June 2003

Localization and Tracking in Robot Networks

Guilherme A. S. Pereira
Universidade Federal de Minas Gerais

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Mario F. M. Campos
Universidade Federal de Minas Gerais

From the *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, held June 30-July 3, 2003, Coimbra, Portugal.

Localization and Tracking in Robot Networks

Abstract
We are interested in the real time pose estimation of a group of networked mobile robots based on exteroceptive information from their vision systems. Additionally, we are also interested in simultaneously tracking a rigid unknown object. We present a localization and object tracking approach based on statistical operators and simple graph searching algorithms. The approach was implemented in our team of five car like robots equipped with omni-directional video cameras and IEEE 802.11b wireless networking.

Comments
From the *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, held June 30-July 3, 2003, Coimbra, Portugal.

Localization and Tracking in Robot Networks

Guilherme A. S. Pereira[§] Vijay Kumar[†] Mario F. M. Campos[§]

[§]VERLab, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil.

[†]GRASP Lab., University of Pennsylvania, Philadelphia, PA, USA.

{gpereira, mario}@dcc.ufmg.br, kumar@grasp.cis.upenn.edu

Abstract

We are interested in the real time pose estimation of a group of networked mobile robots based on exteroceptive information from their vision systems. Additionally, we are also interested in simultaneously tracking a rigid unknown object. We present a localization and object tracking approach based on statistical operators and simple graph searching algorithms. The approach was implemented in our team of five car like robots equipped with omni-directional video cameras and IEEE 802.11b wireless networking.

1 Introduction

A robotic sensing system may be as simple as a single sensor or composed by multiple sensors. In the last scenario, data are processed and combined to provide information that is more reliable and complete when compared to the single sensor approach. We are interested in situations where sensors are placed on networked mobile robots programmed to perform a large variety of cooperative tasks such as search and rescue, surveillance, manipulation, and tracking.

In this paper our objective is to estimate in real time the position and orientation of a group of mobile robots using only information from their vision systems. Additionally, we are also interested in simultaneously tracking a rigid unknown object. Our approach is based on the fact that the combination of multiple simultaneous observations of the same object can provide information that is more complete, more accurate, and more robust when compared to a single observation.

The approach proposed here is closely related to those presented in [1] and [2] in the sense that the robots have access to their teammates sensor data (or some related information) and combine these information with the one coming from its own sensors. In those papers, the robots use distributed sensing to improve self localization [1] or target localization [2]. Both papers rely their methodologies on Kalman Filters. Papers [3] and [4] present solutions for the relative multirobot localization problem by combining information exchanged by the robots using least square optimization. In this paper we present a different approach for localization and object tracking based on statistical operators and simple graph searching algorithms. Furthermore, differently from the previous approaches, we formulate the problem in such a way that localization and object tracking can be solved by the same algorithm. We also show how the advantages related to optimality of previous works can be easily incorporated in our methodology. As an example, we show how to incorporate an Extended Kalman Filter (EKF) in order to improve object tracking. In order to validate our methodology we show results obtained with our team of five car like robots (Figure 1) equipped with omni-directional video cameras and IEEE 802.11b wireless networking.

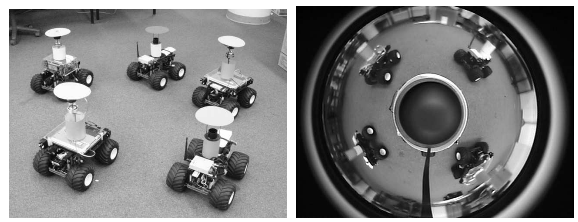

Figure 1: The GRASP Lab. ClodBusters (left) and a sample image from an omni-directional camera (right).

2 Mathematical Modeling

Consider a planar world, $\mathcal{W} = \mathbb{R}^2$, occupied by a rigid polygonal object with m edges and a group $\mathcal{R} = \{R_1, R_2, \ldots R_n\}$ of n robots. The i^{th} robot R_i is represented by the configuration $q_i = (x_i, y_i, \theta_i)$. The object is described by its corner set $\mathcal{O} = \{O_1, O_2, \ldots O_m\}$ where each O_j is represented by the configuration $o_j = (x_j, y_j)$.

