Correction of Dead-Reckoning Errors in Map Building for Mobile Robots

Matteo Golfarelli, Dario Maio, *Member, IEEE*, and Stefano Rizzi

***Abstract*—Map building is an important issue for all the applications in mobile robotics in which the environment is unknown and, in general, in order to have a robot exhibit a fully autonomous behavior. A major problem in map building is due to the imprecision of sensor measures. In this paper, we propose a technique, called elastic correction, for correcting the dead-reckoning errors made during the exploration of an environment by a robot capable of identifying landmarks. Knowledge being acquired is modeled by a relational graph whose vertices and arcs represent, respectively, landmarks and routes. Elastic correction is based on an analogy between the graph modeling the environment and a mechanical structure: the map is regarded as a truss where each route is an elastic bar and each landmark a node. Errors are corrected as a result of the deformations induced from the forces arising within the structure as inconsistent measures are taken. The elasticity parameters characterizing the structure are used to model the uncertainty on odometry. The paper presents results from simulations showing the effectiveness of the method for reducing the overall metric error and proving its robustness with reference to topological errors and to unpredictable sensor errors.**

***Index Terms*—Error correction, mobile robotics, odometry.**

I. Introduction

MOST mobile robots need a map of the environment to successfully carry out the navigational tasks assigned to them. In fact, a reliable map can be used by the robot to determine its position and to plan a path to reach its destination quickly and safely. Several techniques for environment representation have been devised in the literature, including topological maps [1], symbolic graphs labeled with metric information [2], and analogic descriptions [3]. Though in some applications a detailed map of the environment is given to the robot *a priori* [4], map building is still an important issue for all the applications in which the environment is unknown (e.g., undersea and space exploration) and, in general, in order to have a robot exhibit a fully autonomous behavior.

The problem of building an accurate map of the environment is strictly related to that of self-positioning: on the one hand, the robot needs to know its position in order to build a map; on the other, it can determine its position more easily if it has a map.

Manuscript received November 24, 1997; revised July 2, 1999 and June 26, 2000. This paper was recommended for publication by Associate Editor R. Simmons and Editor V. Lumelsky upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS'98), Victoria, BC, Canada, 1998.

The authors are with DEIS, University of Bologna, 40136 Bologna, Italy (e-mail: mgolfarelli@deis.unibo.it; dmaio@deis.unibo.it; srizzi@deis.unibo.it).

Publisher Item Identifier S 1042-296X(01)02767-7.

A. Self-Positioning

Two positioning techniques are typically used, namely *relative* and *absolute*. Relative positioning is based on odometers which estimate the current position by determining the offset from the initial position, computed by counting the revolutions of the robot wheels [5]. Absolute positioning is based either on an absolute sensor such as a GPS [6] or on the possibility of recognizing landmarks whose position in the environment is known; landmarks may be identified by processing data obtained by sensors such as sonars, lasers, cameras, etc. [7], [8].

Problems influencing the effectiveness of these techniques in the real world are caused by the imprecision of measurements, which produces metric errors. In particular, odometers typically produce both *systematic* and *nonsystematic* errors: while the former depend entirely on the characteristics of the mobile platform used [9], the latter are due to undesired interactions between the robot and the environment, such as sliding of the wheels [10]. Systematic errors can be predicted; some are deterministic (e.g., if the actual wheel diameter is smaller than the nominal one, the platform will always overestimate the distance covered), some can be modeled by a probabilistic distribution (e.g., the encoder finite resolution causes a normally distributed error). The latter, which we will call dead-reckoning errors, are inherently associated to every sensor and play a significant role in determining the global error.

B. Map Building

An autonomous robot should be capable of recognizing and correcting the errors made while building the map of an unknown environment. In [2], a technique which eliminates errors by active navigation, using a landmark-based graph to represent the environment, is proposed. In [11], a graph-like topological map is built and knowledge-derived constraints are used to circumscribe errors. The errors in building a graph-like map are classified as metric or topological in [12], and some techniques to overcome them are described. The main drawback of these approaches is that they do not take all the information previously collected into account in order to evaluate the correctness of the new measurements being acquired.

Our approach to map building, called *elastic correction* (EC), can be applied to correct the dead-reckoning errors made by a robot exploring an environment where landmarks are present. The environment is modeled by a relational graph whose vertices and arcs represent, respectively, the landmarks sensed and the interlandmark routes experienced [13]. The robot calculates the relative position of each landmark compared to the one it met immediately before, by applying dead-reckoning; when it meets

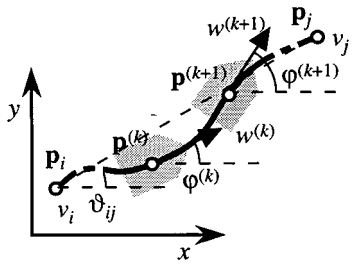


Fig. 1. The robot pose at steps k and $k+1$, while covering route r_{ij}.

a landmark it has already seen, self-positioning and error correction are achieved together by combining the new measurements collected with the knowledge accumulated so far. The original contribution of EC is to base error correction on an analogy between the graph modeling the environment and a mechanical structure: the map is regarded as a truss where each route is an elastic bar and each landmark a node. Errors are corrected as a result of the deformations induced by the forces arising within the structure as inconsistent measurements are taken. The elasticity parameters characterizing the structure model both the uncertainty on odometry and the measurements collected so far; this enables the exploitation of previously collected knowledge. Two other methods sharing this feature with EC are described in [14] and [15], and will be analyzed in Section III-E.

The paper is organized as follows. Section II reports our assumptions concerning the navigation model. Section III outlines the necessary background on the matrix methods for structural analysis and describes the EC algorithm. Section IV evaluates the results obtained by simulating the behavior of a robot platform mounting odometers and a magnetic compass, in particular it proves the robustness of EC in the presence of topological errors and magnetic fields.

II. The Navigation Model

Consider a mobile robot whose mission is to explore an unknown environment; as a minimum requirement, the robot mounts a couple of wheel encoders and is capable of identifying landmarks in the environment (for instance, it mounts a sonar array and/or a camera).

A. Self-Positioning

Let the *pose* of the robot at time step k be expressed by its position in a Cartesian plane, $\mathbf{p}^{(k)} = [x^{(k)} \quad y^{(k)}]^T$, and by its orientation $\varphi^{(k)}$. The dead-reckoning formula determines the pose at step $k+1$ as a function of the pose at step k and of the data measured from sensors at step $k+1$. In particular, if only odometry is sensed, the moduli of the linear and angular velocities, $w^{(k+1)}$ and $v^{(k+1)}$, are used

$$\begin{aligned}\mathbf{p}^{(k+1)} &= \begin{bmatrix} x^{(k)} + Tw^{(k+1)} \cos\varphi^{(k)} \\ y^{(k)} + Tw^{(k+1)} \sin\varphi^{(k)} \end{bmatrix} \\ \varphi^{(k+1)} &= \varphi^{(k)} + Tv^{(k+1)} \end{aligned} \tag{1}$$

where T is the sampling interval of sensors (see Fig. 1). On the other hand, if a compass is mounted, while $\mathbf{p}^{(k+1)}$ is still calculated as in (1), $\varphi^{(k+1)}$ may be measured directly; thus, each new positional estimate is not affected by the errors made in measuring the robot's orientation at the previous steps.

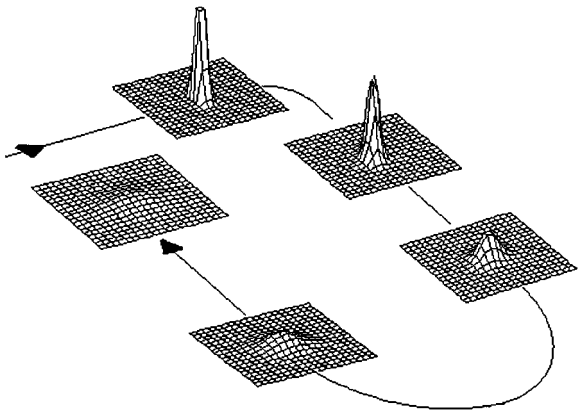


Fig. 2. Positional uncertainty in five places along a sample path.

