection and repair trajectories.**

There are different methods of localization, the most widely used require the use of expensive laser sensors to create a map using information from the environment and localize from it. This paper proposes a less expensive localization system for a mobile robot based on the installation of ArUco markers in the environment and the use of a ring of 8 calibrated cameras mounted on the robot that allow a 360º vision. This ensures a correct localization regardless of the working area. It is necessary to map the markers with respect to a common coordinate system.

We propose a method to create the map using the ring. We validate the proposal through experiments comparing the localization obtained with the proposed system and a localization using a state-of-the-art SLAM method employing laser sensors.

Keywords— Mobile Robot, Computer Vision, Robot Localization, ArUco Markers, Ring of Cameras, ROS.

I. INTRODUCTION

Mobile robots are appearing in an increasingly automated industry in a growing range of applications. One example is the deployment of autonomous mobile robots for inspection of surface defects. Other applications seek to replace human workers in dangerous tasks, such as grinding.

In the production of heavy steel plates, both problems appear together. Currently, the inspection and repair are performed by human operators. The repair work usually involves long working days in uncomfortable positions that can lead to physical problems for the worker.

This paper presents a localization method for a mobile robot performing sheet metal inspection and repair process without leaving the sheet. Therefore, the robot moves in a localized space, the width of the sheet metal varies from 1.4 to 3.3 m and their length may change from 4 to 18 m. Normally, sheet metal inspections are performed in large indoor environments. So, there may not be enough references in the environment for certain types of localization systems.

Localization problem has been discussed from multiple approaches, being common the use of different sensors that provide direct information about the location of the robot at each time, or about the changes that are produced in its environment.

One of the most widely used alternatives in the localization of mobile robots is the use of Simultaneous Localization and Mapping (SLAM) techniques. These systems build a map of the environment that is used to localize the robot. Expensive laser scanners are generally used for the construction of the environment map. In addition, systems based on natural feature extraction and matching could be used but it can be problematic in certain scenarios if there are not enough natural landmarks.

Sensor fusion is also common in robotics, usually using Kalman filters [1] or particle filters [2], [3], which allows combining estimates from different sources to achieve a more robust pose. In [4], information from inertial sensors is fused with a visual odometry method.

GPS systems are another widely used method. However, systems based on satellite signals are not appropriate for this task because GPS signals are not available in indoor scenarios.

Other strategy is by placing beacons at known positions in the environment to facilitate the localization of the robot. For example, visual beacons, such as luminous beacons of different geometries and colors or fiducial markers.

Different fiducial marker systems can be found, composed by a set of defined markers and an algorithm that allows their detection and identification. These systems have been used in multiple vehicle localization applications. For example, researchers from the AVA group at the University of Cordoba developed different applications to localize mobile robots using ArUco markers with a single camera [5]. Other research using computer vision and artificial markers can be seen in [6], [7] and [8]. The paper in [9] proposes the localization of a mobile robot by a particle filter combining the information from an omnidirectional camera and a range sensor, using fiducial markers.

Other strategy is using a robot tracking system. For that, fiducials are also used, as in [10], in which the mobile robot is tracked by arranging an artificial marker on top of it using a system of multiple cameras fixed in the environment. In [11] they use geometric features of the robot itself instead artificial markers.

In this paper, we propose a localization system using an 8-camera ring fixed in the robot and a set of artificial markers placed in the environment, surrounding the plate being inspected by the robot. It is necessary to know the position of the markers with respect to a common coordinate system. Therefore, a way to locate them and create a map of the markers is proposed, so that the robot can move over the sheet as accurately as possible.

Research supported by: Daorje S.L.U inside project code CN-18-011 and a scholarship under the "Severo Ochoa" program for predoctoral research and teaching with Ref: PA-20-PF-BP19-067 financed by Asturias Regional Government.

