Identification of moving objects by a team of robots based on kinematic information

L. Montesano
montesano@unizar.es
Dept. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain

L. Montano
montano@unizar.es
Dept. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain

Abstract— **This paper describes a method for the identification of moving objects by a team of robots based on kinematic information. The objective is to be able to identify moving objects observed by different robots without using specific landmarks. Our method uses a bayesian approach and is based on the matching of maps of dynamic objects built by the members of the team. These maps contain the relative position of moving objects and their velocity at a given time. Experimental results using data from a real environment carried out to validate the method are presented and discussed.**

I. INTRODUCTION

Cooperative robotics has received considerable more attention from the robotic community due to the advances attained in the classic problems of mobile robotics. Besides, the scenarios where the robots are supposed to perform their tasks have become unstructured, dynamic and cluttered, increasing the challenge and demanding more complex capabilities from the robots. In this context, it becomes necessary to be able to characterize these complex environments and to share the information between the robots of a robot team. The work presented in this paper addresses the problem of mobile object identification by a team of robots. We assume that no common absolute reference system or relative pose estimation of the members of the team is available. This joint identification of the moving objects is necessary to perform tasks like relative localization of the robots, decision making, working area exploration or tracking of specific targets in a multirobot framework.

Usually, robot or object identification is performed using cameras, which provide information such as color, texture, shape and size. Often artificial landmarks, as color patches or bar codes, are used when a robot needs to identify other robots. For example, some recent works have addressed the problem of localization of a team of robots [3], [9], [6]. All of them use external systems or artificial landmarks to identify the members of the team. However, in some situations moving object identification can be difficult or even impossible using this kind of information. Identification of moving objects that do not belong to the team of robots (i.e. people) is a simple example of one of these situations. On the other hand, the robustness of the identification system will be increased if other type of information can be used together with the landmarks. It will also provide an alternative when the use of artificial landmarks on the robots is not possible or when they are temporaly not usable.

In the last years several authors have addressed the tracking of moving objects from a mobile platform (see [1] [10], [11], [7]). The kinematic information of the moving objects provided by some of these systems has not been fully exploited, as it is used only for tracking purposes. In the vision domain, Caspi et al. [2] use the dynamic information of each sequence of images (moving objects, changes of illumination) to resolve ambiguities that cannot be solved with the classic image to image alignment. The relative position of the cameras is not known but it remains constant. In our case, the robots are not moving together and they can even be sensing different areas.

We propose in this paper to exploit the location and kinematic information provided by a tracking system to identify the objects and the robots themselves. The basic idea is that if the same object is being tracked by two different robots, the trajectories and therefore the kinematic information observed by each robot must be compatible. The problem we have to solve can be stated as a correspondence problem between the information provided by every robot in the team. We have developed a tracking system similar to [5] using a 2D laser range finder. Using the information provided by the tracker, each robot creates a map of dynamic objects containing the locations and velocities of every moving object within the field of view of the sensor at a given time. We propose a probabilistic framework to cope with the correspondences between the objects tracked by each robot. A bayesian approach is used to compute the beliefs of the correspondences using the most probable correspondences at each time. These correspondences are obtained using a map matching technique [8] adapted to the maps of dynamic objects provided by the robots. To use this technique, the invariants applicable to the kinematic information contained in this type of maps have been identified. Experimental results obtained using real data have been carried out to validate the method and are presented and discussed. Finally, some extensions and future work are suggested.

II. IDENTIFYING MOVING OBJECTS

Our problem can be formulated as follows. Let $T_R = \{r_1...r_R\}$ be a team of R robots without any information about their absolute or relative positions. The robots move in a dynamic environment where some other moving objects can also be present. Each robot is equipped with a system that provides an estimation of its own displacement and velocity as well as the position and velocity of the moving objects within the field of view of its sensors. The objective is to identify the correspondences between the moving objects sensed by each robot. In other words, we aim to identify which object seen by the robot r_i corresponds with each object seen by robot r_j. Notice that the robots themselves are moving objects and therefore they can be observed by the other members of the team. Therefore, robot identification is a particular case of the general formulation presented below.

We first propose a solution for object identification by a pair of robots where the number of objects does not vary, i.e. new objects do not appear and the tracked objects do not dissapear from the sensing area of a robot. Then we show how to manage a variable number of objects in the scene and how the method scales to three or more robots.