The physical locations of the robots coupled with the characteristics of the hardware and the requirements of the sensing and control algorithms dictate the sensing network for the group of robots. This network can be represented by a graph. Thus, let $G = (\mathcal{V}, \mathcal{E}, \mathcal{Z}, \mathcal{P})$ be a sensing graph where $\mathcal{V} = \mathcal{R} \cup \mathcal{O}$ is the set of vertices, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges that represent the presence of measurements between two vertices, $\mathcal{Z}$ is the set of measurements, and $\mathcal{P}$ is a set of variances that represent the quality of those measurements. Observe that our graph is a directed graph. Then, if a vertex v_j has a incoming edge from a vertex v_i, it means v_i has sensing information about v_j. This edge will be represented by $e_{ij} = (v_i, v_j)$.

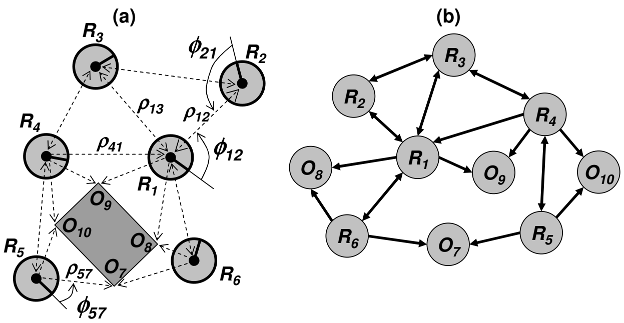

Figure 2: (a) – A group of robots localizing and tracking a rectangular object; and (b) – a sensing graph for this snapshot.

Clearly in this case v_i has to be a robot ($v_i \in \mathcal{R}$) while v_j can be either a robot or an object corner ($v_j \in \mathcal{R} \cup \mathcal{O}$). Each element e_{ij} is associated with an element $z_{ij} \in \mathcal{Z}$. An element z_{ij} is a tuple (ρ_{ij}, ϕ_{ij}), compose by the range and bearing measurements of the j^{th} vertex in relation to the i^{th} vertex in i's reference frame. Each element z_{ij} is associated with one element $p_{ij} = (\sigma^2_\rho, \sigma^2_\phi) \in \mathcal{P}$, where σ^2_ρ and σ^2_ϕ are the variances of ρ_{ij} and ϕ_{ij} respectively. The representation of measurements by range and bearing came naturally due to the omnidirectional vision systems used by our robots as sensors. In these systems the coordinates can be directly measured and their quality directly estimated. Observe we do not assume any kind of proprioceptive information such as robot's velocity and acceleration. Figure 2 shows an example of our graph modelling.

3 Measurements Transformation

In the previous section we have assumed that the variables measured by each robot are target's range and bearing. This representation is very convenient for estimation of robots' orientations but needs to be converted in order to estimate the robots positions. Observe in Figure 3(a) that ρ_{ij} and ϕ_{ij} can be converted to x_{ij} and y_{ij}, which represent R_j's position in R_i's reference frame as:

$$\begin{aligned} x_{ij} &= f(\rho_{ij}, \phi_{ij}) = \rho_{ij}\cos(\phi_{ij}) \\ y_{ij} &= g(\rho_{ij}, \phi_{ij}) = \rho_{ij}\sin(\phi_{ij}) \,. \end{aligned} \tag{1}$$

In order to transform the associate covariance matrix of ρ_{ij} and ϕ_{ij} and obtain a covariance matrix for x_{ij} and y_{ij}, we will use the Jacobian of the transformation as proposed in [5]. This is a approximated methodology that works very well when the variables can be represented by unbiased normal distributions with small standard deviations. We are also assuming that ρ_{ij} and ϕ_{ij} are independent and consequently we can write the covariance matrix of the robots measurements as:

$$\mathbf{P}_{\rho\phi} = \begin{bmatrix} \sigma^2_\rho & 0 \\ 0 & \sigma^2_\phi \end{bmatrix} \,.$$

