ICRA 2005 Workshop on Cooperative Robotics, Barcelona, Spain, April 22, 2005.

Fusing vision-based bearing measurements and motion to localize pairs of robots

Luis Montesano[1] José Gaspar[2] José Santos-Victor[2] Luis Montano[1]
[1]I3A, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza
Zaragoza, Spain
{montesano,montano}@unizar.es
[2]Instituto de Sistemas e Robótica,
Instituto Superior Tecnico
Lisboa, Portugal
{jag,jasv}@isr.ist.utl.pt

***Abstract*— This paper presents a method to cooperatively localize pairs of robots fusing bearing-only information provided by a camera and the motion of the vehicles. The algorithm uses the robots as landmarks to estimate the relative location between the platforms. Bearings are obtained directly from the camera, as opposed to measuring depths which would require knowledge or reconstruction of the world structure.**

We present the general recursive Bayes estimator and three different implementations based on an extended Kalman filter, a particle filter and a combination of both techniques. We have compared the performance of the different implementations using real data acquired with two platforms, one equipped with an omnidirectional camera, and simulated data.

Keywords: Cooperative robots, bearing-only measurements, localization.

I. Introduction

In the last years cooperative robotics has received considerable attention. Teams of robots are able to overcome the limitations of single robots and allow to attack more difficult tasks. Furthermore, they increase the degree of autonomy and robustness by introducing redundancy. However, the use of teams of robots increases the complexity of the system. New challenges appear providing new research areas. In this context cooperative localization is considered one of the basic capabilities required for autonomous operation of teams of robots.

In this paper we address the problem of cooperatively localizing two robots using bearing-only measurements and the motion of the robots. When the robots navigate based on proprioceptive sensors, e.g. odometry, they build and maintain their own, unrelated, referential frames. By providing exteroceptive sensors, such as omnidirectional cameras that track other robots, the various referential frames can be fused to a single one. The common referential frame opens the way for cooperation and sharing of the information acquired by each of the robots.

Omnidirectional cameras allow localizing robots in a 360^o azimuthal field-of-view and extracting their bearing (azimuthal) locations. Bearings are obtained directly from omnidirectional cameras as raw data, without requiring any smoothness or rigidity assumptions about the scene-objects. Hence, extracting bearings is a simple task as compared to e.g. vision-based SLAM ([3], [25]), stereo matching and reconstruction ([4], [28]), baseline detection and estimation ([12]), ground-plane based navigation, or other depth reconstruction techniques. In order to account for the lack of depth and orientation measurements, we propose to use the motion of the vehicles (odometry readings or motion commands) together with the bearing observations. Combining both types of information allows us to estimate the initial relative location of the platforms and keep track of it.

We have implemented our localization algorithm within a Bayesian framework. We have evaluated three implementations of the classical Bayes filter: an extended Kalman filter, a particle filter and a combination of both techniques. We have performed several tests in our laboratory to validate the method. The results suggest that we can fuse the motion of the platforms with the bearing measurements to estimate their relative location.

This paper is organized as follows. Next Section presents the related work. Section III describes the problem. In Section IV and V we derive the Bayesian estimator for our problem and we describe our implementation. The experimental results are presented in Section VI.

II. Related work

In a computer vision framework the relative localization of two robots corresponds to an object-pose estimation problem. Seminal work on object-pose estimation assumed known geometrical object-structures useful both for object detection and tracking [14], [8], [13]. In order to deal with the uncertainty and noise in the measurements, the approaches for tracking and estimating the pose of the objects were based on stochastic models and algorithms, e.g. Extended Kalman Filter.

Many research works conducted in the last years show that the assumptions of rigidity and knowledge of the object-structure can be relaxed in tracking applications. For example Murase [17] proposed appearance based models, i.e. models based on images, to represent and recognize objects at various poses and illumination conditions. Blake [2] proposed active contours to track objects with time varying silhouettes based on the condensation algorithm. Recently, Okuma [20] proposed a boosted particle filter for tracking and detection of hockey players based on the color histograms associated to their uniforms.

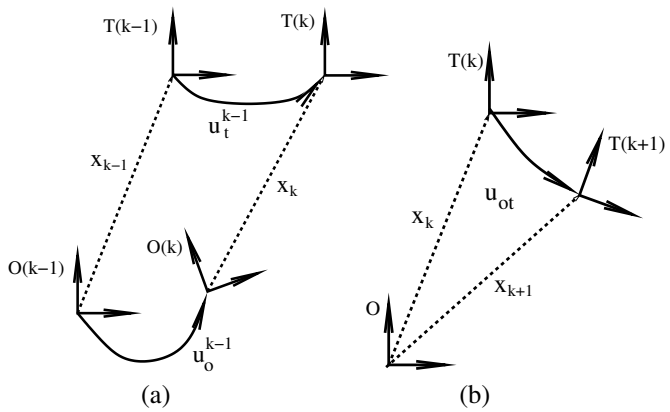


Fig. 1. Interpretation of the relative location between to moving platforms. (a) world fixed reference system (b) robocentric approach.