A. Moving object identification for two robots

Let r_a and r_b be two different robots and N_a and N_b be the number of objects being tracked by r_a and r_b including the robot itself. We denote $\{O_{a,1},...,O_{a,N_a}\}$ and $\{O_{b,1},...,O_{b,N_b}\}$ the objects being tracked by r_a and r_b, respectively. $O_{a,1}$ represents r_a and $O_{b,1}$ represents r_b. At the moment we assume that N_a and N_b are fixed. Let $\mathbf{Z}_{r_a}(k) = \{\mathbf{z}_{a,1}(k),...,\mathbf{z}_{a,N_a}(k)\}$ and $\mathbf{Z}_{r_b}(k) = \{\mathbf{z}_{b,1}(k),...,\mathbf{z}_{b,N_b}(k)\}$ denote the observations of the objects sensed by each robot at time k. In our particular case the observations contain the position and velocities of the moving objects (See Sec. IV).

The problem to solve is to establish the correspondences between the objects of both robots at each point in time by using the observations $\mathbf{Z}^k_{r_a} = \{\mathbf{Z}_{r_a}(1),...,\mathbf{Z}_{r_a}(k)\}$ and $\mathbf{Z}^k_{r_b} = \{\mathbf{Z}_{r_b}(1),...,\mathbf{Z}_{r_b}(k)\}$ obtained until time k. We can formulate the method as a data association problem, in which we calculate the belief of the hypotheses of all the possible pairings between the objects being tracked by the robots.

Let x_{ij} be a discrete binary variable representing the hypothesis that $O_{a,i} = O_{b,j}$ for $i \in 1...N_a$ and $j \in 1...N_b$. Let $x_{i\emptyset}$ be a discrete binary variable representing the hypothesis that $O_{a,i} \neq O_{b,j}, \forall j \in 1...N_b$. We define $\mathbf{X}_{r_a r_b} = (x_{11},...,x_{1\emptyset},...,x_{N_a 1},...,x_{N_a\emptyset})^T$ as the state vector representing all the pairing hypotheses.

To calculate the belief of a hypothesis, we can compute the posterior probability $P(x_{ij,k}|\mathbf{Z}^k)$ of each discrete binary variable x_{ij}, where $\mathbf{Z}^k = \{\mathbf{Z}^k_{r_a}, \mathbf{Z}^k_{r_b}\}$ is the set of all observations until time k. Table I shows a graphic representation of the estimated probabilities of the variables of the state vector. Assuming a Markov process and applying Bayes rule we obtain:

	1	...	N_b	$\emptyset$
1	$P(x_{11})$		$P(x_{1N_b})$	$P(x_{1\emptyset})$
...		$P(x_{ij})$		
N_a	$P(x_{N_a1})$	...	$P(x_{N_aN_b})$	$P(x_{N_a\emptyset})$
$\emptyset$	$P(x_{\emptyset 1})$		$P(x_{\emptyset N_b})$	

TABLE I

BELIEFS OF THE STATE VECTOR.

$$P(x_{ij,k}|\mathbf{Z}^k) = \frac{P(\mathbf{Z}(k)|x_{ij})P(x_{ij,k}|\mathbf{Z}^{k-1})}{\sum_j P(\mathbf{Z}(k)|x_{ij})P(x_{ij,k}|\mathbf{Z}^{k-1})}, \quad i \in 1...N_a \;\; j \in \emptyset...N_b \tag{1}$$

If the number of objects remains constant, the prior information $P(x_{ij,k}|\mathbf{Z}^{k-1})$ is equal to the posterior probability $P(x_{ij,k-1}|\mathbf{Z}^{k-1})$ computed in the previous step. In the next Subsection we show how to compute this term when the number of objects varies. The term $P(\mathbf{Z}(k)|x_{ij})$ represents the likelihood of the observation $\mathbf{Z}(k)$ given an association x_{ij} (see Sec. III). The denominator is just a normalization factor. Rows and columns of Table I describe a complete set of mutually exclusive events.

$$\sum_j P(x_{ij}) = 1, \;\; j \in \emptyset...N_b; \quad \sum_i P(x_{ij}) = 1, \;\; i \in \emptyset...N_a$$

The row restriction is used to compute the normalization factor and the column one to estimate the elements of the last row of table I.

Actually the complete correspondence space for N_a and N_b objects is formed by all the possible joint associations. The dimension of this space has a complexity of order $\mathcal{O}(N_a^{N_b+1})$ [4]. Instead of considering all these possible states, we are just considering the states x_{ij} for objects $O_{a,i}$ and $O_{b,j}$, no matter which are the pairings for the rest of the objects. This allows us to reduce the complexity of the problem, while resolving the individual correspondence problem. The reduction is achieved by losing the interdependencies between the different individual associations. The resulting number of variables of the state vector, all the possible pairing hypotheses, is $N_a(N_b+1)$.