The Jacobian of the transformation, $\mathbf{J}$, relates the

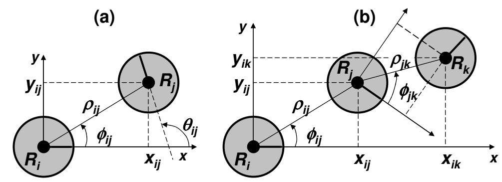

Figure 3: (a) – Local transformation of variables; (b) – Sequential transformation.

deviation of original and transformed variables as:

$$\begin{aligned} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{J}\begin{bmatrix} \Delta\rho \\ \Delta\phi \end{bmatrix} &= \begin{bmatrix} \frac{\partial f}{\partial \rho} & \frac{\partial f}{\partial \phi} \\ \frac{\partial g}{\partial \rho} & \frac{\partial g}{\partial \phi} \end{bmatrix}\begin{bmatrix} \Delta\rho \\ \Delta\phi \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi) & -\rho\sin(\phi) \\ \sin(\phi) & \rho\cos(\phi) \end{bmatrix}\begin{bmatrix} \Delta\rho \\ \Delta\phi \end{bmatrix} \,. \end{aligned} \tag{2}$$

If we multiply both sides of Equation (2) by the respective transposes and take the expectation matrix of the result we have the transformed covariance matrix as:

$$\mathbf{P}_{xy} = \mathbf{J}\,\mathbf{P}_{\rho\phi}\,\mathbf{J}^T \,, \tag{3}$$

which is a matrix of the form:

$$\mathbf{P}_{xy} = \begin{bmatrix} \sigma^2_x & \alpha\,\sigma_x\sigma_y \\ \alpha\,\sigma_x\sigma_y & \sigma^2_y \end{bmatrix} \,, \tag{4}$$

where σ^2_x and σ^2_y are covariances along x and y respectively and α is their correlation coefficient.

Together with x_{ij} and y_{ij}, another variable of interest is the robots' relative orientation, θ_{ij}. Although it can not be estimated by a single robot measurement, if two robots exchange their bearing measurements, the relative orientation can be estimated as:

$$\theta_{ij} = \phi_{ij} - \phi_{ji} + \pi \,, \tag{5}$$

and its variance as:

$$\sigma^2_{\theta_{ij}} = \sigma^2_{\phi_{ij}} + \sigma^2_{\phi_{ji}} \,.$$

We are also interested in transforming variables measured by one robot to another robot's reference frame. In Figure 3(b), for example, it could be interesting for R_i to have the the relative position and orientation of R_k even when they cannot sense each other. In this specific situation R_i can sense R_j which can sense R_k. Therefore, we want to transform R_j's measurements to R_i's reference frame.

Assuming positions $(x_{jk},\ y_{jk})$ and $(x_{ij},\ y_{ij})$, and orientations θ_{jk} and θ_{ij} respectively of R_k in relation to R_j and R_j in relation to R_i are available, the expected values of x_{ik}, y_{ik} and θ_{ik} are given by a well known frame transformation:

$$\begin{aligned} x_{ik} &= x_{ij} + x_{jk}\cos(\theta_{ij}) - y_{jk}\sin(\theta_{ij}) \\ y_{ik} &= y_{ij} + x_{jk}\sin(\theta_{ij}) + y_{jk}\cos(\theta_{ij}) \\ \theta_{ik} &= \theta_{ij} + \theta_{jk} \,. \end{aligned} \tag{6}$$

Repeating the procedure explained before, we can obtain the Jacobian matrix of the transformation as:

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & -x_{jk}\sin(\theta_{ij}) - y_{jk}\cos(\theta_{ij}) & \cos(\theta_{ij}) & -\sin(\theta_{ij}) & 0 \\ 0 & 1 & x_{jk}\cos(\theta_{ij}) - y_{jk}\sin(\theta_{ij}) & \sin(\theta_{ij}) & \cos(\theta_{ij}) & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} ,$$

and the covariance matrix of the measurements of R_j in R_i's reference frame as:

$$\mathbf{P}_{ik} = \mathbf{J} \begin{bmatrix} \mathbf{P}_{ij} & \alpha \mathbf{P}_{ij} \mathbf{P}_{jk} \\ \alpha \mathbf{P}_{ij} \mathbf{P}_{jk} & \mathbf{P}_{jk} \end{bmatrix} \mathbf{J}^T , \qquad (7)$$

where α is a matrix that correlates $\mathbf{P}_{ij}$ and $\mathbf{P}_{ki}$.