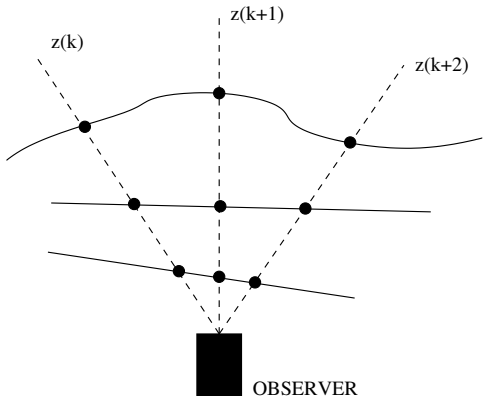


Fig. 2. Unobservable target trajectory: infinite trajectories generate the same bearing measurements, $z(.)$. The figure shows three of these trajectories.

The most recent references show that an object can be tracked, without knowing its structure, using other object-features such as image corners, contours, patterns, color patches, etc. A number of the tracking methods do not provide pose estimation, due to the fact that feature models do not contain structure information. However they obtain the location of the object within the image. In the present work, we consider extracting bearing information from the location of one robot in the image acquired by another one and combining it with the motion of both platforms to estimate their relative localization.

On the other hand, cooperative localization has been an active research area during the last years in the robotic community. There exist different approaches depending on several aspects as the information used (map availability, static/dynamic features) or the multi-robot architecture (communication capabilities, centralized/decentralized). Related to this paper several authors have used the robots as landmarks ([24], [10], [16]). In the absence of static references these methods estimate the relative pose among the robots instead of their location on a global reference system. However, all these algorithms use range-bearing sensors that provide an accurate estimation of their relative positions.

Some authors have used bearing-only measurements to estimate the location of the robots or the objects around them [19], [23], [26], [22]. The proposed algorithms usually use probabilistic techniques to fuse the information provided at the same time by different sources with known uncertain locations. To increase the robustness of an static object tracker Huster [11] fuses bearing information from a monocular camera and inertial rate sensors. In our case we want to compute the relative location of two moving platforms. Using only bearing information of the position of the robots makes the problem non observable and we need to incorporate some extra information, the motion of the platforms, to turn it solvable.

Finally, bearing-only information has also been studied in other tracking contexts. For instance, Bar-Shalom in [1] shows how to apply the Extended Kalman Filter to estimate the location of a constant velocity moving target and how the trajectories of the observer improves the resulting estimation. Bearing-only measurements have also been used to show improvements on particle filters for uniform straight motion of the target [18]. Finally, Trawny [27] shows how different motions result in different localization accuracies and optimizes the motion of the members of the team to improve the localization capabilities.

III. Problem description

We are interested on estimating the relative location $\mathbf{x}$ of two moving platforms. As we do not have any static reference we use a robocentric approach [10]. We choose one of the platforms as the reference system, the observer, and compute the relative pose of the other one, the target, with respect to it. In this reference system the observer remains static while the target motion is the composition of the motion of both platforms. Figure 1(a) depicts the relations between the locations of two moving platforms. The observer and target motion is $\mathbf{u}_o$ and $\mathbf{u}_t$ respectively. When the reference system moves with the observer (Figure 1(b)) the final location of the target is the result of the combination of the movements of both platforms $\mathbf{u}_{ot}$.

According to the previous description our problem is equivalent to estimating the position of a moving target from a stationary platform. Bearing-only measurements of a moving target do not contain enough information to estimate its position [1]. Let $\mathbf{p}_t = (x_t, y_t)$ be the position of the target and $\mathbf{p}_o = (x_o, y_o)$ the position of the observer. We will denote $\mathbf{p}_o(k)$ to the position of the target at time step k. The observations are bearings with respect to a reference direction of the target at different points in time,

$$z^k = \tan^{-1} \frac{y_o(k) - y_t(k)}{x_o(k) - x_t(k)} + w^k$$

where w^k is the measurement noise at time k. Due to the relative motion of the target, which is a combination of its own motion and that of the observer, there exist infinite possible locations of the platform satisfying the measurement equation. Figure 2 illustrates some of the possible trajectories for a set of three observations.

One, therefore, needs to incorporate some more information to compute the relative pose of the platforms. If one knows the relative motion of the target, its location can be estimated using three measurements and triangulation techniques. The relative motion restricts the set of possible trajectories to a single one. In order to incorporate the motion of the target one must estimate not only its position but the whole location including the orientation $\mathbf{x}_t = (x_t, y_t, \theta_t)$.