B. Managing appearing and disappearing objects

In a real dynamic scenario moving objects enter and leave the sensing area of the robots. Hence, the variables forming the state vector $\mathbf{X}_{r_a r_b}$ change. The terms $P(x_{ij,k}|\mathbf{Z}^{k-1})$ of this state vector are computed based on the posterior probabilities calculated in the previous step. Using the representation of table I, a new object $O_{a,l}$ detected by r_a adds a new row to the table. We need to give an a priori probability $P(x_{lj,k}|\mathbf{Z}^{k-1})$ to the new states x_{lj}:

$$P(x_{lj,k}|\mathbf{Z}^{k-1}) = \beta P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1}), \qquad j \in 1...N_b \tag{2}$$

$$P(x_{\emptyset j,k}|\mathbf{Z}^{k-1}) = P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1}) - P(x_{lj,k}|\mathbf{Z}^{k-1}) \tag{3}$$

The constant β determines the maximum amount of a priori belief that is allocated to the new objects. On the other hand, if object $O_{a,l}$ disappears from the sensing area of r_a, the row corresponding to this object is removed. The beliefs associated to the row are added to the last row, increasing the belief that objects of r_b are not associated to any of the objects of r_a:

$$P(x_{\emptyset j,k}|\mathbf{Z}^{k-1}) = P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1}) + P(x_{lj,k-1}|\mathbf{Z}^{k-1}), \qquad j \in 1...N_b$$

The same reasoning is applied for objects of r_b just by using columns instead of rows.

C. Scalability to a team of N robots

The previous Subsections describe a method for object identification by a pair of robots. A simple extension for a team of R robots consists in the application of the method for each couple of robots. Therefore, the number of possible pairs of robots is $\frac{R(R-1)}{2}$. For a team of R robots the complexity of the identification process scales with order $\mathcal{O}(R^2)$. However, as every robot of the team cooperates in the identification process, we can compute the correspondences for each pair of robots just once. In this case the complexity for each robot increases with order $\mathcal{O}(R)$, as each robot need to process $\frac{R-1}{2}$ pairs.

III. THE MATCHING ALGORITHM

To complete our formulation, it remains to compute the likelihood $P(\mathbf{Z}(k)|x_{ij})$ of an observation for the pairing of objects i and j (see Eq. (1)). As stated in Subsection II-A the space of all possible associations has order $\mathcal{O}(N_a^{N_b+1})$, which makes unfeasible to compute in real time the probability of each possible association even with a small number of objects. The observations $\mathbf{Z}_r(k)$ of a robot are interpreted as a map of dynamic objects at time k. We use a joint compatibility test (JC) similar to the one presented in [8] together with a branch and bound technique [4] to find the most probable association between the observations $\mathbf{Z}_{r_a}(k)$ and $\mathbf{Z}_{r_b}(k)$ of two robots at time k. The constraints used to restrict the number of associations to be explored while matching the maps of dynamic objects of two robots at time k are presented in Sec. IV.

The JC test is based on the Mahalanobis distance. Hence, the uncertainty of the observations of the objects is represented by gaussian distributions. Each map $\mathbf{Z}_r(k)$ is described by the mean vector $\hat{\mathbf{Z}}_r(k) = (\hat{\mathbf{z}}_{r,1}, ..., \hat{\mathbf{z}}_{r,n_r})$ and the covariance matrix $C_{\mathbf{Z}_r(k)}$ calculated from the tracking system we have developed. A function $\mathbf{f}_A(\mathbf{Z}_{r_a}, \mathbf{Z}_{r_b}) = 0$ is defined using the invariants presented in the next Section between the pairs of a possible association $A = \{i_1 j_1, ..., i_m j_m\}$, where m is the number of pairings of the association. Usually, $\mathbf{f}_A(\mathbf{Z}_{r_a}, \mathbf{Z}_{r_b})$ is a non linear function and it has to be linearized,

$$\begin{aligned}
\mathbf{f}_A(\mathbf{Z}_{r_a}, \mathbf{Z}_{r_b}) &\simeq \mathbf{h}_A + \mathbf{H}_A(\mathbf{Z}_{r_a} - \hat{\mathbf{Z}}_{r_a}) + \\
&\quad + \mathbf{G}_A(\mathbf{Z}_{r_b} - \hat{\mathbf{Z}}_{r_b}) \\
\mathbf{h}_A &= \mathbf{f}_A(\hat{\mathbf{Z}}_{r_a}, \hat{\mathbf{Z}}_{r_b})
\end{aligned}$$