4 Measurements Combination

In the previous section we have shown how to transform the robots' measurements and how to estimate their covariance matrices. In this section we will see how to combine these information in order to have pose estimation of both robots and targets.

As proposed in [5], by using the Kalman filter equations for static-state estimation, if two estimates $\hat{q}_1 = [x_1\ y_1\ \theta_1]^T$ and $\hat{q}_2 = [x_2\ y_2\ \theta_2]^T$ of the same variable q, are expressed in the same reference frame, and have covariance matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, then a better estimate $\hat{q}$ of q can be obtained as:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_1 \left(\mathbf{P}_1 + \mathbf{P}_2\right)^{-1} \\ \hat{q} &= \hat{q}_1 + \mathbf{K} \left(\hat{q}_2 - \hat{q}_1\right) \\ \mathbf{P} &= \mathbf{P}_1 - \mathbf{K}\,\mathbf{P}_1 , \end{aligned} \qquad (8)$$

where $\mathbf{K}$ is the Kalman gain and $\mathbf{P}$ is the resulting covariance matrix. It is easy to verify that in the case of independent one dimension measurements these equations reduce to:

$$\begin{aligned} \hat{\beta} &= \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\,\hat{\beta}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\,\hat{\beta}_2 \\ \sigma^2 &= \frac{\sigma_2^2\,\sigma_1^2}{\sigma_1^2 + \sigma_2^2} , \end{aligned} \qquad (9)$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ are the estimates and σ_1^2 and σ_2^2 are their covariance. These one dimensional equations are very useful and can be used, for example, in order to merge two values of range measured by two neighboring robots before their transformation in x and y using (1).

The previous equations are very direct and can combine any number of estimates if they can be paired. It's well know, however, that the Kalman filter is an iterative way to solve a weighted least squares problem since its equations are based on the minimization of the sum of the squares of the errors. In this way, if more than two measurements are available any general weighted least squares method (including iterative and more efficient ones) can be used in order to produce the same results obtained when several applications of (8) are made. Thus, if all measurements can be grouped into a linear system as:

$$\mathbf{A}\,\hat{q} = \mathbf{Q} , \qquad (10)$$

a closed formula for the weighted least squares method is given by:

$$\hat{q} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{Q} , \qquad (11)$$

where $\mathbf{Q} = [\hat{q}_1\ \ \hat{q}_2\ \ \ldots\ \ \hat{q}_k]^T$ is the vector of measurements to be combined and $\mathbf{W} = diag([\mathbf{P}_1^{-1}\ \ \mathbf{P}_2^{-1}\ \ \ldots \mathbf{P}_k^{-1}])$ is the weight matrix. The covariance matrix of the result is given by:

$$\mathbf{P} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1} .$$

As an example, consider the combination of three one dimensional measurements. The linear system can be posed as:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} , \qquad (12)$$

and the weighted least square problem can be solved as:

$$\hat{\beta} = \left[[1\ 1\ 1] \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_2^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_3^2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right]^{-1} [1\ 1\ 1] \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_2^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_3^2} \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} ,$$

$$\hat{\beta} = \frac{\sigma_2^2\sigma_3^2\hat{\beta}_1 + \sigma_1^2\sigma_3^2\hat{\beta}_2 + \sigma_1^2\sigma_2^2\hat{\beta}_3}{\sigma_1^2\sigma_2^2 + \sigma_2^2\sigma_3^2 + \sigma_1^2\sigma_3^2} . \qquad (13)$$

One can easily verify that (13) can also be obtained by combining $\hat{\beta}_1$ and $\hat{\beta}_2$ using (9) and than combining the result with $\hat{\beta}_3$ using the same equation. Thus, observe that for linear systems, the recursive application of (8) and the least squares methodologies are equivalent and provide the same results. However, (11) cannot be directed applied for non-linear systems because, without initial values for the variables, it is not possible to compute a complete linear approximation of the system using the Jacobian. In these cases a recursive combination using (8) on the locally linearized systems is necessary.