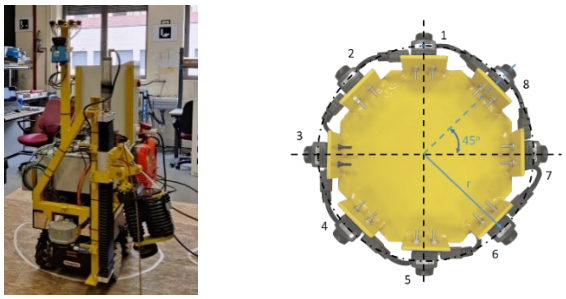
Although we describe the localization for a mobile robot to inspect and repair heavy steel plates, the method is easily adaptable to any other task that require robot localization.

The paper is organized as follows. Section II describe the algorithms used to compute the map of ArUcos and the robot localization. Section III shows results achieved with our prototype at the lab. Finally, section IV presents our conclusions.

II. Methods

A. Hardware

Fig. 1 (a) shows a view of the robot used to solve this problem. The base is an omnidirectional Summit XL Steel robot from Robotnik. On top of the robot, a ring of 8 cameras with the configuration shown in Fig. 1 (b) is installed. The cameras are the UI-1007XS-C model from IDS imaging. Cameras are previously calibrated, both intrinsically and extrinsically, referenced to the robot's reference system.



(a) (b)

Fig. 1. (a) Mobile Robot used in experiments. (b) 8-Camera ring configuration.

The robot is equipped with additional laser sensors to run another type of localization to compare the results obtained in the experiments. This localization is performed with a SLAM algorithm using the *gmapping* ROS package [12]. Markers used are ArUco markers [13] and are fixed in the environment in which the robot moves.

B. Process Workflow

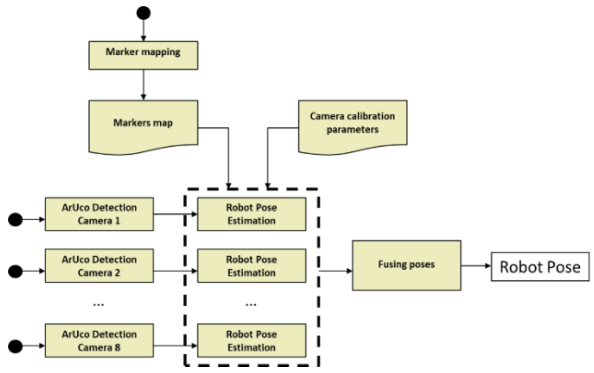


Fig. 2. Process workflow.

A schematic view of the overall process is shown in Fig. 2.

The problem starts with the robot placed on the plate without knowing its location in the environment. To localize it we use the markers placed in the surroundings. For this, it is key to know the poses of the markers relative to a common reference system, so we can refer the robot to a known system. It is also essential to have the 8-camera ring calibrated, both intrinsically and extrinsically, knowing their poses with respect to the robot.

Once we have the markers map, we can localize the robot using the cameras information. Each camera gives an estimation of a robot pose using the markers within its field of view. Finally, pose estimations are fused and a final pose referred to the common reference system is obtained.

C. ArUco markers system

The ArUco marker system [13] consist in a set of markers and an algorithm that allows their detection and identification. An ArUco marker, see Fig. 3, is a square synthetic marker composed of a black border and an internal binary matrix that defines an identifier. The size of the marker determines the size of the internal matrix. There are different ArUco dictionaries that group markers with different identifiers.

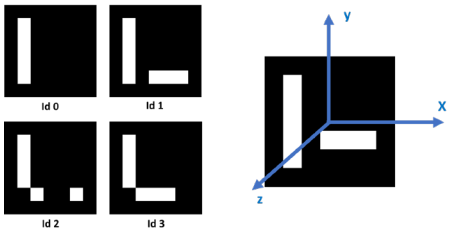


Fig. 3. ArUco Markers. In blue, system of coordinates of ArUco markers.

The detection of the markers in the images is performed by the *aruco_detect* package [14]. This package provides the 4 corners and the identifier of each marker present in the image. Once the markers have been detected in the images, the pose of the markers with respect to the camera is computed.