The position of a robot calculated by dead-reckoning is affected by errors due to the inherent imprecision of sensors. In [16], an estimate of the positional uncertainty for a robot moving along a path and calculating its current position without a compass is derived. Each position calculated is associated with a density function expressing the probability that, due to errors in measurements, the robot is positioned in the surrounding area. The area in which the robot may stand with non-negligible probability is an ellipse whose shape and dimensions depend on the length and complexity of the path [17].

In the Appendix, we derive an estimate of the probability density function of the position for a robot mounting a compass. While from a theoretical point of view every density function could be used, previous experience [14], [18], [16] shows that the errors can be effectively modeled using a Gaussian distribution with null mean [3]. In this case, the uncertainty on the robot position is distributed around the position calculated at step k, $\mathbf{p}^{(k)}$, according to the following normal density function:

$$\begin{aligned}\delta(x, y, \mathbf{C}^{(k)}) = &\frac{1}{2\pi|\mathbf{C}^{(k)}|^{(1/2)}} \\ &\cdot \exp\left(-\frac{1}{2}[x - x^{(k)} \quad y - y^{(k)}]\right. \\ &\left. \cdot (\mathbf{C}^{(k)})^{-1} \begin{bmatrix} x - x^{(k)} \\ y - y^{(k)} \end{bmatrix}\right) \end{aligned} \tag{2}$$

where $\mathbf{C}^{(k)}$ is the covariance matrix calculated recursively, as shown in (24).

Fig. 2 shows how positional uncertainty evolves while a robot equipped with a compass is moving along a path.

B. Map Building

The map built by the robot is structured as a nondirected graph $\mathcal{M} = (V, R)$. Each vertex $v_i \in V$ represents a landmark sensed and is labeled with its estimated position, $\mathbf{p}_i = [x_i \quad y_i]^T$. Arc $r_{ij} \in R$ represents the route connecting v_i and v_j and is labeled with the number of times the route has been covered so far, t_{ij}, and with a covariance matrix $\mathbf{C}_{ij}$ expressing the uncertainty on r_{ij}. We will denote by $\vartheta_{ij} = \arctan(y_j - y_i)/(x_j - x_i)$ (*route orientation*, $0 \le \vartheta_{ij} < \pi$) the absolute orientation of

the segment connecting v_i to v_j, and by s_{ij} (*route stretch*) the Euclidean distance between $\mathbf{p}_i$ and $\mathbf{p}_j$.

The very first landmark met within the environment is assumed to be in position $\mathbf{p}_0 = \mathbf{0}$. Every time the robot covers route r_{ij} to reach v_j from v_i, a new estimate for the position of v_j is calculated by dead-reckoning, starting from the position estimated for v_i (see Fig. 1). The covariance matrix expressing the uncertainty on the new estimate is determined as shown in the Appendix; its value depends on the length and the winding of the route covered. Finally, $\mathbf{C}_{ij}$ is calculated as the average of the t_{ij} covariance matrices determined so far.

Apparently, this approach to map building requires the robot to be capable of recognizing exactly the landmarks it has already met. Depending on the nature of landmarks, recognizing them univocally may be unfeasible; most probably, the robot will only be capable of distinguishing between a limited number of landmark categories. On the other hand, the estimate of positional uncertainty we associate to each pose may have a determinant role in achieving univocal identification of landmarks. When the robot reaches a landmark v, it searches its map $\mathcal{M}$ to see if it has already met a landmark of the same category whose current positional estimate in $\mathcal{M}$ falls within the area where the robot estimates v to be with non-negligible probability; as shown in Section II-A, this area is an ellipse centered in the position estimated for v. If a matching landmark is found, then v is recognized; otherwise, it is assumed to be a new landmark. Throughout the next section, we will assume that perfect landmark identification is achieved. Of course, if two or more nondistinguishable landmarks were very close to each other, they would be erroneously recognized as a single landmark; in Section IV-B we will discuss the robustness of EC with reference to these topological errors.

III. Elastic Correction

Elastic correction is applied in two different ways when a route is covered for the first time (*first-sight correction*) and when it is covered again (*refinement correction*). After introducing the necessary background on structural analysis in Section III-A, in Section III-B we define the mathematical model underlying our approach; in Sections III-C and III-D we describe, respectively, first-sight and refinement correction. In Section III-E we draw a comparison between EC and the techniques proposed in [14] and [15].

A. Background on Matrix Methods for Structural Analysis

Let an elastic structure which can be modeled as an assemblage of members connected at node points be given. The problem of determining the displacements of the nodes when one or more of them are loaded with a force can be solved by applying the *stiffness method* [19].

Let n be the number of nodes in an elastic bidimensional structure; the relationships linking the displacements of the nodes to the forces applied to them are expressed in matrix form by the *stiffness equation*

$$\mathbf{f} = \mathbf{K}\mathbf{u} \tag{3}$$

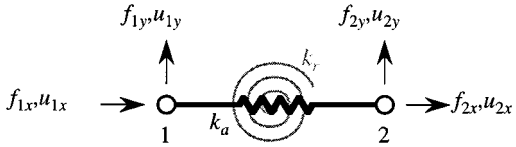


Fig. 3. Truss basic element, including a linear elastic spring (in black) and a rotational elastic spring (in gray).

where

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \cdots \\ \mathbf{f}_n \end{bmatrix} = \begin{bmatrix} f_{1x} \\ f_{1y} \\ \cdots \\ f_{nx} \\ f_{ny} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \cdots \\ \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ \cdots \\ u_{nx} \\ u_{ny} \end{bmatrix}$$

are, respectively, the column matrix of x- and y-components of the nodal forces and of the nodal displacements; $\mathbf{K}$ is a symmetric square matrix with rank $2n$ (*stiffness matrix*), whose elements are the stiffness coefficients of the structure.

The structures we are interested in analyzing are pin-jointed trusses whose elements can be modeled by combining a linear elastic spring and a rotational elastic spring, as shown in Fig. 3. As reported in [19], the stiffness matrix $\mathbf{K}$ associated to a single truss element of length s and directed along the x axis is

$$\mathbf{K} = \begin{bmatrix} k_a & 0 & -k_a & 0 \\ 0 & \dfrac{k_r}{s^2} & 0 & -\dfrac{k_r}{s^2} \\ -k_a & 0 & k_a & 0 \\ 0 & -\dfrac{k_r}{s^2} & 0 & \dfrac{k_r}{s^2} \end{bmatrix} \tag{4}$$

where k_a and k_r are, respectively, the spring constants for the linear and the rotational springs. In the general case in which the truss element forms an angle ϑ with the x axis, its stiffness matrix can be obtained from the stiffness matrix $\mathbf{K}$ of an element with the same spring constants but directed along the x axis, by rotating $\mathbf{K}$ by ϑ. The stiffness matrix for a complex structure is calculated by superimposing the stiffness matrices of the single elements belonging to the structure. Each member matrix is expanded to the order of the total structure matrix; this is done by adding columns and rows of zeros for the nodal displacements which are irrelevant for the member in question. The expanded member matrices are then summed together.

Node i is said to be *constrained* if its displacement $\mathbf{u}_i$ is known and assigned *a priori*, *free* otherwise; a constrained node for which $\mathbf{u}_i = \mathbf{0}$ is said to be *hinged*. In order to solve the stiffness equation, the n nodes are partitioned into two sets α and β which include, respectively, the free and the constrained nodes. The relationship between displacements and forces can thus be rewritten as

$$\begin{bmatrix} \mathbf{f}_\alpha \\ \mathbf{f}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{\alpha\alpha} & \mathbf{K}_{\alpha\beta} \\ \mathbf{K}_{\beta\alpha} & \mathbf{K}_{\beta\beta} \end{bmatrix} \begin{bmatrix} \mathbf{u}_\alpha \\ \mathbf{u}_\beta \end{bmatrix} \tag{5}$$

where $\mathbf{f}_\alpha$ and $\mathbf{f}_\beta$ represent, respectively, the loads applied to the free nodes and the reactions in the constrained nodes. Unknown nodal displacements are obtained by inverting $\mathbf{K}_{\alpha\alpha}$ and solving a linear system with degree twice the number of free nodes

$$\mathbf{u}_\alpha = \mathbf{K}_{\alpha\alpha}^{-1}(\mathbf{f}_\alpha - \mathbf{K}_{\alpha\beta}\mathbf{u}_\beta). \tag{6}$$

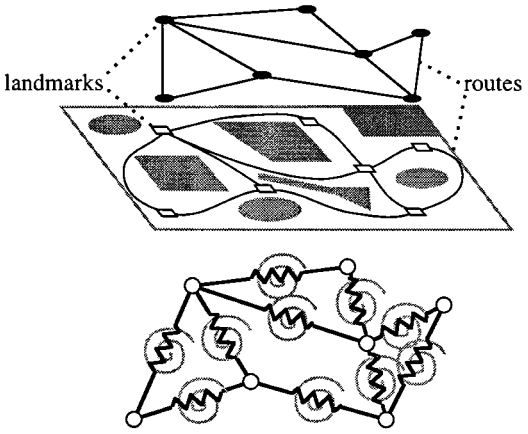


Fig. 4. Graph-based representation of an environment (top) and equivalent truss (bottom).