For an stationary observer there is a degenerated case when the target moves on a straight line toward the observer. In the case of two moving robots there must be some relative displacement between the platforms. If the vehicles remain static, just rotate, move parallel one to the other or move along the line joining them, the bearing measurements do not change. Moreover, depending on the motion of the platforms the amount of information provided by the observations differs ([1], [21], [27]).

Finally, the method has to deal with the uncertainties and noises in the system. As pointed out before, the sensor measurements provide an initial relative location of the robots with a high uncertainty in the range and orientation of the target platform. Therefore, one has to be able to cope with a big initial uncertainty. In the next Section we present a Bayesian estimation technique to estimate the relative pose of two moving vehicles using bearing measurements and the trajectories of both vehicles.

IV. Relative pose estimation

We assume that the robots are able to measure their own displacement. Let $\mathbf{u}^{k-1}$ be the motion of the robot between time $k-1$ and k. The observer robot is equipped with sensors that provide bearing measurements z^k of the position of the other robot at time k. We assume that the robots are synchronized and model the synchronization error as another source of noise in the system.

Let r_o be the observer. We will denote z_o^k the observations and $\mathbf{u}_o^k$ the odometry readings obtained by r_o at time k. $Z_o^k = \{z_o^1, ..., z_o^k\}$ and $U_o^k = \{\mathbf{u}_o^0, ..., \mathbf{u}_o^k\}$ represent the observations and odometry readings of the observer up to time k. The target motion measurements up to time k are $U_t^k = \{\mathbf{u}_t^0, ..., \mathbf{u}_t^k\}$. Our objective is to estimate the relative location $\mathbf{x} = (x\ y\ \theta)$ of robot r_t, the target, with respect to robot r_o based on the information available up to time k. Thus, the posterior distribution we are estimating is $p(\mathbf{x}_k|Z_o^k, U_o^{k-1}, U_t^{k-1})$ and the corresponding recursive Bayes filter is,

$$p(\mathbf{x}_k|Z_o^k, U_o^{k-1}, U_t^{k-1}) \propto p(z_o^k|\mathbf{x}_k) \; p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{Z}_o^{k-1}, \mathbf{U}_o^{k-1}, \mathbf{U}_t^{k-1}) \tag{1}$$

where $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{Z}_o^{k-1}, \mathbf{U}_o^{k-1}, \mathbf{U}_t^{k-1})$ is obtained from the posterior of the previous step and the motions given by $\mathbf{u_o^{k-1}}$ and $\mathbf{u_t^{k-1}}$,

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, Z_o^{k-1}, U_o^{k-1}, U_t^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_o^{k-1}, \mathbf{u}_t^{k-1}) p(\mathbf{x}_{k-1}|Z_o^{k-1}, U_o^{k-2}, U_t^{k-2}) d\mathbf{x_{k-1}} \tag{2}$$

The previous formulation assumes independent Markov processes for both the observation z_o^k and the evolution of $\mathbf{x}$. Equation 1 estimates the posterior at time k using the posterior computed at time $k-1$, a motion model $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_o^{k-1}, \mathbf{u}_t^{k-1})$ and a measurement model $p(z_o^k|\mathbf{x}_k)$.

V. Implementations of the Bayes Estimator

In this section we provide three different implementations for equation (1): the classical Extended Kalman Filter (EKF), a sampled based approach known as particle filters (PF) and a combination of both techniques (PF-EKF). The EKF assumes Gaussian distributions and noises and requires to linearize the process and measurement models. The lack of an accurate initial relative location makes the EKF unappropriate for the first stages of the estimation. In other words the EKF may diverge due to the initial uncertainty and the linearization error if the measurements do not contain enough information.

On the other hand particle filters do not require any of the previous assumptions and are able to represent any kind of distribution given a sufficient number of particles. However, they are computationally more expensive and introduce a discretization error which depends on the number of particles used. Therefore, we propose an hybrid approach (PF-EKF) that uses a particle filter in the first steps and switches to an EKF when the distribution of the particles is close to a Gaussian.

To describe the motion of the platforms we use two operators over reference systems: the composition $\oplus$ and the inversion $\ominus$ of locations,

$$\mathbf{x}_1 \oplus \mathbf{x}_2 = \begin{pmatrix} \cos\theta_1 x_2 - \sin\theta_1 y_2 + x_1 \\ \sin\theta_1 x_2 + \cos\theta_1 y_2 + y_1 \\ \theta_1 + \theta_2 \end{pmatrix} \tag{3}$$

$$\ominus\mathbf{x}_1 = \begin{pmatrix} -\cos\theta_1 x_1 - \sin\theta_1 y_1 \\ \sin\theta_1 x_1 - \cos\theta_1 y_1 \\ -\theta_1 \end{pmatrix} \tag{4}$$