A pin-hole model is assumed to model the camera. To estimate the pose of the marker with respect to the camera that detects it, the Perspective-n-Point (PnP) problem is solved.

Solving the Perspective-n-Point (PnP) problem is possible to estimate the pose of a calibrated camera given a series of 3D points in the world and their corresponding 2D projections in the image. A single marker yields the four correspondences needed, from its corners, to solve the problem. The camera pose consists of 6 degrees of freedom, orientation (roll, pitch, yaw) and translation with respect to the world (X, Y, Z).

After solving the PnP problem given the 3D positions of the corners in the marker system itself and their corresponding 2D projections on the image, the calibrated camera pose with respect to the marker system is obtained.

The coordinate system associated with each marker can be seen in Fig. 3, in which the Z axis will be perpendicular to the plane containing the marker and the origin located at the center of the marker.

D. Aruco mapping

To localize the robot using markers placed in the environment, it is necessary to know their poses relative to a common reference system. To map the markers, the robot will be placed in the environment in such a way that all markers are visible with any of the cameras in the ring. The map stores the poses of the ArUco markers referred to a common reference system, which will be called the world system, fixed to one of the markers.

The developed mapping algorithm can be divided into different steps. First, markers will be detected in the images and one of them will be chosen to define the world coordinate system. Subsequently, the poses of the rest of the markers with respect to the reference system will be estimated.

To ensure that the map is created with the smallest possible error, the marker that will define the position and orientation of the global coordinate system will be the one that is closer to the camera and whose Z-axis is as parallel as possible to the camera axis. Once the first marker is located and the world system is set, the poses of the rest of the markers will be referred to the global coordinate system.

As previously explained, from the images and the calibration parameters of the cameras, the transformation between the camera and the markers that appear in the images is obtained. By having all the cameras extrinsically calibrated, referred to the robot's reference system, the transformations between cameras are also known. From this data, the necessary transformations are applied to refer the poses of the markers to the common reference system.

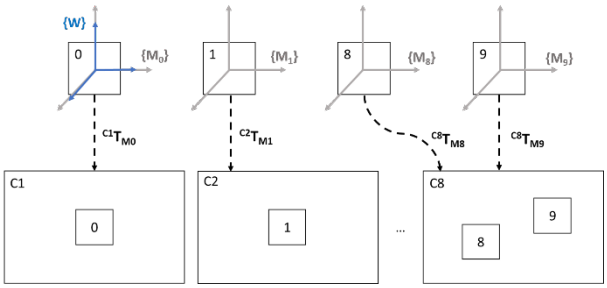


Fig. 4. Schematic mapping system transformations. Where {M0}, {M1}, {M8}, {M9} are the coordinate systems of the markers in 3D space and C1, C2 y C8 are the cameras that take the images of the markers. In this case, world reference frame is set on marker with id 0.

An example is shown in Fig. 4. C1, C2 and C8 refer to the cameras that see the markers. First, the world reference frame {W} is set to the marker with id 0 {M_0}. The calculation of the pose of markers 1, 8 and 9 is performed by applying the appropriate transformations. In the case of marker 1:

$$ {}^{W}T_{M1} = {}^{W}T_{M0} \cdot {}^{M0}T_{C1} \cdot {}^{C1}T_{C2} \cdot {}^{C2}T_{M1} \quad (1) $$

All the transforms are known. ${}^{W}T_{M0}$ is the transform defined between the system of the desired world {W} and that of the selected marker {M_0}. In this case, it only involves a rotation. ${}^{M0}T_{C1}$ and ${}^{C2}T_{M1}$ are obtained as explained in C and ${}^{C1}T_{C2}$ are the transformation between cameras 1 and 2. The same process is repeated for all of the markers. In this example:

$$ \begin{aligned} {}^{W}T_{M8} &= {}^{W}T_{M0} \cdot {}^{M0}T_{C1} \cdot {}^{C1}T_{C8} \cdot {}^{C8}T_{M8} \\ {}^{W}T_{M9} &= {}^{W}T_{M0} \cdot {}^{M0}T_{C1} \cdot {}^{C1}T_{C8} \cdot {}^{C8}T_{M9} \end{aligned} \quad (2) $$

E. Robot localization

We performed different steps to locate the robot, as can be seen in Algorithm 1. First, we estimate a pose of the robot according to the information provided by each of the cameras, resulting in eight different poses. Next step merges the eight pose estimations. Finally, a linear Kalman filter is applied to optimize the robot pose.