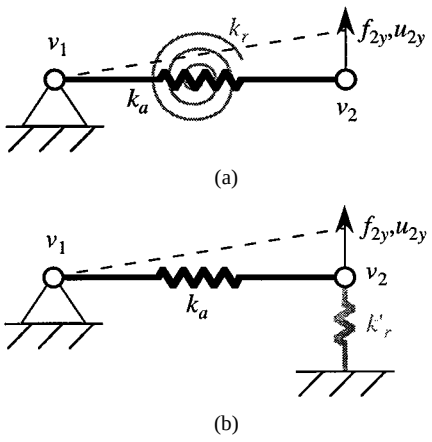


Fig. 5. Equivalence between (a) an element including a rotational spring and (b) one including a linear spring. The triangular symbol denotes a hinged node.

Unknown reactions are calculated by substituting (6) into (5)

$$\mathbf{f}_\beta = \mathbf{K}_{\beta\alpha}\mathbf{K}_{\alpha\alpha}^{-1}\mathbf{f}_\alpha - (\mathbf{K}_{\beta\alpha}\mathbf{K}_{\alpha\alpha}^{-1}\mathbf{K}_{\alpha\beta} - \mathbf{K}_{\beta\beta})\mathbf{u}_\beta. \quad (7)$$

Further details on the stiffness method can be found in [19].

B. Environment Modeling

Elastic correction is based on the analogy between the environment map $\mathcal{M}$ and a pin-jointed truss whose elements and nodes represent, respectively, routes and landmarks (see Fig. 4); the parameters defining the stiffness of each element when loaded sum up the characteristics of the corresponding route. The more elastic an element, the greater the change in length and orientation that it will experience when loaded; thus, stiffness should be proportional to the certainty on the stretch and orientation of the corresponding route.

An element representing a route r with stretch s and orientation ϑ may be thought of as a bar long s, oriented according to ϑ and behaving as follows.

- It can be compressed elastically along its axis to model uncertainty on the route stretch.
- It can neither be bent nor twisted.
- It can rotate elastically to model uncertainty on the route orientation.

A bar with these characteristics can be modeled by combining a linear axial spring and a rotational spring (see Fig. 3) whose spring constants k_a and k_r must be defined in function of the positional uncertainty of the robot.

For simplicity, we consider a bar representing a route r oriented along the x axis ($\vartheta = 0$). Since k_a expresses an axial deformation along x, it is reasonable to define it in terms of the average error Δx made by the robot in determining the x coordinate of the other end of r, i.e., the average error made on the stretch of r: $k_a \propto 1/\Delta x$, where

$$\Delta x = \iint_{-\infty}^{\infty} \delta(x, y, \mathbf{C})|x|\, dx\, dy = \sqrt{\frac{2}{\pi} C_{xx}} \quad (8)$$

and δ is the density function as defined in (2), $\mathbf{C}$ is the covariance matrix associated to r, and C_{xx} is its upper-left element.

Similarly, it is possible to calculate the average error Δy made by the robot in determining the y coordinate of the other end of r. Unfortunately, k_r does not express an axial deformation along y; in order to associate k_r to Δy, it is necessary to determine the spring constant k'_r of a linear spring, which under certain conditions may be equivalent to the rotational one. When the robot goes from landmark v_1 to landmark v_2, it evaluates the positional uncertainty for v_2 by assuming that the position estimated for v_1 is correct. Within our mechanical analogy, this corresponds to considering the structure shown in Fig. 5(a), ruled by the following equation:

$$f_{2y} = \frac{k_r}{s^2}\, u_{2y} \quad (9)$$

where s is the length of the element. This structure is equivalent to that shown in Fig. 5(b), ruled by

$$f_{2y} = k'_r u_{2y} \quad (10)$$

if and only if $k_r = s^2 k'_r$. Thus, consistently with what occurred for the x axis, we may assume $k'_r \propto 1/\Delta y$, where

$$\Delta y = \iint_{-\infty}^{\infty} \delta(x, y, \mathbf{C})|y|\, dx\, dy = \sqrt{\frac{2}{\pi} C_{yy}}$$
$$\Rightarrow k_r \propto \frac{s^2}{\Delta y} \quad (11)$$

and C_{yy} is the lower-right element of $\mathbf{C}$.

The certainty on the stretch and orientation of r also depends on the number t of times r has been covered; in fact, the higher t, the higher the amount of data concerning r collected. Thus, from (4) and (11), the stiffness matrix for a bar representing a route r oriented along the x axis turns out to be

$$\mathbf{K} = \gamma \cdot \begin{bmatrix} \frac{t}{\Delta x} & 0 & -\frac{t}{\Delta x} & 0 \\ 0 & \frac{t}{\Delta y} & 0 & -\frac{t}{\Delta y} \\ -\frac{t}{\Delta x} & 0 & \frac{t}{\Delta x} & 0 \\ 0 & -\frac{t}{\Delta y} & 0 & \frac{t}{\Delta y} \end{bmatrix} \quad (12)$$

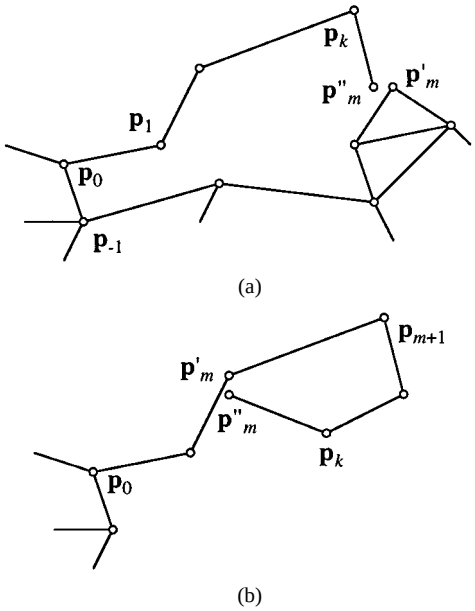


Fig. 6. Error correction on (a) open and (b) closed polygons.

where γ is a proportionality constant which we will assume equal to 1.

The stiffness matrix for a bar representing a route r oriented along direction ϑ is obtained as follows.

1) by rotating by $-\vartheta$ the covariance matrix $\mathbf{C}$;
2) by calculating Δx and Δy by (8) and (11);
3) by building the stiffness matrix $\mathbf{K}$ by (12);
4) by rotating $\mathbf{K}$ by ϑ.

Once the stiffness matrices for all the routes experienced have been determined, the stiffness matrix for the whole structure is assembled as described in Section III-A.

C. First-Sight Correction

Suppose the robot is exploring an unknown area starting from a known landmark v_0. It meets the unknown landmarks $v_1, \ldots, v_k$ by covering a sequence of unknown routes, and finally reaches a landmark v_m which it has already met. When v_m is reached, the new positional estimate $\mathbf{p}''_m$ computed by dead-reckoning may be compared to the previous one $\mathbf{p}'_m$; due to sensor errors, the two estimates will certainly differ. Let

$$\overline{\mathbf{u}}_m = \mathbf{p}''_m - \mathbf{p}'_m. \tag{13}$$

If v_m has been met before v_0, the segments orderly connecting $\mathbf{p}_0$ to $\mathbf{p}''_m$ form an open polygon [see Fig. 6(a)]. Otherwise, the segments orderly connecting $\mathbf{p}'_m$ to $\mathbf{p}''_m$ form a closed polygon [see Fig. 6(b)]. The graph representing the environment should be metrically consistent at each time during exploration, hence, the two positional estimates for v_m must be forced to exactly agree; we assume the error on the stretch of each route to be proportional to the positional uncertainty induced by that route.