At each sample period k, we compute the JC of the pairings of an association A using an innovation test on the joint innovation $\mathbf{h}_A$ as follows,

$$D_A^2 = \mathbf{h}_A^T \mathbf{C}_A^{-1} \mathbf{h}_A \; < \; \chi^2_{d,\alpha} \tag{4}$$

where $\mathbf{C}_A$ is the covariance of the joint innovation, α is the desired confidence level and $d = dim(\mathbf{f}_A)$. The probability of a spurious pairing being jointly compatible with all the pairings of an association decreases with the number of pairings of the association. Thus, the JC algorithm provides the set of longest jointly compatible associations, $H_{JC} = \{A_1, ..., A_M\}$. Note that due to occlusions, uncertainty and ambiguities, the JC algorithm can provide several hypotheses with the same number of pairings. Using the joint compatibility test instead of an individual compatibility one is justified by the nature of the constraints used. In the velocity space of Fig. 1 clutter appears frequently. The joint compatibility test manages these cluttered situations appearing in dynamic environments by taking into account the correlations between the observations of the objects of each single robot. For further details of the JC see [8].

To compute the likelihood $P(\mathbf{Z}(k)|x_{ij})$ we sum over all the hypotheses H_{JC} provided by the JC algorithm. The likelihood of a given association hypothesis $A \in H_{JC}$, $P(A|x_{ij})$, is computed as the product of the likelihood of all the pairings contained in the assciation A,

$$P(\mathbf{Z}(k)|x_{ij}) = \sum_{A \in H_{JC}} P(A|x_{ij}) = \sum_{A \in H_{JC}} \prod_{(l,q) \in A} g_{H_{JC}}(D_A^2)^c \; \gamma_{H_{JC}}^{1-c}$$

$$c = \begin{cases} 0, & if \; (l = i \; and \; q \neq j) \; or \; (l \neq i \; and \; q = j) \\ 1, & otherwise \end{cases}$$

If a pairing (l,q) and x_{ij} are compatible, the likelihood $P((l,q)|x_{ij}) = g_{H_{JC}}(D_A^2)$, where $g_{H_{JC}}$ is a function of the Mahalanobis distance of the hypothesis A. If they are not compatible, the term $P((l,k)|x_{ij})$ receives a residual value $\gamma_{H_{JC}}$. In our current implementation $g_{H_{JC}}$ is a bell-shaped exponential function centered at 0. The maximum and the decrease of the function as well as the residual likelihood depend on the maximum number of possible pairings of an association and the number of pairings of H_{JC}. This way, the longer the hypothesis the higher the likelihood associated to the pairings belonging to it. The subscript

Algorithm 1: Step k of the object identification algorithm.

Iteration k
INPUT: $P(x_{ij,k-1}|\mathbf{Z}^{k-1}),\ \forall\ i,j;\ \mathbf{Z}(k)=\{\mathbf{Z}_{r_a}(k),\ \mathbf{Z}_{r_b}(k)\}$.
Step 1: Compute the prior probabilities $P(x_{ij,k}|\mathbf{Z}^{k-1})$
1.1- Remove objects
for all removed object l of r_a **do**
$P(x_{\emptyset j,k}|\mathbf{Z}^{k-1})=P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1})+P(x_{lj,k-1}|\mathbf{Z}^{k-1}),\forall j$
end for
Same loop over removed objects of r_b
1.2- Add new objects
for all new object l of r_a **do**
$P(x_{lj,k}|\mathbf{Z}^{k-1})=\beta P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1}),\quad \forall j$
$P(x_{\emptyset j,k}|\mathbf{Z}^{k-1})=P(x_{\emptyset j,k-1}|\mathbf{Z}^{k-1})-P(x_{lj,k}|\mathbf{Z}^{k-1})$
end for
Same loop over new objects of r_b
Step 2: Compute the likelihood of the observations
2.1- Generate the most probable associations:
$H_{JC}=JC(\mathbf{Z}(k))$
2.2- Compute likelihoods:
for all $x_{ij}\in\mathbf{X}$ **do**
$P(\mathbf{Z}(k)|x_{ij})=\sum_{A\in H_{JC}}\prod_{(l,q)\in A} g_{H_{JC}}(D_A^2)^c\ \gamma_{H_{JC}}^{1-c}$
end for
Step 3: Compute the posterior probabilities $P(x_{ij,k}|\mathbf{Z}^k)$
for all $x_{ij}\in\mathbf{X}$ **do**
$P(x_{ij,k}|\mathbf{Z}^k)=\frac{P(\mathbf{Z}(k)|x_{ij})P(x_{ij,k}|\mathbf{Z}^{k-1})}{\sum_j P(\mathbf{Z}(k)|x_{ij})P(x_{ij,k}|\mathbf{Z}^{k-1})}$,
end for
OUTPUT: $P(x_{ij,k}|\mathbf{Z}^k),\ \forall i,j$.