First, the markers present in the images captured by the cameras at each instant are detected. Then, a robot's pose is estimated from those markers present in each of the cameras.

In this case, following the strategy explained in section C, the PnP problem is solved with the projections of the corners of the markers and the 3D positions of the corresponding markers with respect to the world system, obtained from the previously built map, see Fig. 5.

Algorithm 1: Robot localization

Step 1. Detect ArUco markers in the images of the cameras.
Step 2. Estimate one possible robot pose from the markers seen by each camera
Step 3. Fuse robot poses
Step 4. Filter measurement according to the previous ones and the motion command executed by the robot.
Step 5. Apply Linear Kalman Filter to optimize the final robot pose

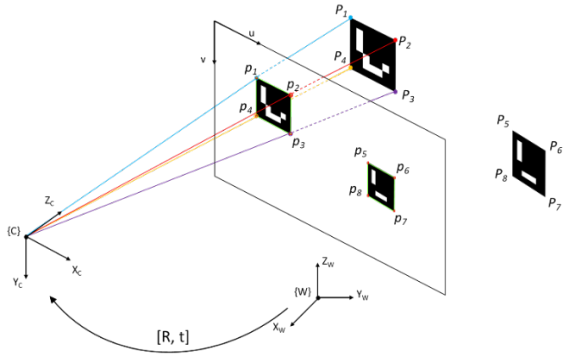


Fig. 5. Camera pose estimation from 3D marker corners (Pi) in the world system and their projections on the image (pi).

After solving the problem, the transformation between the camera coordinate system and the world system is obtained, ${}^{C}T_{W}$. In this way, the transformation matrix of the robot and the world system, ${}^{W}T_{R}$, can be estimated directly as:

$$ {}^{W}T_{R} = {}^{W}T_{C} \cdot {}^{C}T_{R} \quad (3) $$

Where ${}^{W}T_{C}$ is the inverse transformation of ${}^{C}T_{W}$, and ${}^{C}T_{R}$ is the transformation that relates the camera and the robot coordinate systems. For each of the cameras, an estimation of the robot pose is obtained based on the markers present in each camera. In order to choose the best location, an estimation of the accuracy of the measurement is made. It is based in the reprojection error, the distance of the marker from the camera and the relative rotation between them.

The reprojection error is used to discard the estimation if it is higher than a chosen value. Then, the distance and orientation of the marker to the camera are weighted according to experimentally defined thresholds in the laboratory. Finally, the overall accuracy (Ac_i) is the mean of these two. For example, the distance from the marker to the camera, at a distance greater than a threshold of 5 meters (TH_d) it is given a weight of 0. At a closer distance, is done as in (4). The same is done for relative orientation between camera and marker Z-axes with a TH_{or} of 60º. The orientations on the other axis are not considered because we placed the markers in such a way that they are not rotated on these axes.

$$Ac_d = \frac{TH_d - dist}{TH_d}$$
$$Ac_{or} = \frac{TH_{or} - ang}{TH_{or}} \quad (4)$$
$$Ac_i = \frac{Ac_d + Ac_{or}}{2}$$

Thus, the estimates associated with the smallest error among the eight cameras at each instant is selected. If there are multiple pose estimations with a small error and the Euclidean distance between them is less than a specified gap, the final position of the robot will be the weighted average based on the estimate of the accuracy of the position estimation, see (5).

$$P = \sum_{i=0}^{n} P_i \cdot \frac{Ac_i}{\sum_{j=0}^{n} Ac_j} \quad (5)$$

Then, the given pose is filtered to eliminate outliers. This allows to discard a measurement if it differs significantly from the previous ones or has a value that is not consistent with the motion command executed by the robot. After passing this first outlier elimination step, the estimate is incorporated into a Linear Kalman Filter to make the final localization more robust.