In the discussion above we are assuming that data to be combined are always measurements of the same variable estimated in the same reference frame. When measurements are made in different frames, the transformation steps discussed in the previous section must be used. However, when linear combinations of the variables of interest are available, they can be used directly in (11) provided that $\mathbf{A}$ is adequately designed.

5 Localization Approach

Our localization approach assumes that each robot has a unique identification (ID) both for communication and sensing. At first, we also assume object corners have distinct sensing IDs, but this assumption can be relaxed if simple heuristics based on pattern classification, such as the one presented in [6], is used to solve the problem of associating robots measurements. Our approach is centralized in the sense that each robot collects sensing information from other robots and combines this information using its own computational resources. Thus, given the previous background, localization in a network of robots can be addressed by combining a series of operations that involves transformations and combinations.

We've defined four basic operations in this paper: (**i**) a transformation from the robots measurements $z_{ij} = (\rho_{ij}, \phi_{ij})$ to the target coordinate $q_{ij} = (x_{ij}, y_{ij})$ that we represent here by a superscript t (Equation (1)); (**ii**) a combination/transformation from the robots measurements z_{ij} and z_{ji} to robot pose $q_{ij} = (x_{ij}, y_{ij}, \theta_{ij})$, that will be denoted by "$\circ$" (Equation (5) for estimating θ, (9) for combining two ρs, and (1) for coordinate transformation); (**iii**) a transformation from the measurements of one robot to another reference frame, that will be denoted by "$\vee$" (Equation (6)); and (**iv**) a combination of two measurements in the same reference frame that will be denoted by "$\wedge$" (Equation (8)).

Assuming each robot has its own sensing data and the information collected from the other robots organized in a graph similar to the one presented in Figure 2, the localization can be performed by using the previous operators in an graph searching algorithm similar to Breadth First Search (BFS). This algorithm visits, only once, all nodes of a tree by visiting all the nodes at the same depth before going deeper. Here, since we are not considering trees, the nodes can be visited more than once. Thus, if there are more than one path between the root of the graph and a specific node, all these paths will be used. Thus, the first time a vertex is visited its position is estimated. From then on, each time a node is reached its previously estimated pose is combined with the pose recently estimated using this new path.

Because we allow a node to be visited more than once, theoretically, the algorithm could enter in loops. Loops are the cause situations of interdependence where, for example, the pose of v_i can be computed using information from v_j and the pose of v_j can be computed using information from v_i. In order to avoid this problem, the original graph is transformed into a directed graph where loops are removed. The new graph is similar to a tree (however, it is not a tree) with the root being the robot chosen as the origin. Loops are avoided by removing edges between robots with the same depth. Because objects do not have measurements they never create loops and their edges are never deleted. The same occurs with unidirectional edges between two robots. A bidirectional edge between two robots of different depths may also create loops. These situations are avoided because the algorithm treats the graph as a tree and never moves towards the root. Exceptions are made with unidirectional edges that are always followed independently of the direction it points to.

Figure 4 shows an example of the derived graph and four steps of the algorithm for the situation presented in Figure 2 when R_4 is the origin. The vertices are shaded gray when their poses are estimated or upgraded. Notice how the edges between R_1 and R_2 (dotted line) were eliminated in order to avoid a loop.