For closed polygons, according to the mechanical model described in Section III-B, $\mathbf{p}'_m$ should be hinged and a displacement $-\overline{\mathbf{u}}_m$ should be applied to $\mathbf{p}''_m$; thus, the constrained nodes are $\beta = \{\mathbf{p}'_m, \mathbf{p}''_m\}$. This displacement moves $\mathbf{p}''_m$ on $\mathbf{p}'_m$ restoring the metric consistency of the graph; the displacements $\overline{\mathbf{u}}_\alpha$ calculated for the free nodes ($\alpha = \{\mathbf{p}_{m+1}, \ldots, \mathbf{p}_k\}$) determine the new positions for the landmarks and the new stretches and orientations for the routes in the polygon. Since no external loads are applied, the unknown nodal displacements may be calculated by (6) with $\mathbf{f}_\alpha = \mathbf{0}$

$$\overline{\mathbf{u}}_\alpha = -\mathbf{K}^{-1}_{\alpha\alpha}\mathbf{K}_{\alpha\beta}\overline{\mathbf{u}}_\beta \tag{14}$$

where $\overline{\mathbf{u}}_\beta = [\mathbf{0} \quad -\overline{\mathbf{u}}_m]^T$.

While for closed polygons this approach is correct since metric inconsistency is only due to the errors made by the robot while covering the path starting and ending in v_m, and correction affects the corresponding set of routes, the same is not true for open polygons. In fact, for example, in Fig. 6(a) the path $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}''_m$ could be mostly correct while the positional inconsistency could derive mainly from the path $\mathbf{p}'_m, \ldots, \mathbf{p}_{-1}, \mathbf{p}_0$. Taking also the errors made outside the polygon into account implies considering a surrounding area determined by the set of routes connecting the η landmarks nearest to v_m. The structure we define to model the problem includes all the landmarks belonging to this area; among these, the landmarks lying on the external border are hinged. Let α and β be the sets of free and constrained nodes, respectively.

Restoring metric consistency requires both $\mathbf{p}'_m$ and $\mathbf{p}''_m$ (belonging to α) to be moved to the same position $\overline{\mathbf{p}}_m$. In order to accomplish this, an infinite number of couples of forces $\mathbf{f}'$ and $\mathbf{f}''$, to be applied respectively to $\mathbf{p}'_m$ and $\mathbf{p}''_m$, could be used. All these couples satisfy the following linear system:

$$\mathbf{A}\,\mathbf{f} = -\overline{\mathbf{u}}_m \tag{15}$$

where $\overline{\mathbf{u}}_m = \mathbf{p}''_m - \mathbf{p}'_m$, $\mathbf{f} = [\mathbf{f}' \quad \mathbf{f}'']^T$, and $\mathbf{A}$ is a (2×4) matrix obtained from $\mathbf{K}^{-1}_{\alpha\alpha}$ assuming that the only forces applied to the truss are $\mathbf{f}'$ and $\mathbf{f}''$. Within this infinite set, we choose the couple $\overline{\mathbf{f}'}$ and $\overline{\mathbf{f}''}$ with minimum magnitude, which produces the minimum truss deformation.

The problem of determining $\overline{\mathbf{f}'}$ and $\overline{\mathbf{f}''}$ can be formulated as a constrained optimization problem where the function to be minimized is $\|\mathbf{f}\|_2^2$ while the constraint is expressed by (15). This problem can be solved using the Lagrangian method [20]

$$L(\mathbf{f}, \lambda) = \mathbf{f}^T\mathbf{f} - (\mathbf{A}\mathbf{f} + \overline{\mathbf{u}}_m)^T\lambda \tag{16}$$

determining the linear system

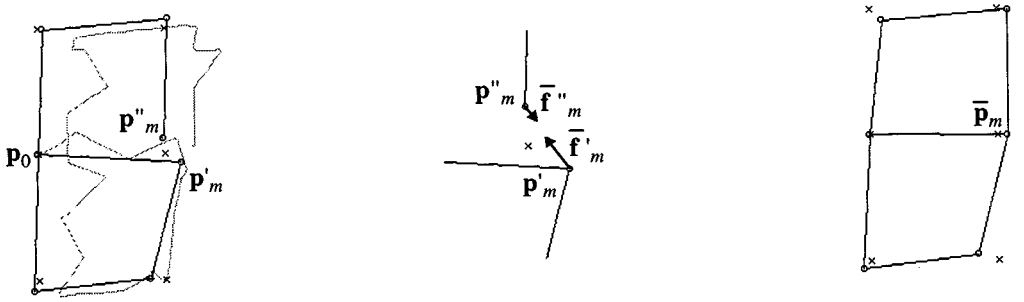
$$\begin{cases} 2\mathbf{f} - \mathbf{A}^T\lambda = \mathbf{0} \\ \mathbf{f}^T\mathbf{A}^T + \overline{\mathbf{u}}_m^T = \mathbf{0} \end{cases} \tag{17}$$

that admits an analytic solution.

Fig. 7 shows the error made by the robot while covering an open polygon, how the inconsistency is solved and the corresponding forces. In general, the two forces have both different magnitudes and different directions, due to the truss structure which induces different constraints on $\mathbf{p}'_m$ and $\mathbf{p}''_m$.

D. Refinement Correction

Every time the robot covers a route r_{ij} it has already covered, it can exploit the new information acquired to improve the current estimate of the stretch and orientation of r_{ij} and, thus, that of the positions of its end landmarks v_i and v_j. Reasonably, the estimates for r_{ij} should be equal, at each time, to the average



(a) (b) (c)

Fig. 7. Open polygon correction on a rectangular mesh. Crosses represent the true positions of vertices. (a) The measured path (in gray) and the measured graph (in black). (b) The forces applied to solve the polygon. (c) The corrected graph.

of the data measured so far. Let s_{ij} and ϑ_{ij} be the current estimates for the stretch and orientation, and $\tilde{s}_{ij}$ and $\tilde{\vartheta}_{ij}$ be the new values measured. The new estimates are calculated as

$$s'_{ij} = \frac{t_{ij}s_{ij} + \tilde{s}_{ij}}{t_{ij}+1}$$

$$\vartheta'_{ij} = \begin{cases} \dfrac{t_{ij}\vartheta_{ij} + \tilde{\vartheta}_{ij}}{t_{ij}+1}, & \text{if } |\vartheta_{ij} - \tilde{\vartheta}_{ij}| \le \dfrac{\pi}{2} \\ \dfrac{t_{ij}\vartheta_{ij} + \tilde{\vartheta}_{ij} - \pi}{t_{ij}+1}, & \text{if } |\vartheta_{ij} - \tilde{\vartheta}_{ij}| > \dfrac{\pi}{2} \wedge \vartheta_{ij} < \tilde{\vartheta}_{ij} \\ \dfrac{t_{ij}\vartheta_{ij} + \tilde{\vartheta}_{ij} - t_{ij}\pi}{t_{ij}+1}, & \text{if } |\vartheta_{ij} - \tilde{\vartheta}_{ij}| > \dfrac{\pi}{2} \wedge \vartheta_{ij} > \tilde{\vartheta}_{ij}. \end{cases} \tag{18}$$

The desired displacements for v_i and v_j are calculated by imagining rotating r_{ij} around its midpoint in order to let it assume the new stretch and orientation

$$\overline{\mathbf{u}}_i = -\overline{\mathbf{u}}_j = \begin{bmatrix} \dfrac{s'_{ij}}{2}\cos\vartheta'_{ij} - \dfrac{s_{ij}}{2}\cos\vartheta_{ij} \\ \dfrac{s'_{ij}}{2}\sin\vartheta'_{ij} - \dfrac{s_{ij}}{2}\sin\vartheta_{ij} \end{bmatrix}. \tag{19}$$

This solution, however, is not satisfactory since it does not take into account all the knowledge of the environment collected so far. Using global knowledge to correct the error on a single route is essential when the certainty on the routes is not evenly distributed. In fact, if the certainty on the last route experienced is low and comparable to that on the other routes, overall deformation should be significantly determined by the new measurement acquired. On the other hand, if the certainty on the other routes is much higher, the new measurement taken should only weakly affect the graph, which is mainly determined by the other routes.

Another issue arising when correcting the error on a route is how metric consistency for the graph representing the environment is maintained. In fact, correcting the stretch and orientation of r_{ij} implies modifying the stretches and orientations of the adjacent routes. Nevertheless, putting all the correction on the adjacent routes does not appear to be sound.