1) Linear Kalman Filter

The Kalman filter [15] is a recursive prediction filter that uses state space techniques to estimate the state of a dynamic system. In this paper, we use a linear Kalman filter to discard bad estimations and achieve a more reliable localization. The Kalman filter is divided into two main stages, prediction and correction. In the first step, the dynamic model predicts the state of the system. In the second step, the prediction is corrected with the observation model. This procedure is repeated for each time interval.

The successive states $s_t \in R^n$ of a controlled process are related to a dynamic model describing the transform of the state vector in time, shown in (6).

$$s_t = As_{t-1} + w_t \quad (6)$$

Where A is the state matrix and w_t is the process noise. The state vector composing the matrix will be defined with a total of 9 states (equation (7)): The position (x,y), with its first and second derivatives, i.e., velocity and acceleration. The rotation information in the axis Z (ϕ), with their first and second derivatives, corresponding to angular velocity and acceleration. We only consider the localization on the (x,y), since it is a mobile robot moving on the XY plane. Also, only the rotation on the Z-axis of the robot is considered.

$$X = (x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}, \phi, \dot{\phi}, \ddot{\phi}) \quad (7)$$

The measurement of the robot position at each instant is related to the state s_t by the linear observation model of equation (8), where v_t represents the measurement noise and C the observation matrix.

$$z_t = Cs_t + v_t \quad (8)$$

The measurement vector consists of 3 measurements, corresponding to the position (x,y) and orientation, in Z axis (ϕ), of the robot.

As previously mentioned, the first stage of the Kalman filter is the prediction. At each time a first estimation of the current state is performed, called a priori state s_t^- calculated by ignoring the dynamic noise and solving the equations describing the dynamic model, equation (9). Its covariance matrix S_t^- is calculated during this stage according to equation (10), where S_{t-1} is the covariance of the a posteriori estimation error at the previous instant and Λ_w is the covariance of the process noise that measures the quality of the motion model with respect to reality.

$$s_t^- = As_{t-1} \quad (9)$$

$$S_t^- = AS_{t-1}A^T + \Lambda_w \quad (10)$$

Subsequently, occurs the correction stage. In this stage, the prediction s_t^- is improved with the observations made at instant t, (z_t), obtaining the a posteriori estimate s_t according to equation (11). Where the difference $(z_t - z_t^-)$ is the residual measurement that reflects the difference between the predicted measurement $z_t^- = Cs_t^-$ and the actual mean z_t.

In this equation, the estimated state and measurements are weighted to calculate the corrected state according to G_t, the filter gain matrix. G_t is calculated according to equation (13), where Λ_v is the covariance matrix of the measurements.

Its covariance matrix S_t is calculated according to the error propagation law, equation (12).

$$s_t = s_t^- + G_t(z_t - z_t^-) \quad (11)$$

$$S_t = S_t^- - G_t C S_t^- \quad (12)$$

$$G_t = S_t^- C^T (C S_t^- C^T + \Lambda_v)^{-1} \quad (13)$$

III. Experiments and Results

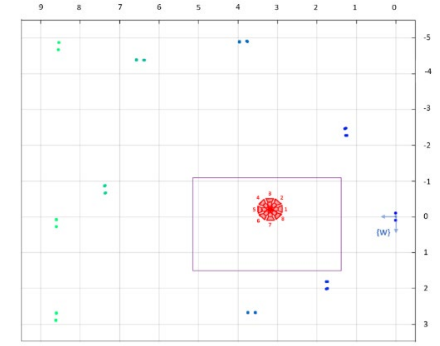
The robot described in section II.A has been used to test our algorithms. The experiments were performed in a laboratory of approximately 11x9m, with 10 ArUco markers placed in the environment, see Fig. 6.