H_{JC} shows this dependency for both the function $g_{H_{JC}}$ and the residual value $\gamma_{H_{JC}}$. Algorithm 1 summarizes a complete step of the algorithm to compute the probabilities of the state vector at time step k.

IV. CONSTRAINTS USING KINEMATIC INFORMATION

In this Section we analyze how to use the kinematic information for matching the observations obtained by the robots. In the absence of other types of information, we characterize a moving object by the position of a characteristic point (i.e. centroid) $\mathbf{p}=(x,y)^T$ and its velocity $\dot{\mathbf{p}}=(\dot{x},\dot{y})^T$ in the robot reference system. Using the notation of previous Sections, we denote an observation of an object $O_{r,i}$ in the reference of the robot r at time k as $\mathbf{z}_{r,i}(k)=(\mathbf{p}_{ri}(k),\dot{\mathbf{p}}_{ri}(k))^T$. As the objective is to match and identify objects between two unrelated robots, we are interested in those invariant characteristics in their reference system.

We analyze next four invariants that can be used as unary or binary constraints to match objects. These invariants are used as constraints in Sec. III to match the maps of dynamic objects of two robots.

(1) Euclidean distance between two locations.

The euclidean distance is a well known invariant that has been widely used as a binary constraint [4] for object recognition, robot localization and SLAM.

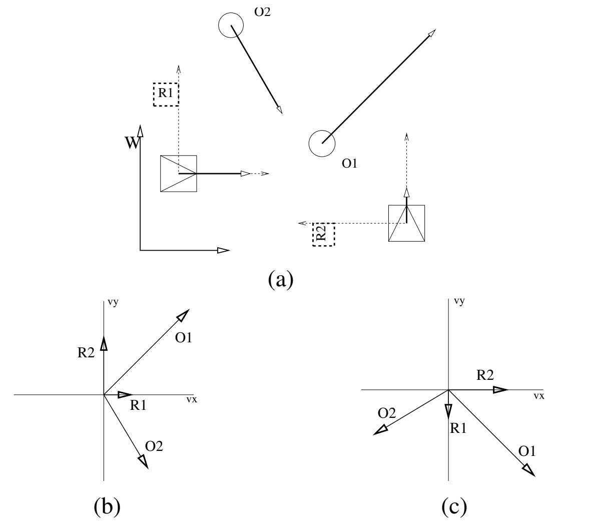


Fig. 1. (a) Scenario with two mobile robots and two moving objects, (b) Velocities of moving objects viewed by Robot 1 expressed in Robot1 reference system (c) Velocities of moving objects viewed by Robot 1 expressed in Robot2 reference system

(2) Velocity module and (3) angle between a pair of velocity vectors

Velocity constraints are not so widely used but are useful to solve the proposed problem. They depend only on the relative orientation of the two reference systems. Let a and b be two different reference systems and $\mathbf{q_{ba}}=(tx\ ty\ \theta)^T$ the relative pose of a with respect to b. Let $\mathbf{p}_a$ and $\mathbf{p}_b$ be the position of an object in the reference systems a and b, respectively. The equation transforming the location of the object from one reference system to the other and its derivative are

$$\mathbf{p}_b=R_{ba}\mathbf{p}_a\ +\ \mathbf{t}_{ba}\qquad \dot{\mathbf{p}}_b=J_{ba}\dot{\mathbf{p}}_a \tag{5}$$

where R_{ba} and $\mathbf{t}_{ba}$ are the rotation matrix and the relative position between a and b, respectively, and J_{ba} is the Jacobian of the transformation from the reference a to the reference b. The previous equations assumes that the relative position between both reference systems remains constant.