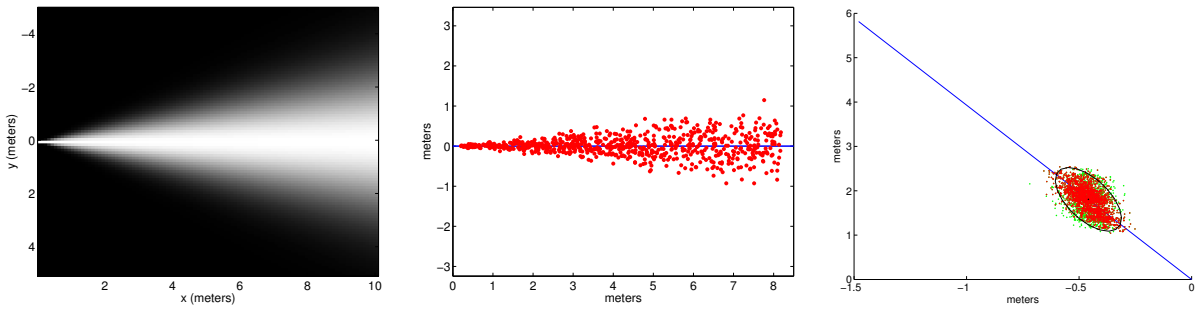
Using these operators the movement of the target platform seen from the observer is a combination of the movements of each platform(see Figure 1(a)),

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_o^{k-1}, \mathbf{u}_t^{k-1}) = \ominus\mathbf{u}_o^{k-1} \oplus \mathbf{x}_{k-1} \oplus \mathbf{u}_t^{k-1}$$

where $\mathbf{u}_t = (dx_t, dy_t, d\theta_t)$ is the motion of the target platform and $\mathbf{u}_o = (dx_o, dy_o, d\theta_o)$ is the motion of the observer platform. Note that $\mathbf{u}_t^k$ and $\mathbf{u}_o^k$ are noisy measurements of the true displacement of the robot corrupted with noises $\mathbf{v}_t^k$ and $\mathbf{v}_o^k$ respectively. On the other hand the observation model is, as presented in Section III,

$$z_o^k = h(\mathbf{x}) + w^k = \tan^{-1}\frac{y(k)}{x(k)} + w^k \tag{5}$$

where w^k is the measurement noise.



(a) (b) (c)

Fig. 3. (a) observation model for the particle filter implementation, (b) initial distribution of particles, (c) the distribution after a given number of steps is close to a Gaussian

A. Extended Kalman filter implementation

In this section we implement Equation (1) using an EKF. The EKF framework models all the random variables as Gaussians and approximates the nonlinear process and measurement equations with the partial derivatives of the nonlinear functions. We denote $\hat{\mathbf{x}}$ to the current estimate of the state vector $\mathbf{x}$ and $\mathbf{P}$ to the associated covariance matrix. The odometry and bearing measurements are also represented by their realizations, $\hat{\mathbf{u}}_t$, $\hat{\mathbf{u}}_o$, $\hat{z}_o$. The measurements are corrupted by independent white Gaussian noises with covariance matrices $\mathbf{V_t}$, $\mathbf{V_o}$ and $\mathbf{R}$ respectively. We use a first order approximation for the process and measurement models,

$$\begin{aligned} f(\mathbf{x}, \mathbf{u}_o, \mathbf{u}_t) &\approx f(\hat{\mathbf{x}}, \hat{\mathbf{u}}_o, \hat{\mathbf{u}}_t) + \nabla_f \begin{pmatrix} \mathbf{x} - \hat{\mathbf{x}} \\ \mathbf{u}_o - \hat{\mathbf{u}}_o \\ \mathbf{u}_t - \hat{\mathbf{u}}_t \end{pmatrix} \\ h(\mathbf{x}) &\approx h(\hat{\mathbf{x}}) + \nabla_h (\mathbf{x} - \hat{\mathbf{x}}) \end{aligned}$$

where ∇_f is the Jacobian of the function f with respect to the state vector $\mathbf{x}$ and the odometry readings $\mathbf{u}_o$ and $\mathbf{u}_t$ and ∇_h is the Jacobian of the function vector h with respect to the state vector $\mathbf{x}$ evaluated at the current state $\hat{\mathbf{x}}_k$ and motions $\hat{\mathbf{u}}_o$ and $\hat{\mathbf{u}}_t$.