The mechanical model defined in Section III-B allows both issues to be dealt with. Let $\mathcal{M}$ be the graph representing the environment, and $r_{ij} \in \mathcal{M}$ be the last route experienced. Our approach consists of the following two phases.

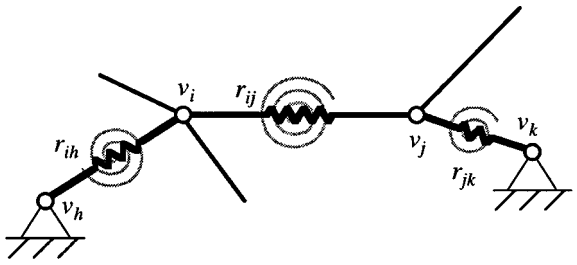


Fig. 8. Reference structure for calculating the forces to be applied.

1) Let SR_i, SR_j be the sets of routes (not including r_{ij}) entering landmarks v_i and v_j, respectively. In the first phase, the forces producing the desired displacements $\overline{\mathbf{u}}_i$ and $\overline{\mathbf{u}}_j$ on the ends of r_{ij} are calculated on a reference structure including r_{ij} and the two routes, r_{ih} and r_{jk}, that have maximum stiffness within SR_i and SR_j, respectively; both r_{ih} and r_{jk} are hinged in the vertex not shared with r_{ij} (see Fig. 8). Since all the nodes are constrained, (7) with $\alpha = \emptyset$ is used to calculate forces $\overline{\mathbf{f}}_\beta$

$$\overline{\mathbf{f}}_\beta = \mathbf{K}_{\beta\beta}\overline{\mathbf{u}}_\beta \tag{20}$$

where $\overline{\mathbf{u}}_\beta = [\overline{\mathbf{u}}_i \quad \overline{\mathbf{u}}_j \quad \mathbf{0} \quad \mathbf{0}]^T$.

2) Let SL be the set of the η landmarks nearest to the midpoint of r_{ij}. In the second phase, the forces previously calculated for v_i and v_j, $\overline{\mathbf{f}}_i$ and $\overline{\mathbf{f}}_j$, are applied to the same two vertices, but within a larger structure consisting of the set SR of routes which includes r_{ij} and all the other routes involving at least one landmark in SL. All the vertices in $\beta = \{v_w \notin SL \mid \exists r_{wz} \in SR\}$ are hinged; those in $\alpha = SL$ are free. If β turns out to be empty (i.e., less than η landmarks have been visited so far), at least one landmark must be hinged in order to avoid rigid rotations of the whole structure; the most distant landmark is chosen. Finally, (6) with $\mathbf{u}_\beta = \mathbf{0}$ is used to calculate the unknown displacements

$$\overline{\mathbf{u}}_\alpha = \mathbf{K}_{\alpha\alpha}^{-1}\overline{\mathbf{f}}_\alpha \tag{21}$$

where $\overline{\mathbf{f}}_\alpha = [\overline{\mathbf{f}}_i \quad \overline{\mathbf{f}}_j \quad \mathbf{0} \quad \ldots \quad \mathbf{0}]^T$.

In a few words, our approach heuristically restricts refinement correction to a local area, containing η landmarks, around the reobserved route r_{ij}. Choosing a value η for the number of free nodes in the structure entails determining an upper bound on the computational complexity of each correction; conversely,

the lower η is, the less effective error correction is. In the tests shown in Section IV we used $\eta = 50$.

E. Related Literature

Among the error correction techniques in the literature, those proposed in [15], [14], and [21] (which we will call, respectively, TH, ML, and HE) share with EC the capability of taking all previously collected measurements into account and, at the same time, that of using locally acquired knowledge to improve positional estimates in the rest of the map.

In TH, the map building problem is formulated as a constrained, probabilistic maximum-likelihood estimation problem. An *expectation step* and a *maximization step* are alternated. In the first, the current best map and the data collected so far are used to compute a probabilistic estimate for the robot position, while in the second the robot computes the most likely map based on the probabilities computed in the expectation step. While drawing a formal comparison between EC and TH is very difficult, it is worth discussing some similarities and differences. The hypotheses made about landmark recognition and error modeling are substantially equivalent. While in EC metric knowledge and uncertainty are condensed, respectively, in the geometric and mechanical characteristics of the truss, TH maintains a discrete grid storing probability of having a landmark in each map position. Thus, in EC the complexity is lower since it is proportional to the number of landmarks, while in TH it depends on the grid size. Finally, when a new observation takes place, the positions of the other landmarks change, in EC, due to the forces propagated across the structure, and in TH, by propagating backward in time the positional uncertainty.

ML shares with EC the idea of posing constraints between couples of nodes; in particular, it operates on a graph-based representation of the poses assumed by the robot during navigation. The graph metrics is estimated by minimizing an *energy* function whose definition is based on the relationships (*constraints*) between couples of poses. As a matter of fact, solving the stiffness equation (3) in EC can be seen as minimizing a different energy function, namely the elastic energy of the truss. Besides, due to the sensor model adopted in ML, two types of constraints are considered: those determined by odometry (*weak constraints*) and those determined by matching two range scans of the same object (*strong constraints*). The sensor model we assumed in EC entails, besides the constraints on odometry (modeled as in ML), the constraints arising from landmark recognition. A quantitative comparison between EC and ML is drawn in Section IV-A.

Both EC and ML can be classified as relation-based approaches meaning that the map uncertainty is expressed by uncertain relationships between objects; on the other hand, also location-based approaches have been widely investigated. In particular, in HE the locations of objects and of the robot are considered as state variables, and the object variance/covariance matrices are represented as state information fusing multiple observations by means of an extended Kalman filter. The authors emphasize that, in order to avoid the uncertainty to be underestimated, correlation between objects must be maintained and propagated during exploration. This task is very hard in HE, since covariance matrices for object poses must be explicitly updated. On the contrary, EC does not require the update to the covariance matrices to be propagated since the uncertainty on the object poses is not explicitly represented but is derived by "composing" the route uncertainties.

The other correction methods proposed in the literature can hardly be quantitatively compared with EC, since they are either based on assumptions and sensory equipment radically different from ours [12] or were devised within a different framework [22]. For example, in [23], the authors present an algorithm based on an adaptive place network (APN) containing both metric and topological information about the structure of the environment. The APN uses no landmarks to represent the objects in the environment; it is instead made up of place units, each corresponding to a region of Cartesian space, and place links, representing the relationship between adjacent place units. Corrections are local to the last covered route and consist in a change of the link confidence; while calculating the new confidence value, the algorithm does not take any kind of information about the surrounding routes into account. In [24], the authors propose a technique to learn metric information about the environment, which requires *a priori* topological knowledge of the environment.

IV. Experimental Results

We have extensively tested EC in simulation in order to evaluate its effectiveness and robustness. The sensorial model on which most tests are based was derived by a Pioneer I robot, which mounts a fluxgate compass. In particular, the sensory covariance matrix $\mathbf{M}$ was measured experimentally; the (absolute) average odometric and compass errors turned out to be, respectively, 5% of the distance measured and 0.03 rad. The covariance matrix $\mathbf{C}$ adopted for each route was then obtained as shown in (24).