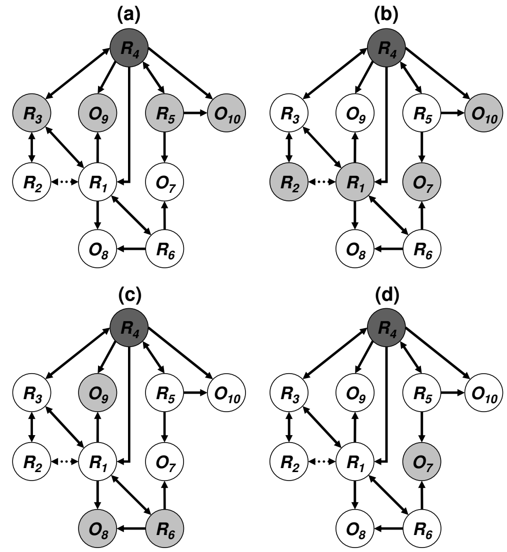

Figure 4: Four steps of the localization algorithm. All robots and objects are localized in relation to R_4. Sensor information between R_1 and R_2 (dotted link) is eliminated in order to avoid a loop. In step (a) R_3, R_5, O_9 and O_{10} are localized. In step (b) R_1, R_2 and O_7 are localized and O_{10} position is updated with information from R_5. The algorithm proceeds until all information is used or a specified graph depth is achieved.

Observe also that edge $e_{41} = (R_4, R_1)$ was used only in the second level of the graph since the absence of e_{14} prevents the computation of θ_{41}. After this variable is estimated by another path, e_{41} is used in order to improve x_{41} and y_{41} estimates.

If we use the previous algorithm to localize O_9, for example, the sensor combination is expressed using the previously defined operators as:

$$\hat{q}_{49} = z^t_{49} \wedge \left\{ \left([(z_{43} \circ z_{34}) \vee (z_{31} \circ z_{13})] \wedge z^t_{41} \right) \vee z^t_{19} \right\} .$$

This is actually the combination of R_4 and R_1 measurements, but because R_4 can not localize R_1 directly, a path passing by R_3 needed to be used.

One of the issues with the previous algorithm is that some of the edges (such as e_{12} and e_{21} in the previous example) are not used in the robots' and object's pose estimation. In order to avoid wasting useful information one could divide the algorithm in two parts assuming the robot's orientations and positions can be computed separately. This idea was previously used in [4]. The two parts are: (I) – estimation of the robot's orientations using the same algorithm; and (II) – estimation of the robots and targets positions using a linear weighted least squares method that assumes that the positions of two robots (R_i and R_j) in a third robot (R_k) coordinate frame are linearly related by:

$$\begin{aligned} x_{ki} - x_{kj} &= \rho_{ij} \cos(\theta_{ki} + \phi_{ij}) \\ y_{ki} - y_{kj} &= \rho_{ij} \sin(\theta_{ki} + \phi_{ij}) \,. \end{aligned}$$

where θ_{ki} and ϕ_{ij} are assumed to be known. The necessity of knowing θ_{ki} explains the division of the algorithm in two parts. Besides computing the orientations, the first part of the algorithm is responsible for computing the number of robots connected to the network and consequently defining the variables to be computed. Some edges are still wasted in this part of the algorithm, what is reasonable under the observation that bearing measurements tend to be much better than the range ones.

Another improvement can be done by using a dynamic Kalman Filter in order to estimate the object's position. Because the object is rigid, if a model that relates its corners is used, tracking can be performed even when some corners cannot been seen by the robots. A simple discrete model where the dimensions of the object are not necessarily known is:

$$\begin{aligned}
x_{j+1}(k+1) &= x_j(k) + vx_j\,T + d_j(k)cos(\theta_j(k)) \\
y_{j+1}(k+1) &= y_j(k) + vy_j\,T + d_j(k)sin(\theta_j(k)) \\
d_j(k+1) &= d_j(k) \\
\theta_j(k+1) &= \theta_j(k) + \omega(k) \\
vx_j(k+1) &= vx_j(k) \\
vy_j(k+1) &= vy_j(k) \\
\omega(k+1) &= \omega(k)\,,
\end{aligned}$$

where vx_j and vy_j are the velocity components of O_j, d_j and θ_j are the size and orientation of the edge between O_j and O_{j+1}, T is the sample time, and ω is the object angular velocity. This model considers that the object's velocity components are constant. Since it is not always true, during the filter project, low values must be assigned to the variables that represent the confidence level of the three last lines of model. Because we are using a nonlinear model, an Extend Kalman Filter (EKF) is necessary. The measurements used in this filter will be x and y of each object corner relative to the origin robot. Once the robots's pose are estimated, the transformation in Equations (6) and (7) are used in order to compute the vector of measurements and its covariance. Observe that the EKF introduces another step in the algorithm. Thus, the vertices of the graph relative to object corners are removed from the localization step because sensor combination is now performed by the filter.