Using the previous Jacobians we provide next the resulting Kalman filter equations for time k. The predicted relative location $\hat{\mathbf{x}}_{k|k-1}$ using the linearized process model and its associated covariance matrix $\mathbf{P}_{k|k-1}$ are,

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_o^{k-1}, \hat{\mathbf{u}}_t^{k-1}) \\ \mathbf{P}_{k|k-1} &= \nabla_f \begin{pmatrix} \mathbf{P}_{k-1} & 0 & 0 \\ 0 & \mathbf{V_{o_k}} & 0 \\ 0 & 0 & \mathbf{V_{t_k}} \end{pmatrix} \nabla_f^T \end{aligned}$$

with ∇_f evaluated at $\hat{\mathbf{x}}_{k-1}$, $\hat{\mathbf{u}}_\mathbf{o}^{\mathbf{k-1}}$ and $\hat{\mathbf{u}}_\mathbf{t}^{\mathbf{k-1}}$. The corresponding update step is,

$$\mathbf{S}_k = \nabla_h \mathbf{P}_{k|k-1} \nabla_h^T + \mathbf{R}, \quad \mathbf{W}_k = \mathbf{P}_{k|k-1} \nabla_h^T \mathbf{S}_k^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{W}_k v_k, \quad \mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^T$$

where $v_k = \hat{z}_o^k - h(\hat{\mathbf{x}}_{k|k-1})$ is the innovation and the Jacobian ∇_h is evaluated at $\hat{\mathbf{x}}_{k|k-1}$.

B. Particle filter implementation

Particle filters are sequential Monte-Carlo techniques to estimate posterior distributions [5]. They represent the distributions by a set of M samples $S = \{\mathbf{x}_k^{[1]}, ..., \mathbf{x}_k^{[M]}\}$. The usual way to implement the recursive Bayes estimator of Equation (1) is to use the posterior obtained in the previous step $k-1$ and the motion model to guess the distribution at time k. For each sample $\mathbf{x}_{k-1}^{[i]}$ a new sample $\mathbf{x}_k^{[i]}$ is generated from the possible locations described by $\mathbf{u}_o^{k-1}$ and $\mathbf{u}_t^{k-1}$,

$$sample\ \mathbf{x}_k^{[i]}\ from\ p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_o^{k-1}, \mathbf{u}_t^{k-1}) \tag{6}$$

The set of all these samples conforms what is known as the proposal distribution. For each robot we use a motion model similar to the one described in the Carnegie Mellon Navigation toolkit [15]. Note that we are composing two uncertain motions. The number of particles must be sufficient to sample all the possible locations induced by both motions requiring more particles than a single one. The samples from the proposal are distributed according to $p(\mathbf{x}_k \mid Z_o^{k-1}, U_o^{k-1}, U_t^{k-1})$ and do not include the information of the last observation z_o^k. To take into account the difference between the proposal distribution and the target distribution $p(\mathbf{x}_k | Z_o^k, U_o^{k-1}, U_t^{k-1})$ the samples are weighed according to their likelihood,

$$\alpha^{[i]} = \frac{target}{proposal} \propto p(z_o^k | \mathbf{x}_k^{[i]}) \tag{7}$$

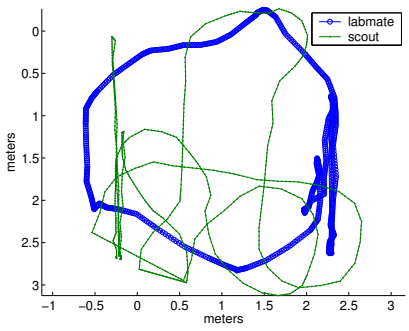
The new set of particles approximating $p(\mathbf{x}_k | Z_o^k, U_o^{k-1}, U_t^{k-1})$ is then generated by sampling from the proposal distribution according to the importance factors $\alpha^{[i]}$.

The measurement model is the likelihood of the observation given the current state. We model the noise of each bearing as an independent zero mean Gaussian distribution with a standard deviation σ_w,

$$p(z_o^k | \mathbf{x}_k^{[i]}) = N(z_o; h(\mathbf{x}_k^{[i]}), \sigma_w)$$

Figure 3(a) shows the likelihood model for a bearing measurement corresponding to the x axis. The darker the location is, the lower the likelihood associated to it.

(a)



(b) (c) (d)

Fig. 4. (a) Image acquired by the Labmate. (b) Labmate and Scout robots. (c) Ground truth: transformation of a perspective to an orthographic view of the ground plane; the chess patterns indicate the points used for calculating the dewarping homography. (d) Ground truth trajectories computed from the external camera.

C. Combining both approaches

In order to benefit from the advantages of each method we propose to use a combination of both techniques. During the first steps of the estimation process a Gaussian distribution does not approximate well the uncertainty. As we do not have any prior information about the relative location of the platforms, we use the measurements to initialize the filters. According to the measurement equation (5) we do not have any information about the distance between the platforms or their relative orientation. Therefore we use a particle filter to sample all the possible initial locations (figure 3(b)).

After a certain number of steps the particles will usually converge to an unique mode corresponding to the correct relative pose. We periodically try to fit a Gaussian to the current particle filter. When the particles are close to a Gaussian we switch to an EKF to track it (Figure 3(c)). In order to recover from potential failures, the normalized innovation squared test [1] is used. If the test fails, a new particle filter is initialized using the last measurements.