The software has been developed for ROS-Kinetic running on a Ubuntu 16.04 LTS operating system. We used the open source OpenCV library to process the images and the *aruco_detect* package for the markers detection. Also, the *gmapping* package was used, which applies a SLAM algorithm to know the location of the robot in the environment using laser sensors attached to the robot. In order to estimate the performance of the localization obtained by the proposed system based on the ring of cameras and ArUco markers.

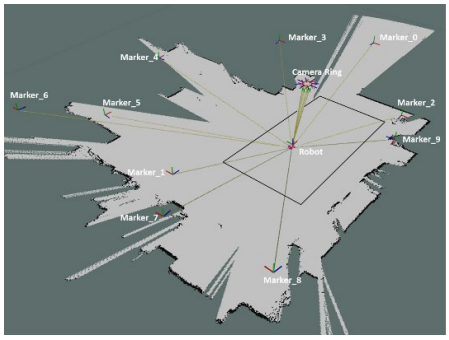
Fig. 6. Laboratory where the experiments were performed. ArUco markers are fixed in the environment.

A. *Aruco mapping*

First, it is necessary to make a map of the markers placed in the environment. The results obtained in the generation of the map can be seen in Fig. 7. We can assume that the map has been generated correctly by comparing the real environment with its visualization in RViz.



(a)



(b)

Fig. 7. Environment with the mapped markers, the camera ring and the plate on which the robot moves. (a) Zenithal representation (b) Rviz view.

B. *Robot localization*

To evaluate the proposed localization method, 15 different positions were marked in the environment. We compare the robot localization obtained with the proposed method and the localization provided by the gmapping package.

The results do not consider the localization on the Z-axis, since it is a mobile robot moving on the XY plane. Also, only the rotation on the Z-axis of the robot is considered.

Fig. 8 shows the positioning results. To verify the accuracy of the localization, the robot is moved to certain points on the sheet. In this case, 15 positions have been considered. The localization calculated by the SLAM package and the localization obtained according to the implemented method are compared.

TABLE I. shows the absolute error between the location of the SLAM and that of the implemented method. Both in the X and Y coordinates as well as the Euclidean error. Maximum Euclidean error reached was 17.27 centimeters. However, the usual errors are much lower, with a median error of 3.10 cm.

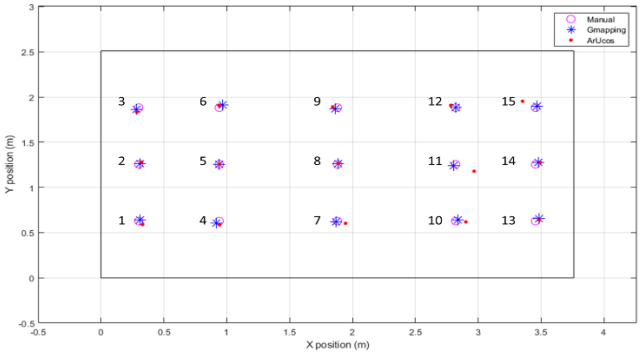


Fig. 8. Locations according to the SLAM method of gmapping, the proposed method based on a ring of cameras and ArUco markers and the manually measured positions. Black rectangle represents the plate on which the robot moves, with dimensions 3.67x2.51m.

TABLE I. ERRORS IN POSITION: GMAPPING AGAINST PROPOSED SYSTEM

1			2			3		
X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)
2.015	4.969	5.363	1.255	1.480	1.941	0.499	3.132	3.172
4			**5**			**6**		
X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)
2.521	1.812	3.104	0.302	0.221	0.374	2.870	0.609	2.932
7			**8**			**9**		
X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)
7.425	1.556	7.587	0.257	0.652	0.701	2.01	2.061	2.881
10			**11**			**12**		
X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)
6.611	2.452	7.051	16.135	6.152	17.268	3.899	2.478	4.619
13			**14**			**15**		
X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)	X error (cm)	Y error (cm)	Euclidean error(cm)
0.144	1.468	1.475	1.346	0.461	1.423	11.463	5.827	12.859

Fig. 9 shows a comparative between the final position of the robot after fusing the intermediate estimates given by each of the cameras. We can see that the final estimate is correct even though some of the cameras give estimates with a high error. The ring of cameras makes it possible to locate the robot accurately, as it allows a good view of the markers at all times, even if it is in different cameras each time.