From Eqs. 5 we conclude that two objects moving with the same velocity $\dot{\mathbf{p}}$ cannot be distinguished. Using a polar representation (v,ϕ) of the velocity vector $\dot{\mathbf{p}}$ it is easily shown that the module v is an invariant in the reference system. On the other hand the direction of the vector ϕ is rotated an angle θ,

$$v_b=v_a\qquad \phi_b\ =\ \phi_a\ +\ \theta \tag{6}$$

Therefore, the module is used as an unary constraint and the angle between two velocity vectors is used as a binary constraint. Unfortunately, in most situations the velocity module is not discriminant enough. Usually, objects of the same type move with similar velocities in module. For instance, people walking speed is usually between

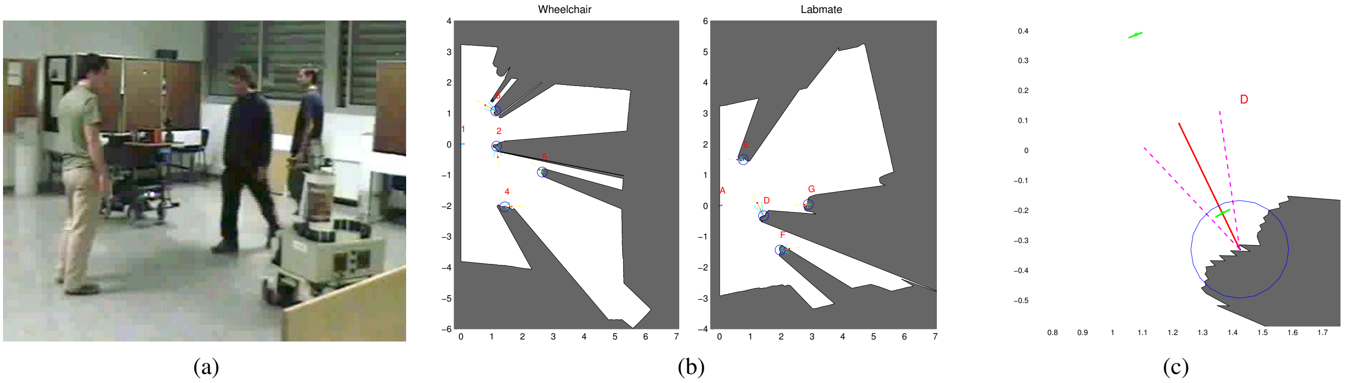


Fig. 2. (a) Scenario with two mobile robots and three moving people. (b) Map of dynamic objects generated by the tracking system of the wheelchair and the labmate respectively. (c) Detail of the location and velocity of an object and their uncertainties.

$0.7m/seg$ and $1m/seg$ and it is difficult to discriminate them in the presence of noise. Fig. 1(a) depicts 2 robots and 2 moving objects with their absolute velocities in an unknown global reference system. Fig. 1(b) and 1(c) show the velocity maps observed by each robot. As stated before the velocity maps are just rotated an angle θ.

(4) Angle between a segment joining a pair of points and a velocity vector.

The fourth invariant uses the orientation of the segment between two points. Without using kinematic information, angles between segments are a ternary constraint. Using the velocity vector we measure the angle between the orientation of the segment joining the positions of two objects and the orientation of one of their velocity vectors. This binary restriction is used to disambiguate a symmetric situation when the previous constraints are not discriminant. That is, when both the distances between two pairs of objects and the angle of their velocity vectors are compatible. Throughout our experiments we found that in some cluttered environments this constraint reduces considerably the number of nodes to be explored and improves the robustness of the algorithm, specially when the uncertainty increases.

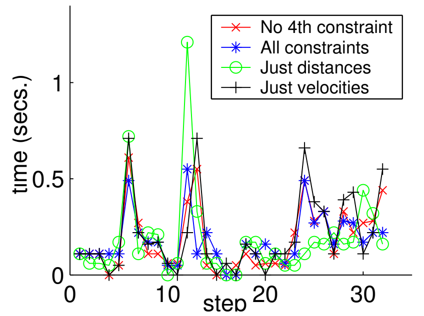
A. *Sources of uncertainty*

We have described four invariants that can be used to match maps of dynamic objects. Unfortunately, sensors information is noisy. Furthermore, a multirobot scenario introduces new sources of uncertainty that must be taken into account. First, sensors can be non perfectly synchronized, unsynchronized or even have different scan rates. This makes that the discretization of the trajectory described by a moving object will not be the same for each robot. Although prediction or smoothing techniques [1] can be used to temporally align the maps, the resulting trajectory observed by each robot will be distorted. The second source or uncertainty is the estimation of the position of the centroid of an object. Usually objects are only partially observed. As they are viewed from different points their estimated centroid positions at a given time will not be the same for different robots. The precision of the estimated position of the centroid will depend on the sensor and on the type of objects being tracked. A priori knowledge of the shape of the objects being tracked allows to better estimate the centroid, but restricts the method to the type of objects being considered. Finally, the motion of the robot is not perfectly known. The uncertainty of the ego motion estimation method used by the tracking system will also introduce noise in the estimated velocities of the objects. Results provided in Sec. V show how our method cope with these uncertainties.