This strategy combines the computational efficiency of the Kalman filter with the more powerful representation capabilities of the particle filters needed in the initial steps of the algorithm due to the use of bearing-only measurements. The complexity of the EKF is $\mathcal{O}(m^2)$ where $m = dim(\mathbf{x})$. On the other hand the complexity of the PF is $\mathcal{O}(N)$ with N the number of particles. As m remains constant and is much smaller than the required number of particles N, the EKF implementation is computationally much more efficient for our problem.

VI. Experiments

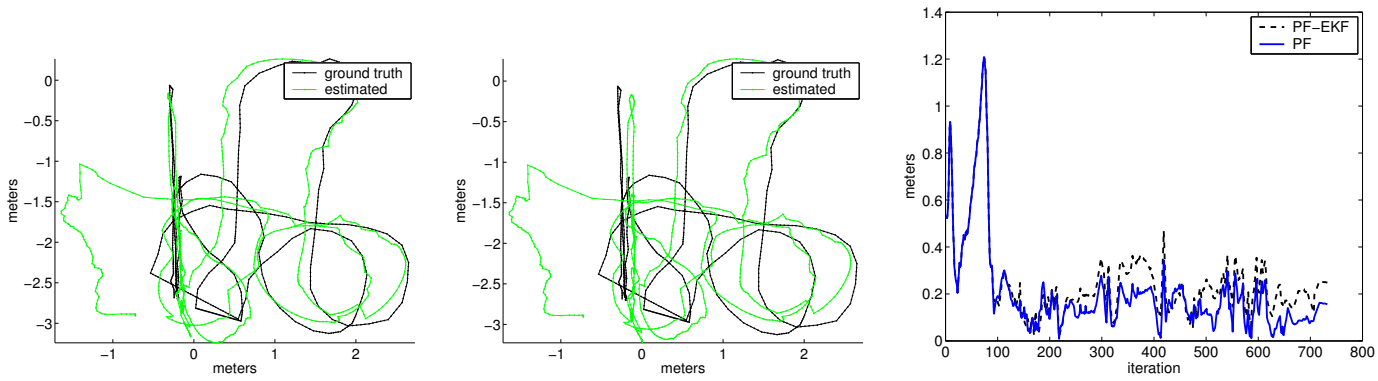
We have implemented our method on two robots, one equipped with an omni-directional camera, to evaluate the method presented in the previous sections. We also conducted an extensive test of the algorithm using simulated data. The main purpose of these simulated experiments was to analyze the performances of the EKF, the PF and the PF-EKF in terms of robustness and precision. In this section we first present our experimental setup, then the results obtained with simulated data and finally those obtained with real data.

A. Setup

The robots, a TRC Labmate from HelpMate Robotics Inc and a Nomadic's Scout, are differential drive mobile platforms (see figure 4(b)). The Labmate is equipped with an omnidirectional camera based on a spherical mirror.

The images acquired by the omnidirectional camera are dewarped to 360^o-wide panoramic views. The dewarping is just a Cartesian to polar transformation when the camera and the mirror are vertically aligned [7]. For general mountings of cameras and mirrors on robots, a projection operator, $\mathcal{P}(\mathbf{r};\vartheta)$, mapping world, $\mathbf{r}$ to image points is fitted to the cameras by a calibration procedure that retrieves the camera parameters ϑ. Given $\mathcal{P}(\mathbf{r};\vartheta)$ and an omnidirectional image, $I(\mathbf{m})$ the panoramic view, p is obtained as the image of a 3D cylinder, C i.e. $p = I(\mathcal{P}(C;\vartheta))$.

We model the omnidirectional camera with a projection operator based on the single projection centre model of Geyer and Daniilidis [9]. An omnidirectional camera based on the spherical mirror does not have the single projection



(a) (b) (c)

Fig. 5. Estimated trajectories using real data acquired by the robots. (a) PF implementation, (b) PF-EKF implementation (c) Position error at each iteration for both implementations.

centre property, however it approximates that property provided that the world structure is much larger than the mirror size [6]. Using the projection operator for dewarping to the panoramic view, instead of just transforming the image from Cartesian to polar coordinates, allows eliminating the errors in the azimuthal readings due to any camera-axis-deviation from the vertical direction. The omnidirectional camera produces 768×576 [pix] omnidirectional images, which are dewarped to 1091×148 [pix] panoramic views (see Figure 4(a)).

The panoramic views allow the Labmate to track the Scout, more precisely measuring its azimuthal location (bearing). This process is simplified as the Scout carries a light that can be easily detected in a region of interest. At the same time the robot acquires an image, the odometry is also observed. The data acquisition rate is about 2Hz. We used the Network Time Protocol to synchronize the platforms. The accumulated drift during the experiments was always less than one frame.