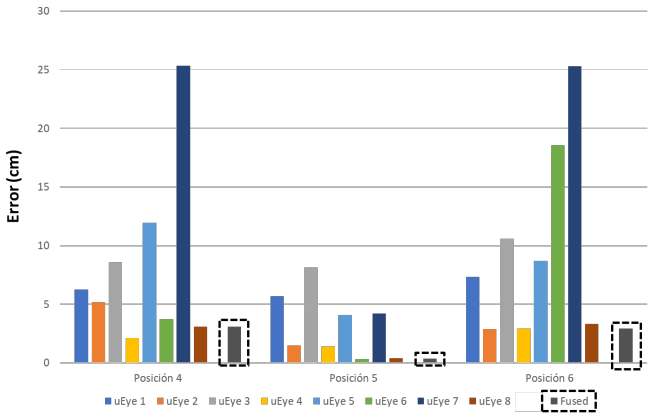


Fig. 9. Comparison between the final position of the robot and the intermediate estimates of the 8 cameras. Data from 3 of the previously fixed positions for the robot that can be seen in Fig. 8.

An experiment was also performed to test the localization of the robot's orientation. As it is a robot that moves along the plane XY, we are only interested in the rotation about the Z axis of the robot. For this purpose, the robot was positioned at a fixed point and 8 different orientations were evaluated. In the same way as the position, the estimated orientation has been checked against the orientation of the gmapping SLAM algorithm.

Fig. 10 shows the different orientations the robot is placed, rotating around itself. Also, TABLE II. shows the errors between the orientation estimated using the method proposed in this paper and the orientation provided by gmapping package. Mean error is 1.901º and the maximum error is 4.501º.

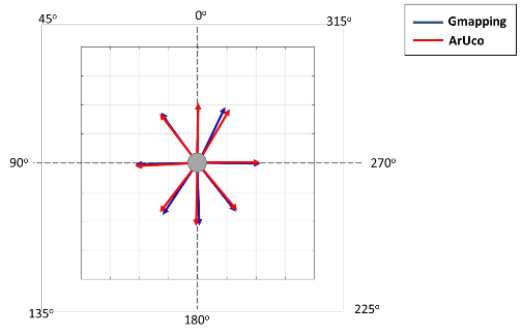


Fig. 10. 8 different orientations of the robot rotating about its Z axis. In blue, the orientation provided by gmapping. In red, the one estimated by the proposed system.

TABLE II. ERRORS IN ORIENTATION: GMAPPING AGAINST PROPOSED SYSTEM

Posición	1	2	3	4	5	6	7	8
Error(°)	0.076	-4.501	0.490	1.359	-3.435	-3.169	1.157	1.028

IV. CONCLUSION

We have proposed a system for mobile robot localization using an 8-camera ring and ArUco markers fixed in the environment. Also, we develop a mapping algorithm for localization the markers with respect to a common reference system. The experiments show very satisfactory results that validate the proposed algorithms.

In this paper we prove that the proposed localization system gives equally good results as a slam algorithm using expensive sensors. Moreover, the environment in which the experiments were conducted is a laser-friendly environment, as it is a rather enclosed space with many saliency points. The proposed fiducial-based method works in any environment. For example, in big industrial spaces as is the case in our final application.

Localization by fiducials has certain limitations. Accuracy is highly dependent on the size of the markers and their position and orientation relative to the camera. By having a ring of cameras, we have a complete 360º view of the environment. This allows the robot to be located accurately, as we ensure that we have a good view of some markers at all times.

An accurate marker map is also key to the robot's localization. To this end, we are working on optimizing it by applying a bundle adjustment. We are also looking to add constraints on the relative positions between the ring cameras, on the one hand, and the four corners of each marker, on the other.