5.1 Centralized × Distributed

While the previous centralized approach works very well for a relative small group of robots, network issues such as traffic and delays, and the lack of computational power in one single robot can pose significant problems when we are considering groups with tens or hundreds of robots. In these cases the use of distributed algorithms become mandatory. Then, we want a way to use the same algorithm and reduce both, the computation and bandwidth needed by decentralizing part of the processing.

We start observing that, in general, mobile robots only need local information in order to perform a task. Thus, if each robot collects information from its immediate neighbors and combine this data locally, it has the information it needs most of the time. In the case where a robot can listen to every robot in the group (i.e., all robots are within its communication range) it could receive all the data but, for example, localize only the robots located within a certain distance (measured by the depth in the graph) from it. In the same communication configuration, the robot could also be more selective and choose to localize only the robots that can see a given object or location.

Another way to decentralize processing, which may be useful when ad-hoc networks are used and the robots cannot talk directly to each other, is localize only the neighboring robots (robots in the communication range). In this situation the robots do not need to work as routers for communication messages and much bandwidth is preserved. If global information is necessary, a robot may ask one or more of its neighbors for the localization information (not the raw data) they have and compute the other robots' (or a specific robot) position by using transformations (6) and (7). When real-time is not a constraint, this procedure can be used recursively until full information for the whole team is available.

5.2 Global Localization

Thus far, only relative localization is considered once each robot computes the position of the others in its own reference frame. When measurements relative to a fixed frame in the world are available for the root robot, simple transformations such as those shown in Equation (6) are used in order to transform the relative estimates into world coordinates. This information can be also used in the least squares methods with minor modifications in the matrices. Notice that if complete measurements about the root robot are available (x, y and θ), this information is sufficient to localize all connected robots in the world coordinates. When partial information is available (x and y for example) global information for at least two robots is necessary.

6 Experimental Results

This section presents experimental results with our team of five mobile robots. The only sensor the robots carries is an omnidirectional camera having as field of view a circle of approximately $1.5m$ of radius. Color markers are used in order to facilitate the identification of robots and object's corners.

Figure 5 shows ground-truth data, obtained using a calibrated overhead camera, for an experiment in which one robot (R_1) moves towards a target that is localized by another robot (R_2). In order to do so, this robot needs to localize the target through information broadcasted by the other robots. The dashed circles, which represent the cameras' field of view, show that R_1 can basically see one or two robots simultaneously. In this figure we also show the estimation of R_1's motion in R_0's reference frame (which coincides with the global reference frame) and R_1's actual trajectory.

Table 1: Localization Results in three different configurations.

	Ground-Truth			Configuration 1			Configuration 2			Configuration 3		
R_i	$x\,(m)$	$y\,(m)$	$\theta\,(^\circ)$	$x\,(m)$	$y\,(m)$	$\theta\,(^\circ)$	$x\,(m)$	$y\,(m)$	$\theta\,(^\circ)$	$x\,(m)$	$y\,(m)$	$\theta\,(^\circ)$
0	0.26	0.32	68.0	0.25	0.32	70.8	0.25	0.32	71.5	0.25	0.31	71.1
1	0.99	2.04	-41.0	0.95	2.01	-44.8	1.07	1.87	-46.1	1.03	1.88	-46.6
2	1.97	1.29	125.0	1.87	1.31	122.2	1.88	1.31	122.11	1.88	1.31	122.2
3	0.17	1.48	40.0	0.12	1.48	38.9	0.12	1.48	40.7	0.12	1.48	40.7
4	1.64	0.20	-125.0	1.64	0.27	-135.7	1.63	0.24	-128.1	1.56	0.42	-94.5