In order to obtain the ground truth of the trajectories of the robots we use one external camera placed in a corner of the room close to the ceiling. The perspective view of the room is transformed to an orthographic view of the ground plane using an homography. The homography is computed based on four points of the ground plane with known real-word coordinates (figure 4(c)). The robots are tracked by hand, marking in the image of the ground plane their locations. The heading of the Labmate is obtained from the orientation of its square-shaped footprint.

B. Simulations

We first tested our algorithm using simulated data. The main objective of these tests was to evaluate the accuracy and robustness of the EKF, the PF and the PF-EKF implementations. We generate 200 datasets where the robots moved at random speeds inside a 10x10 squared room. Both the bearing measurements and the odometry readings were corrupted with Gaussian noise. We limited the detection of the other robot to a maximum range of 8 meters and generate observations at $2Hz$. The PF and PF-EKF implementations used 1000 particles.

The KF implementation is very sensitive in the first steps due to the big uncertainty in the initial relative location. As expected it does not always converge to the good solution. It only converged in 40% of the simulations but in most of them the initialization of the filter was close enough to the true relative location. On the other hand the PF and PF-EKF converged to the solution for all the datasets. Table I summarizes the results for the PF and PF-EKF.

	μ_x	σ_x	μ_y	σ_y	μ_θ	σ_θ
PF	$0.11m$	$0.11m$	$0.10m$	$0.16m$	$0.07rad$	$0.08rad$
PF-EKF	$0.10m$	$0.15m$	$0.11m$	$0.18m$	$0.05rad$	$0.06rad$

TABLE I

MEAN AND STD OF THE ERRORS USING SIMULATED DATA

The results show that both implementations are able to estimate the relative position with similar accuracy. It is worth to note that the algorithm uses bearing measurements of moving landmarks. These measurements only contain information about one of the three state variables. In addition, the accuracy of the estimation depends on the relative motion of the platforms and on the distance between them. The measurement model is less precise as the distance increases. The standard deviations obtained in simulations reflect the variations of the localization error due to the influence of the distance between the platforms and their relative motion.

The differences between both methods are not statistically significant. However, after switching to an EKF the PF-EKF is much faster and it did not loose track of the other robot. The mean execution times per iteration are 109 milliseconds for the PF and 0.13 milliseconds for the PF-KF. These times correspond to a Matlab implementation on a Pentium IV at 1.2 Ghz and illustrate the different complexities of each implementation.

The results suggest that the PF is necessary only in the first steps of the algorithm to cope with the non Gaussian uncertainty that arises from the first observations and the linearization. Once the filter has converged around the truth location, the EKF is able to track the position with less computational cost. The number of steps needed

to switch to the EKF depends on the relative motion of the robots. Those trajectories that induce changes on the relative bearing are best suited for the localization task while those without changes provide less information.

C. Real data

We also tested our method with real data obtained using the setup presented above. In the experiments the vehicles were driven manually during approximately 400 seconds. Figure 4(d) shows the trajectories described by the platforms in one of these experiments.

The algorithm computes relative locations between the moving platforms. To plot the results we used the ground truth trajectory of the Labmate and the relative location estimation at each point in time to compute the trajectory of the Scout. Then we compare this trajectory with the Scout ground truth trajectory. Thus the Labmate is the observer and the Scout is the target. Figures 5(a) and (b) show the results of the PF and the PF-EKF implementations. At the beginning both algorithms are identical. After 137 iterations the particles have already converged around the true location and the PF-EKF starts tracking it with the EKF. Both trajectories are quite similar. Table II shows the mean and the standard deviation of the distance between the estimated position and the ground truth. Due to the precision of the ground truth data we are not able to compare the accuracy between the PF and the PF-EKF algorithms as the differences are not statistically significant.

	μ_x	σ_x	μ_y	σ_y
PF	$0.08m$	$0.06m$	$0.12m$	$0.09m$
PF-EKF	$0.11m$	$0.07m$	$0.15m$	$0.10m$

TABLE II

MEAN AND STD OF THE ERRORS USING REAL DATA

Figure 5(c) shows the position error at each step of the algorithm computed from the mean of the particles after the resampling step. The accuracy of the algorithm depends on the relative location of the robots and on their relative movement. During the first stages the error of both algorithms increases because the robots started moving in almost parallel trajectories. Particles were spread due to the noise in the odometry measurements. The measurement model is more precise for close locations. Those particles placed far away survived until the accumulated error in position is big enough to be removed in the resampling step. As a result the mean of the particle filter was shifted away (see the trajectories of Figure 5(a) and (b)). This situation lasted until the Scout platform turned right and the filter started to converge to the right relative location.