In future, we will study the fusion of fiducial localization with information from other sources, such as an inertial measurement unit (IMU) or movement commands from the navigation system. For example, using an extended Kalman filter, which allows the fusion of information from different sources to estimate the pose of the robot with a lower error.

V. REFERENCES

[1] N. Z. Mohd Nasir, M. A. Zakaria, S. Razali, and M. Y. bin Abu, 'Autonomous mobile robot localization using Kalman filter', *MATEC Web Conf.*, vol. 90, p. 01069, 2017, doi: 10.1051/matecconf/20179001069.

[2] H. Lang, T. Li, G. Villarrubia, S. Sun, and J. Bajo, 'An Adaptive Particle Filter for Indoor Robot Localization', in *Ambient Intelligence - Software and Applications*, vol. 376, A. Mohamed, P. Novais, A. Pereira, G. Villarrubia González, and A. Fernández-Caballero, Eds. Cham: Springer International Publishing, 2015, pp. 45–55. doi: 10.1007/978-3-319-19695-4_5.

[3] A. Burguera, Y. Gonzlez, and G. Oliver, 'Mobile Robot Localization using Particle Filters and Sonar Sensors', in *Advances in Sonar Technology*, S. Rui, Ed. I-Tech Education and Publishing, 2009. doi: 10.5772/39415.

[4] M. Alatise and G. Hancke, 'Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter', *Sensors*, vol. 17, no. 10, p. 2164, Sep. 2017, doi: 10.3390/s17102164.

[5] R. Muñoz-Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, 'Mapping and localization from planar markers', *Pattern Recognition*, vol. 73, pp. 158–171, Jan. 2018, doi: 10.1016/j.patcog.2017.08.010.

[6] A. Babinec, L. Jurišica, P. Hubinský, and F. Duchoň, 'Visual Localization of Mobile Robot Using Artificial Markers', *Procedia Engineering*, vol. 96, pp. 1–9, 2014, doi: 10.1016/j.proeng.2014.12.091.

[7] A. Mutka, D. Miklic, I. Draganjac, and S. Bogdan, 'A low cost vision based localization system using fiducial markers', *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 9528–9533, 2008, doi: 10.3182/20080706-5-KR-1001.01611.

[8] L. E. Ortiz-Fernandez, E. V. Cabrera-Avila, B. M. F. da Silva, and L. M. G. Gonçalves, 'Smart Artificial Markers for Accurate Visual Mapping and Localization', *Sensors*, vol. 21, no. 2, p. 625, Jan. 2021, doi: 10.3390/s21020625.

[9] P. Javierre, B. P. Alvarado, and P. de la Puente, 'Particle Filter Localization Using Visual Markers Based Omnidirectional Vision and a Laser Sensor', in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, Feb. 2019, pp. 246–249. doi: 10.1109/IRC.2019.00045.

[10] K.-T. Song and Y. C. Chang, 'Design and Implementation of a Pose Estimation System Based on Visual Fiducial Features and Multiple Cameras', in *2018 International Automatic Control Conference (CACS)*, Taoyuan, Nov. 2018, pp. 1–6. doi: 10.1109/CACS.2018.8606773.

[11] D. Pizarro, M. Mazo, E. Santiso, M. Marron, and I. Fernandez, 'Localization and Geometric Reconstruction of Mobile Robots Using a Camera Ring', *IEEE Trans. Instrum. Meas.*, vol. 58, no. 8, pp. 2396–2409, Aug. 2009, doi: 10.1109/TIM.2009.2016380.

[12] G. Grisettiyz, C. Stachniss, and W. Burgard, 'Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling', in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005, pp. 2432–2437.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, 'Automatic generation and detection of highly reliable fiducial markers under occlusion', *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014, doi: 10.1016/j.patcog.2014.01.005.

[14] 'aruco_detect - ROS Wiki'. http://wiki.ros.org/aruco_detect (accessed Nov. 27, 2018).

[15] G. Welch and G. Bishop, 'An Introduction to the Kalman Filter', Los Ángeles, California, Estados Unidos, 2001, vol. Course 8, p. 81.