We estimate the error on the map metric by two measurements: the average percentage error on the stretch of the routes

$$\sigma = \frac{1}{n} \sum_{r_{ij} \in R} \frac{|\hat{s}_{ij} - s_{ij}|}{\hat{s}_{ij}}$$

(where $\hat{s}_{ij}$ and s_{ij} are, respectively, the true and the estimated values for the stretch of r_{ij} and n is the number of routes in the map) and the average error on the orientation of the routes

$$\rho = \frac{1}{n} \sum_{r_{ij} \in R} \min\Big\{ \Big| \hat{\vartheta}_{ij} - \vartheta_{ij} \Big|, \min\{\hat{\vartheta}_{ij}, \vartheta_{ij}\} - \max\{\hat{\vartheta}_{ij}, \vartheta_{ij}\} + \pi \Big\}.$$

A. Effectiveness

Fig. 9 shows the result of first-sight correction on a square-meshed map including 100 landmarks and 180 routes; in order to emphasize the effects of correction, higher sensor errors were simulated (9% average odometric error, 0.09 rad compass errors). This phase is primarily aimed at eliminating the metric

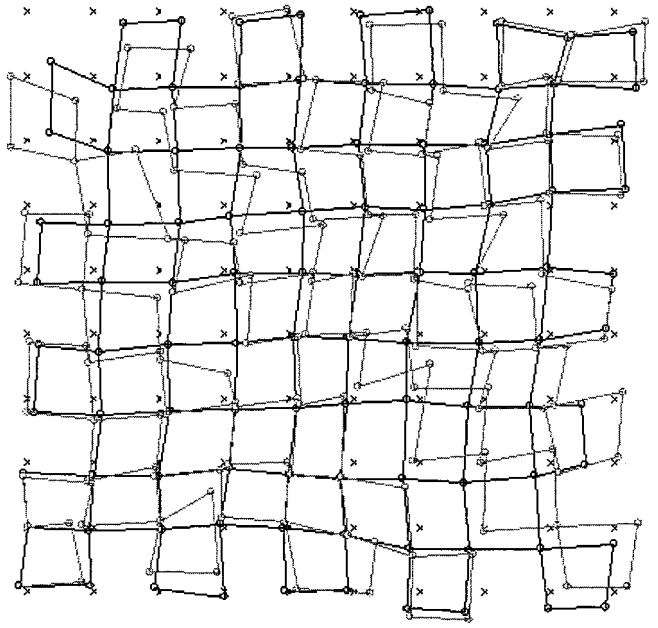


Fig. 9. First-sight correction. The measured (inconsistent) map is gray, the corrected one is black.

inconsistencies due to sensor errors; the resulting map is consistent but still affected by a significant error: σ drops from 9.5% to 7.9%, ρ from 0.098 to 0.078 rad.

The other tests described in this section were made on an irregular heterogeneous map which simulates a real-world environment (190 landmarks, 445 routes). Fig. 10 shows the result of EC when the map is toured several times; it is remarkable that σ and ρ are already reduced by half after four complete tours of the map.

In order to prove the effectiveness of our technique, we compared it with ML, the sensory models being equal. The assumptions on odometry are the same, thus the constraints on odometry in our adaptation of ML were formulated as in [14]. Our sensory model does not include laser scans, but assumes that the robot can determine exactly the relative landmark positions; hence, in our adaptation of ML, strong links are never formulated on couples of unrelated poses, but only on couples of different poses for the same landmark. We performed the tests on the map in Fig. 10 and averaged the errors on ten trials; Fig. 11 reports the average error and the standard deviation computed on all the routes. It clearly appears how, on adopting EC, the average errors on route stretch and orientation are significantly decreased as the map is repeatedly toured. On the contrary, after ten tours, the average errors on adopting ML are not significantly reduced; when few strong links are present (first 2–3 tours), ML does not succeed in effectively counterbalancing the cumulated odometric errors. This result can be interpreted by arguing that, under our sensory model, energy formulation in EC is more effective than in ML.

Both error measurements used so far are referred to routes, that is, to the relative positioning of landmarks, which we believe to be the most relevant issue from the point of view of planning and executing navigational tasks. Nevertheless, we also considered the absolute positioning of landmarks; Fig. 12 shows how the average error on the landmark positions, as compared

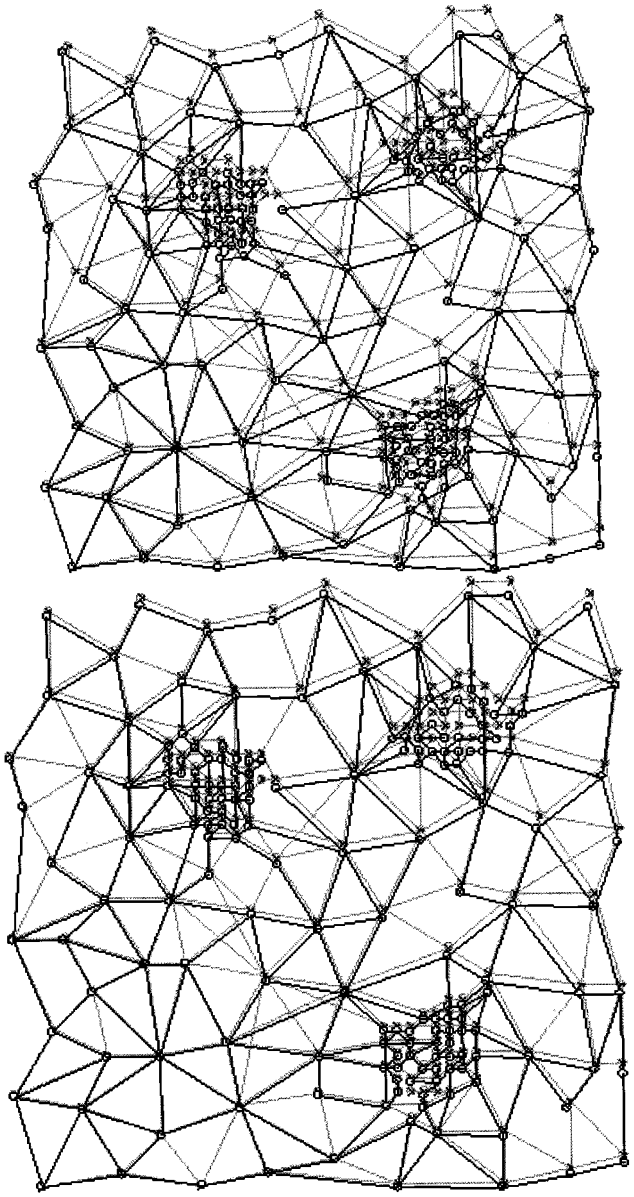


Fig. 10. Correction after 1 tour (top) and after 10 tours (bottom). The true maps are gray, the corrected ones are black.

to the one measured before correction, reduces to 50% after the first two tours.

B. Robustness

In order to evaluate the robustness of EC, we take into account two kinds of problems which commonly occur in real-world environments: topological errors and magnetic fields.

EC takes only metric errors into account. Unfortunately, the map building process may be heavily affected by topological errors. Due to their relevance several works discuss this issue (for instance, see [15]). Although extending EC in order to correct topological errors is beyond the scope of this paper, we conducted some tests aimed at showing how robust EC is to them. Within our graph-based framework, topological errors arise when the following occurs.

1) The robot misses a known or unknown landmark. The topological consistency of the map decreases since a

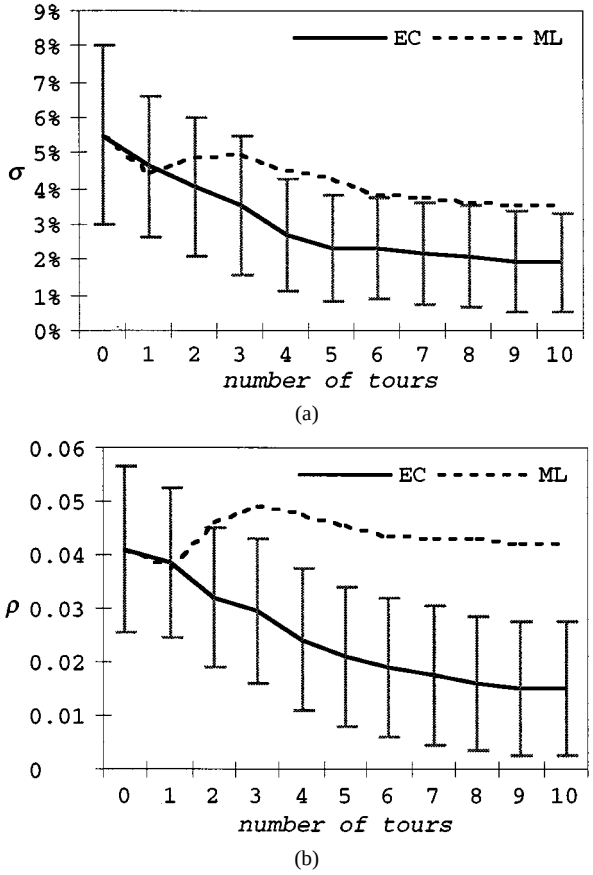


Fig. 11. Errors on (a) route stretch and (b) orientation in function of the number of exploration tours for EC and ML.