Once the method has converged the maximum error for both methods is around 0.4 meters. The error peaks correspond to those parts of the trajectory where the robots are further (up to five meters). The further the robot is, the less informative is the observation in terms of possible $x - y$ positions. This makes that the estimation error and the uncertainty of the relative location increase when the platforms moved away one from the other.

VII. CONCLUSIONS

We have presented a method to relatively localize a pair of robots fusing bearing measurements and the motion of the vehicles. Bearings are obtained as direct readouts of the omnidirectional camera. This is convenient as compared to measuring depth, which would require knowledge or reconstruction of the world / robot structure.

We have proposed three different implementations of the recursive Bayes filter based on an Extended Kalman filter, a particle filter and a combination of both techniques. The latter combines the benefits of each type of filter resulting in a robust and fast algorithm. We have shown several experimental results validating the solution to the problem and evaluating the different implementations.

As future work we are currently extending the algorithm to the case where both robots have detection capabilities. We also plan to study the generation of motions for the robots that improve the accuracy of the relative localization using the estimation available at each step.

VIII. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish government under the project MCYT-DPI2003-7986 and the Integrated Action HP2002-0037, and by the Portuguese government FCT Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III.

REFERENCES

[1] Y. Bar-Shalom, XR Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. J. Wiley and Sons, 2001.

[2] Andrew Blake and Michael Isard, editors. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, 1998.

[3] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *IEEE International Conference on Computer Vision*, pages 1403 – 1410 vol.2, 2003.

[4] A.J. Davison and D.W. Murray. Mobile robot localisation using active vision. In *European Conference on Computer Vision*, 1998.

[5] A. Doucet, N. Freitas, and N.J. Gordon. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, 2001.

[6] J. Gaspar, E. Grossmann, and J. Santos-Victor. Interactive reconstruction from an omnidirectional image. In *9th International Symposium on Intelligent Robotic Systems (SIRS'01)*, Toulouse, France, July 2001.

[7] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, December 2000.

[8] Donald B. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(3):243–270, April 1992.

[9] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical applications. In *European Conference on Computer Vision (ECCV) 2000*, pages 445–461, Dublin, Ireland, June 2000.

[10] A. Howard, M.J. Mataric, and G. Sukhatme. Putting the 'i' in 'team': an ego-centric approach to cooperative localization. In *IEEE Int. Conf. on Robotics and Automation*, Taiwan, 2003.

[11] A. Huster and S. Rock. Relative position sensing by fusing monocular vision and initial rate sensors. In *International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.

[12] H. Ishiguro, T. Sogo, and M. Barth. Baseline detection and localization for invisible omnidirectional cameras. *International Journal of Computer Vision*, 58(3):209–226, Jul.-Aug. 2004.

[13] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, June 1993.

[14] David G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, August 1992.

[15] M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (CARMEN) toolkit. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

[16] L. Montesano, L. Montano, and W. Burgard. Relative localization for pairs of robots based on unidentifiable moving landmarks. In *Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[17] H. Murase and S. K. Nayar. Visual learning and recognition of 3D objects from appearance. *Int. J. Computer Vision*, 14(1):5–24, January 1995.

[18] C. Musso, N. Oudjane, and F. Le Gland. *Sequential Monte Carlo Methods In Practice*, chapter Improving Regularised Particle Filters. Springer, 2001.

[19] T. Nakamura, M. Oohara, T. Ogasawara, and H. Ishiguro. Fast self-localization method for mobile robots using multiple omnidirectional vision sensors. *Machine Vision and Applications*, 14(2):129–138, 2003.

[20] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39 vol.1, 2004.

[21] D.T. Pham. Some quick and efficient methods for bearing only target motion analysis. *IEEE Transactions on Signal Processing*, 41(9), 1993.

[22] P. Pinheiro and P. Lima. Bayesian sensor fusion for cooperative object localization and world modeling. In *Conference on Intelligent Autonomous Systems*, Amsterdam, The Netherlands, 2004.

[23] M. Powers, R. Ravichandran, and T. Balch. Cooperative bearing-only tracking of multiple ambiguous targets. Submitted. Available online: www.cc.gatech.edu/ tucker/ papers/2004AgentsMattPowers.pdf.

[24] S. Roumeliotis and G.A. Beckey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5), 2002.

[25] Stephen Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. Journal of Robotics Research*, 21(8):735–758, 2002.

[26] J. Spletzer, K. Das, R. Fiero, C.J. Taylor, V. Kumar, and J.P. Ostrowski. Cooperative localization and control for multi-robot manipulation. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Hawai, USA, 2001.

[27] N. Trawny and T. Barfoot. Optimized motion strategies for cooperative localization of mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, New Orleans, USA, 2004.

[28] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware, proc. ieee conf. on computer vision and pattern recognition. In *IEEE, Int. Conf. on Computer Vision (ICCV)*, pages 211–218, 2003.