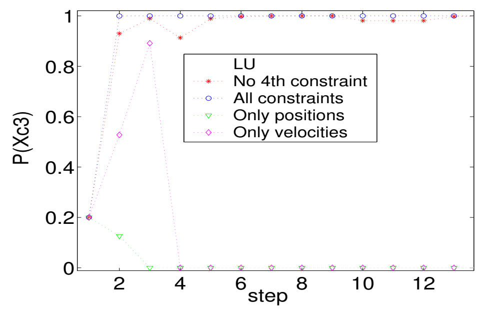
V. EXPERIMENTAL RESULTS

We have tested the previous method in several experiments in our laboratory using the two robots shown in Fig. 2(a). Both robots, a Labmate platform and an automatic wheelchair, are equipped with a Sick laser range finder LMS200. During the experiments the robots moved describing different trajectories while up to four people walked randomly around them. The method is currently implemented in Matlab. It was run off-line using the recorded real data from the experiments in a Pentium III at 800Mhz. In our current implementation the temporal alignment is not corrected. Nevertheless, the maximum synchronization error is limited to 110ms. The results show that the method can cope with the temporal error, mainly because of the uncertainty introduced in the estimation of the centroid.

A tracking system similar to [5], developed at our laboratory, is implemented on each robot and generates the maps of dynamic objects used as the input to the identification method. An example of one of these maps is shown in Fig. 2(b) and (c). The white area represents the free space detected by the last laser scan. Each moving object is represented by the position of its centroid, its velocity (red solid segment) and their associated uncertainties. The circle represents the centroid uncertainty. The

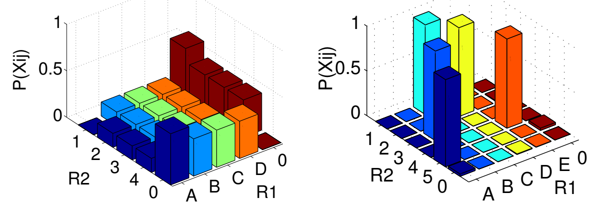


(a)

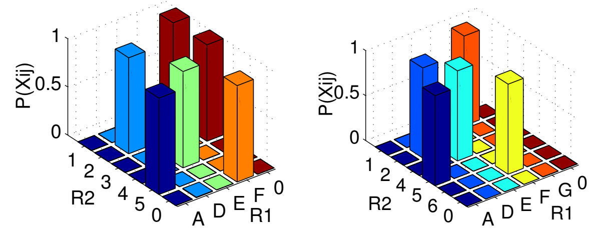


(b)

Fig. 3. (a) Execution times of the algorithm for different set of constraints for LU maps. (b) Belief evolution for a right association event for the LU case.



(a) (b)



(c) (d)

Fig. 4. Correspondences estimated probabilities: (a) a priori probabilities, (b) step 6, (c) step 16, (d) step 20. The (x-y) axes contain the labels of the objects detected by each robot, respectively. The 0 label corresponds to the null association.

angular uncertainty of the velocity vector is represented by the two dotted segments surrounding the velocity vector. The two small segments crossing the velocity direction represent the module uncertainty. In order to test the performance and robustness of the method, the tracking system generated two different maps of dynamic objects using the uncertainty of the centroid estimation process (typically about $\sigma_1 \approx 0.20cm$) and an artificially increased one ($\sigma_2 = 4\sigma_1$). We will refer to each case as low uncertainty (LU) and high uncertainty (HU), respectively.

We have analyzed the influence in the success of the method of: (1) the constraints applied in the matching algorithm using only locations, only velocities or both together; (2) the number of objects in the scene; and (3) the uncertainty of the maps. We provide results about the execution time of the method, the evolution of the beliefs of correspondences and the response time, i.e. the number of steps needed to identify an object.

We now describe how the performance of the algorithm depends directly on the type of constraints used in the matching algorithm. If the maps of dynamic objects do not contain clutter or ambiguities, the results obtained using only locations or only velocities are similar to the ones obtained using both together. However, when the number of objects or their uncertainty increases, clutter and ambiguities appear more frequently and only velocities or locations are not able to find the right correspondences. As expected, the use of both types of information at the same time improves the robustness of the method and allows it to find the correct pairings. However, in some cases the JC algorithm provides some incomplete or even erroneous pairings. The estimated correspondences using Bayes take into account the whole set of observations obtained until time step k filtering the wrong associations provided by the JC.