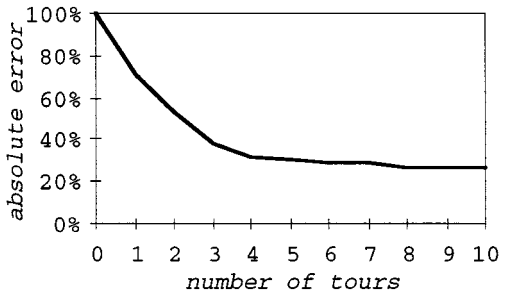


Fig. 12. Absolute landmark positioning error.

"dummy" route is created. The effectiveness of EC is reduced due to the higher positional uncertainty.

2) The robot mistakes a known or unknown landmark for a different known one. EC is applied starting from wrong premises.
3) The robot mistakes a known or unknown landmark for an unknown one. The topological consistency decreases since a dummy route and a dummy landmark are created; EC is not applied.

We will not consider case 3 since it does not affect metric consistency.

As to errors of type 1, their frequency ν does not have any significant impact on the effectiveness of EC. In fact, even if several dummy routes are created, metric consistency is not relevantly affected. For the map in Fig. 10, if ν increases from 0 to 0.2, σ increases from 2.7% to 2.9% while ρ keeps constant at 0.021.

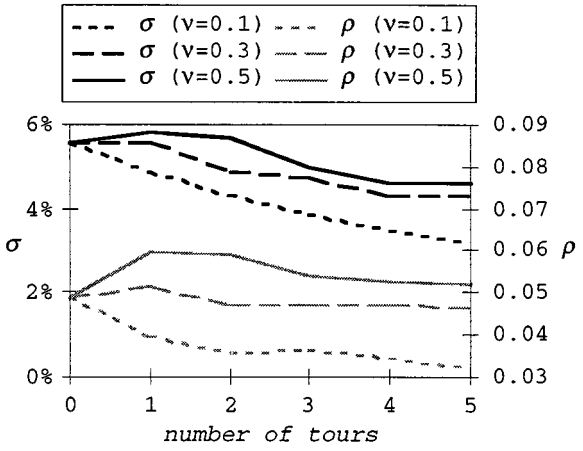


Fig. 13. Errors on route stretch and orientation in presence of topological error.

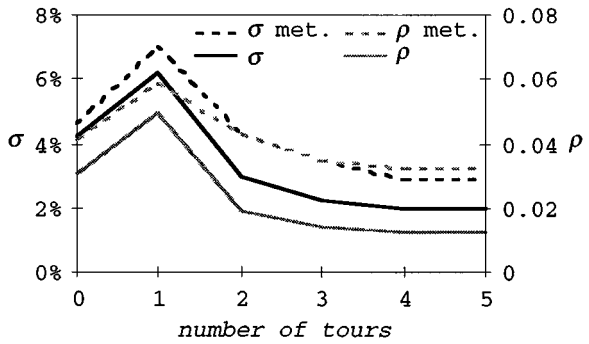


Fig. 14. Errors on route stretch and orientation in presence and in absence of metallic objects.

Fig. 13 shows how metric errors are corrected for different frequencies of type-2 errors. The negative impact of topological errors on metric consistency is reduced by EC since all the topological relationships between the mistaken landmark and the neighboring ones are considered, constraining map deformation.

We close this section by discussing the robustness of EC with reference to magnetic fields induced by big metallic objects (heaters, cabinets, etc.), which may significantly alter the compass readings within a few meters range. We modeled the magnetic field induced by each object as a set of adjacent dipoles; the resulting field is calculated by composing it with the terrestrial magnetic field. The average error induced by a metallic object on compass readings is 0.194, 0.468, and 0.661 rad when the robot is placed, respectively, within 10, 5, and 3 m from it. Fig. 14 compares the effects of EC in the presence and in the absence of metallic objects for the map in Fig. 15 (187 landmarks, 327 routes). Though first-sight correction increases both stretch and orientation errors while restoring metric consistency, refinement correction is still capable of substantially reducing their magnitude. As in the case of topological errors, valid data successfully constrain the deformations due to wrong measures.

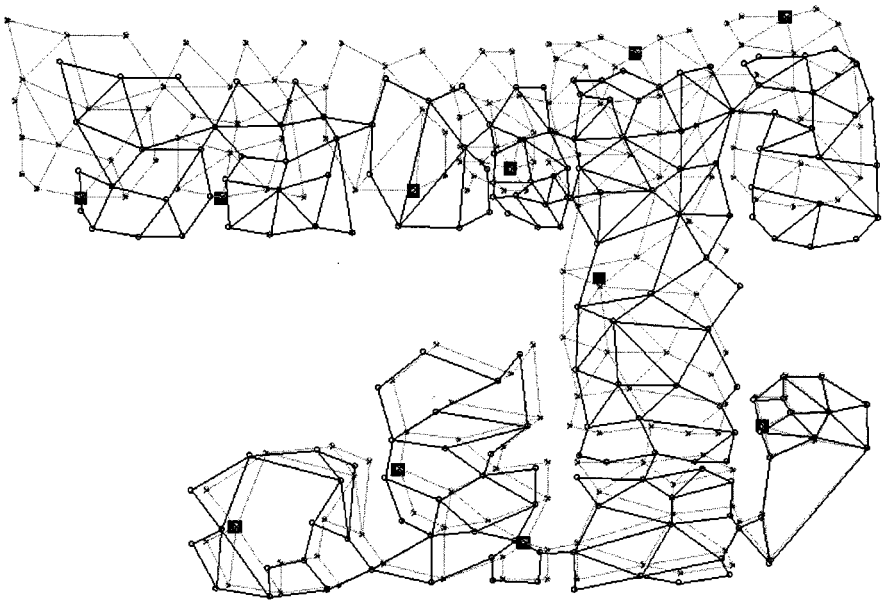


Fig. 15. Correction after five tours. The true map is gray, the corrected one is black; black squares represent metallic objects.

V. Conclusion

In this paper, we have presented a technique for correcting sensory errors during a map building process carried out by a mobile robot. The results of experimental tests confirm the effectiveness of the EC technique in reducing the global metric error and prove its robustness with reference to topological errors and to external magnetic fields. EC works independently of the specific sensors used for dead-reckoning: in fact, it can be used by a robot mounting a compass as well as by a robot using only odometers. Of course, since in the first case the robot's positional uncertainty is lower, the residual error after EC will be significantly smaller.

EC is suitable for real-world applications since its computational complexity is essentially due to the inversion of the stiffness matrix, i.e., $O(\eta^3)$ where η is the number of free nodes in the structure. For instance, for the map in Fig. 10, each correction takes approximately 20 ms on a Pentium 333-MHz architecture.

Appendix

In this section, we derive a formula for calculating the positional uncertainty of a robot mounting a compass. Let the sensory data be expressed by $\mathbf{m} = [\,w \quad \varphi\,]^T$, where w is the linear velocity and φ is the robot orientation. We denote with $\hat{\mathbf{p}}^{(k)}$ and $\mathbf{p}^{(k)}$, respectively, the true and the calculated positions at step k; we denote with $\hat{\mathbf{m}}^{(k+1)}$ and $\mathbf{m}^{(k+1)}$, respectively, the true and the measured sensory data at step $k+1$. Let

$$\hat{\mathbf{m}}^{(k+1)} = \mathbf{m}^{(k+1)} + \Delta\mathbf{m}^{(k+1)}$$

where $\Delta\mathbf{m}^{(k+1)}$ expresses the noise on sensors. In a conventional dead-reckoning method, $\Delta\mathbf{m}^{(k+1)}$ may be assumed to be distributed normally with zero mean and covariance matrix $\mathbf{M}$.