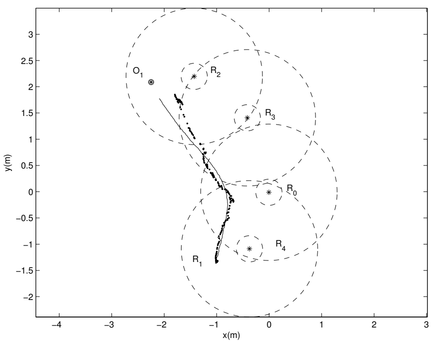

Figure 5: R_1 moves towards O_1 based on information collected by R_2 and shared through the network. The dots represent R_1's position estimated by R_0 and the continuous line the actual trajectory. The inner dashed circles represent the robot size and the outward ones represent the cameras field of view.

Figure 6 shows two snapshots of three robots tracking a triangular box using an EKF. R_0's reference frame is shown. Robots R_1 and R_2 are able to see the box corners while R_0 is only used in order to localize the other two. Even though R_0 cannot see the box corners, it is able to track the box using information from its teammates. The snapshot of the right shows that when one robot goes blind the corners covariance increases but the box is still tracked.

Our last result shows how the robots are able to be globally localized if data from an external calibrated camera is used. Figure 7 shows two images from the external camera used in order to localize the robots in the environment. Localization results in these two configurations plus an intermediary one (Configuration 2) in which R_4 can be seen by the camera, are shown in Table 1. Observe that due to the information from other robots, robots that are not seen by the external camera can still be localized.

7 Conclusions

We have presented a simple and efficient algorithm for localization and tracking in networks of robots based on basic statistical operators and graph algorithms. The algorithm uses well known statistical concepts in order to combine multi-robot information. Our algorithm was implemented and has been extensively used in our lab in a large variety of multirobot tasks that range from object manipulation [6] to sensor deployment [7]. Future work includes the application of images from an autonomous blimp equipped with GPS in order to perform outdoor localization.

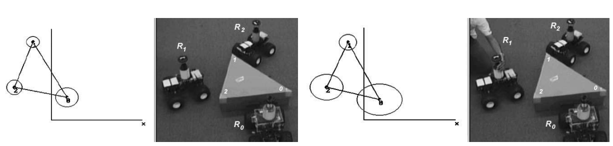

Figure 6: Two snapshots of the experiment where three robots are tracking a triangular box. The ellipses are the 3σ region of confidence.

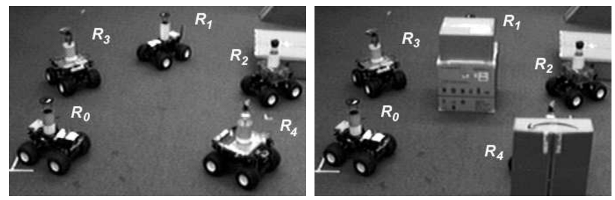

Figure 7: Configurations 1 (left) and 3 (right) which localization results are shown in Table 1.

Acknowledgments

We gratefully acknowledge the support of DARPA MARS grant NBCH1020012, AFOSR grant F49620-01-1-0382, NSF grant IIS02-22927, and CNPq-Brazil grants 200765/01-9 and 350750/1994-7.

References

[1] S. I. Roumeliotis and G. A. Bekey, "Distributed multi-robot localization," *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 781–795, October 2002.

[2] A. W. Stroupe, M. C. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 2001.

[3] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *IEEE/RSJ Int'l Conf. on Intel. Robots and Systems*, 2001.

[4] A. Das, J. Spletzer, V. Kumar, and C. Taylor, "Ad hoc networks for localization and control," in *Proc. of the IEEE Conf. on Decision and Control*, 2002.

[5] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The Int'l J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[6] G. A. S. Pereira, V. Kumar, J. Spletzer, C. J. Taylor, and M. F. M. Campos, "Cooperative transport of planar objects by multiple mobile robots using object closure," in *Experimental Robotics VIII* (B. Siciliano and P. Dario, eds.), pp. 275–284, Springer, 2002.

[7] G. A. S. Pereira, A. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Multi-Robot Systems Workshop*, 2003.