With regard to the execution times, using all constraints is more expensive when there are no ambiguities between the maps. However, for cluttered scenarios the execution time of the matching algorithm using all the constraints is similar for all the cases. Using all the constraints can even improve the execution time, as the bound technique is able to discard more nodes. Fig. 3(a) shows the times obtained in the identification method using different constraints for a total number of 6 moving objects (4 people and 2 robots). The peaks corresponds to ambiguous situations where almost all the possible nodes need to be examined. Table II contains the mean and maximum execution times

Time (sec.)	LU		HU	
# obj	μ	max	μ	max
2	0.01	0.06	0.06	0.06
3	0.03	0.11	0.06	0.11
4	0.11	0.22	0.18	0.55
5	0.29	0.78	0.69	1.4
6	0.53	1.76	1.23	3.5

TABLE II

EXECUTION TIME DEPENDING ON THE NUMBER OF OBJECTS FOR A NON OPTIMIZED MATLAB IMPLEMENTATION.

obtained for different number of objects. It shows how the performance of the JC algorithm degrades when the uncertainty, and consequently the number of wrong compatible hypothesis, increases. However, the maximum obtained for 6 objects correspond to a particular situation produced by a mistake of the tracking system. In a cluttered situation, two filters were created to track the same object which produces this maximum. The number of filters is corrected after two steps and the execution time reduced.

We have also analyzed the number of steps needed to identify an object, which can be interpreted as the response time of our method. As in the previous case, the results depend on the characteristics of the scenario. For the low uncertainty case, the response time does not depend on the constraints used. Only in some particular configurations, the response time obtained by using all the constraints is better than the others. In most of the cases, the correct pairings are achieved using the observations of about three steps. When the uncertainty increases, the response time using all constraints is better than the others. Furthermore, in a high percentage of the cases using only locations or velocities does not obtain the right correspondences. The method response for a correct pairing for the LU and HU maps is shown in fig. 3(b).

Finally, fig. 4 shows the estimated beliefs of our state vector at different time steps using all the constraints. Fig. 4(a) corresponds to the a priori beliefs for a given number of tracked objects. After 5 steps, Fig. 4(b), the estimated probabilities of all the right correspondences are close to one. Fig. 4(c)(d) show how the identification matrix evolves when the number of objects varies. First, the probabilities are redistributed among the new objects keeping the already existing correspondences. After two more steps, the correct correspondences for the new objects have been identified again.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a method for moving objects identification by a team of robots based only on the relative locations and the velocities of the moving objects. The experimental results show how this method is able to quickly identify the right correspondences even in the presence of high uncertainty. We have shown how the joint use of location and kinematic information of the moving objects can be utilized to solve the correspondence problem, improving the performance and robustness of the method. There is still some on going work been carried out to determine the best likelihood estimation for an association provided by the joint compatibility algorithm and its influence on the identification method. The use of an individual compatibility test is also been considered to reduce the amount of shared data and the execution times. The method can be interpreted as a trajectory matching algorithm where at each time step trajectories defined by the last two observed positions are matched. The sequential integration using Bayes extends the matching to the whole trajectory eliminating the spurious matches given by the JC algorithm and increasing the robustness of the identification.

This paper has addressed the identification problem in the correspondence space by matching the maps of moving objects generated by the robots from their observations. We think that the method can be combined with other static map matching techniques to match maps containing both static and dynamic information. The problem could also be solved in the pose space instead of using the correspondence space by estimating the relative pose between the robots. We intend to explore this approach and compare both solutions in terms of complexity, accuracy and scalability.

VII. ACKNOWLEDGMENTS

This work was partially supported by the project MCYT DPI2000-1272.

VIII. REFERENCES

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering. Academic Press., 1988.

[2] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002.

[3] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 2000.

[4] W. Eric. L. Grimson. *Object Recognition by Computer. The role of Geometric Constraints.* The MIT Press, London, 1990.

[5] D. Hahnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[6] 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.

[7] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people tracking. In *IEEE Int. Conf. on Robotics and Automation*, Washington, USA, 2002.

[8] J. Neira and J.D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automations*, 17(6), 2001.

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

[10] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association. In *IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.

[11] C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *IEEE Int. Conf. on Robotics and Automation*, Washington, USA, 2002.