Within the dead-reckoning model for a robot mounting a compass, the position at step $k+1$ is expressed in terms of the position and the sensory measures at step k as follows:

$$\begin{aligned}\mathbf{p}^{(k+1)} &= \begin{bmatrix} x^{(k)} + Tw^{(k+1)}\cos\varphi^{(k+1)} \\ y^{(k)} + Tw^{(k+1)}\sin\varphi^{(k+1)} \end{bmatrix} \\ &\doteq g\left(\mathbf{p}^{(k)}, \mathbf{m}^{(k+1)}\right). \end{aligned} \tag{22}$$

The true position at step k may be expressed by adding a stochastic variable $\Delta\mathbf{p}^{(k)}$ to the calculated position

$$\hat{\mathbf{p}}^{(k)} = \mathbf{p}^{(k)} + \Delta\mathbf{p}^{(k)}$$

where $\mathbf{p}^{(k)} = g(\mathbf{p}^{(k-1)}, \mathbf{m}^{(k)})$. By applying (22) to $\hat{\mathbf{p}}^{(k)}$ and $\hat{\mathbf{m}}^{(k+1)}$ and by considering a modeling error $\Delta\mathbf{e}$ distributed normally with null mean value and covariance matrix $\mathbf{E}$, we find

$$\hat{\mathbf{p}}^{(k+1)} = g\left(\mathbf{p}^{(k)} + \Delta\mathbf{p}^{(k)}, \mathbf{m}^{(k+1)} + \Delta\mathbf{m}^{(k+1)}\right) + \Delta\mathbf{e}^{(k)}$$

which can be linearized in point $(\mathbf{p}^{(k)}, \mathbf{m}^{(k+1)})$

$$\begin{aligned}\hat{\mathbf{p}}^{(k+1)} \approx\, & g\left(\mathbf{p}^{(k)}, \mathbf{m}^{(k+1)}\right) \\ & + \mathbf{J}_{\mathbf{p}}{}^{(k)}\Delta\mathbf{p}^{(k)} + \mathbf{J}_{\mathbf{m}}{}^{(k)}\Delta\mathbf{m}^{(k+1)} + \Delta\mathbf{e}^{(k)} \end{aligned} \tag{23}$$

where $\mathbf{J}_{\mathbf{p}}{}^{(k)}$ and $\mathbf{J}_{\mathbf{m}}{}^{(k)}$ are the Jacobian matrices of g in $\mathbf{p}$ and $\mathbf{m}$, respectively:

$$\begin{aligned}\mathbf{J}_{\mathbf{p}}{}^{(k)} &= \left.\frac{\partial g(\mathbf{p},\mathbf{m})}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}^{(k)},\,\mathbf{m}=\mathbf{m}^{(k+1)}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \mathbf{J}_{\mathbf{m}}{}^{(k)} &= \left.\frac{\partial g(\mathbf{p},\mathbf{m})}{\partial \mathbf{m}}\right|_{\mathbf{p}=\mathbf{p}^{(k)},\,\mathbf{m}=\mathbf{m}^{(k+1)}} \\ &= \begin{bmatrix} T\cos\varphi^{(k+1)} & -Tw^{(k+1)}\sin\varphi^{(k+1)} \\ T\sin\varphi^{(k+1)} & Tw^{(k+1)}\cos\varphi^{(k+1)} \end{bmatrix}. \end{aligned}$$

From the results above, we obtain for $\Delta\mathbf{p}^{(k+1)}$ the following recursive expression:

$$\begin{aligned}\Delta\mathbf{p}^{(k+1)} &= \hat{\mathbf{p}}^{(k+1)} - \mathbf{p}^{(k+1)} = \hat{\mathbf{p}}^{(k+1)} - g\left(\mathbf{p}^{(k)}, \mathbf{m}^{(k+1)}\right) \\ &\approx \mathbf{J_p}^{(k)}\Delta\mathbf{p}^{(k)} + \mathbf{J_m}^{(k)}\Delta\mathbf{m}^{(k+1)} + \Delta\mathbf{e}^{(k)}.\end{aligned}$$

The normally distributed stochastic variable $\Delta\mathbf{p}^{(k+1)}$ expresses the positional error accumulated so far; the associated covariance matrix $\mathbf{C}^{(k+1)}$ can be calculated recursively as

$$\mathbf{C}^{(k+1)} = \mathbf{J_p}^{(k)}\mathbf{C}^{(k)}\left(\mathbf{J_p}^{(k)}\right)^T + \mathbf{J_m}^{(k)}\mathbf{M}\left(\mathbf{J_m}^{(k)}\right)^T + \mathbf{E}. \tag{24}$$

REFERENCES

[1] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. AAAI'94*, Seattle, WA, 1994.

[2] K. Basye, T. Dean, and J. S. Vitter, "Coping with uncertainty in map learning," in *Proc. 11th Int. Joint Conf. Artificial Intelligence*, 1989, pp. 347–352.

[3] C. J. Taylor and D. J. Kriegman, "Exploration strategies for mobile robots," in *Proc. IEEE Conf. Robotics and Automation*, Atlanta, GA, 1993.

[4] R. Talluri and J. K. Aggarwal, "Mobile robot self-location using model-image feature correspondence," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 63–77, Feb. 1996.

[5] D. Krantz, "Non-uniform dead-reckoning position estimate updates," in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MI, 1996.

[6] S. Cooper and H. Durrant-Whyte, "A Kalman filter for GPS navigation of land vehicles," in *Proc. Int. Conf. Intelligent Robotics and Systems*, Munich, Germany, 1994, pp. 157–163.

[7] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Trans. Robot. Automat.*, vol. 13, Apr. 1997.

[8] H. S. Dulimatra and A. K. Jain, "Mobile robot localization in indoor environment," *Pattern Recognit.*, vol. 30, no. 1, pp. 99–111, 1997.

[9] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Automat.*, vol. 12, Dec. 1996.

[10] J. Borenstein, "The CLAPPER: A dual-drive mobile robot with internal correction of dead-reckoning errors," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego, CA, 1994, pp. 3085–3090.

[11] M. Mataric, "A distributed model for mobile robot environment-learning and navigation," MIT Artif. Intell. Lab., Tech. Rep. 1228, 1990.

[12] S. P. Engelson and D. V. McDermott, "Error correction in mobile robot map learning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992.

[13] D. Maio, D. Maltoni, and S. Rizzi, "Dynamic clustering of maps in autonomous agents," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1080–1091, Nov. 1996.

[14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonom. Robots*, vol. 4, pp. 333–348, 1997.

[15] S. Thrun, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning Autonom. Robots*, vol. 31, no. 5, pp. 1–25, 1998.

[16] Y. Tonouchi, T. Tsubouchi, and S. Arimoto, "Fusion of dead-reckoned positions with a workspace model for a mobile robot by Bayesian inference," in *Proc. Int. Conf. Intelligent Robotics and Systems*, Munich, Germany, 1994, pp. 1347–1354.

[17] M. Adams *et al.*, "Control and localization of a post distributing mobile robot," in *Proc. Int. Conf Intell. Robot Syst.*, Munich, Germany, 1994, pp. 150–156.

[18] S. Thrun *et al.*, "Map learning and high-speed navigation in RHINO," in *Artificial Intelligence and Mobile Robots*, D. Kortenkamp *et al.*, Eds. Cambridge, MA: MIT Press, 1998.

[19] H. C. Martin, *Introduction to Matrix Methods of Structural Analysis*. New York: McGraw-Hill, 1966.

[20] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1987.

[21] P. Hébert, S. Betgé-Brezetz, and R. Chatila, "Probabilistic map learning: Necessity and difficulties," in *Proc. Int. Workshop on Reasoning with Uncertainty in Robotics*, Amsterdam, The Netherlands, 1995.

[22] H. Sugiyama, "A method for an autonomous mobile robot to recognize its position in the global coordinate system when building a map," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robotics and Systems*, Yokohama, Japan, 1993, pp. 2186–2191.

[23] B. Yamauchi and R. Beer, "Spatial learning for navigation in dynamic environments," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 496–505, June 1996.

[24] S. Koenig and R. Simmons, "Unsupervised learning of probabilistic models for robot navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996.

Matteo Golfarelli received the Ph.D. degree for research on autonomous agents from the University of Bologna, Bologna, Italy, in 1998.

Since July 2000, he is Associate Researcher at the Computer Science Department, University of Bologna. His research interests also include biometric systems and databases.

Dario Maio (M'90) received the Laurea degree in electronic engineering from the University of Bologna, Bologna, Italy, in 1975.

He is Full Professor at the University of Bologna. He has published in the fields of distributed computer systems, computer performance evaluation, database design, information systems, neural networks, autonomous agents, and biometric systems.

Stefano Rizzi received the Ph.D. degree for his work on autonomous agents from the University of Bologna, Bologna, Italy, in 1996.

Since 1998, he has been an Associate Professor at the Computer Science Department, University of Bologna. He has published in the fields of mobile robotics, pattern recognition, data warehousing, and visual query languages.