Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 48 (2004) 145–162
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Optimization of the POSIT algorithm for indoor autonomous navigation

Tommaso Gramegna*, Lea Venturino, Grazia Cicirelli, Giovanni Attolico, Arcangelo Distante

Istituto di Studi sui Sistemi Intelligenti per l'Automazione-CNR, Via Amendola, 122/D-I-70126 Bari, Italy

Received 6 August 2003; received in revised form 6 April 2004; accepted 4 May 2004

Available online 8 July 2004

Abstract

This paper presents an optimization of POSIT (Pose from Orthographic and Scaling with Iterations), a model-based camera location algorithm, in the domain of indoor mobile robot navigation. The method finds the rotation matrix and the translation vector of the camera with respect to an object. The novelty of the proposed modified algorithm is that it does not need the perfect knowledge of camera parameters. A new definition of the scaling factor has been introduced in the scaled orthographic projection. Due to the peculiarity of the indoor bounded workspace a new formulation of the translation vector has been used. The new method has been successfully applied in a real environment considering a goal-directed navigation task for our real robot Khepera. The experimentation has shown better results in comparison with the original POSIT algorithm.

Keywords: Self-localization; Autonomous robot indoor navigation; Visual servoing control system

1. Introduction

The problem of camera localization relative to real world objects by using a single view arises in several domains of application such as autonomous robot navigation, object tracking, telepresence, virtual and augmented reality and so on [7,12,13,16,26,27,31].

Relating to the robot navigation area, several methods to estimate the robot position and orientation (which is known as 'pose') can be found in literature [9,23,28]. Most of them use visual patterns or landmarks for self-localization, usually represented by 3D models which must be matched to their corresponding 2D projections in the image plane [4]. Sometimes artificial landmarks are preferred to make easier the problem associated with the recognition of naturally occurring patterns [6,26], but this requires to structure the environment that is not always possible.

In visual servoing domain [14] the localization methods are classified on the basis of the required knowledge about the target and the camera parameters. Depending on the knowledge of the 3D target model, model-based [8] or model-free [3] visual servoing approaches can be used. The common problem to both the approaches is the matching problem, currently studied by most researchers in the field. In

* Corresponding author. Tel.: +39-080 592 9447; fax: +39-080 592 9460.
E-mail addresses: gramegna@ba.issia.cnr.it (T. Gramegna), venturino@ba.issia.cnr.it (L. Venturino), grace@ba.issia.cnr.it (G. Cicirelli), attolico@ba.issia.cnr.it (G. Attolico), distante@ba.issia.cnr.it (A. Distante).

0921-8890/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2004.05.001

particular it consists in establishing correspondences between the target model and the current image [15] for model-based approaches and between the reference image and the current one [32] for model-free approaches.

If feature correspondence is known, several techniques for solving the problem of model-based pose recovery have been presented in literature [19]. According to the nature of the camera models and of the mathematical techniques employed, they can be grouped in: analytical, numerical and affine techniques.

The first solution to the problem of model-based pose recovery with a full-perspective camera model belongs to the analytical group and is due to Fischler and Bolles [11]. They present a new solution to the problem of camera location based on the RANSAC paradigm, a methodology proposed for processing large data sets with gross errors or outliers. They obtain solutions considering three and four coplanar feature points as well as six points in general position demonstrating their uniqueness. Unfortunately there is a lack of generality, i.e. no reference is given about the existence and uniqueness of the solution in the general case. The solution based on four coplanar point has been successively implemented for the estimation of the position of a robot [17].

Different analytical methods for the estimation of camera pose from line segments instead of point features have also been developed [2,5,10,18]. One of the weaknesses of the analytical techniques is the presence of multiple solutions. Although different alternatives have been proposed [25] to deal with them, they do not guarantee the uniqueness of the solution. Furthermore, these methods are computationally expensive and, since they are based only on a small number of correspondences, they can produce completely wrong results if some incorrect feature matches occur.

In the case of numerical methods, an error function expresses the distance between each image feature and the projection of the corresponding feature in the real world by using the current camera location. This transformation is corrected by an iterative process starting from the initial estimate of the camera location. Within these methods, Lowe's algorithm [20–22] was the first attempt to use this technique. Ait-Aider et al. developed an adaptation of this method to mobile robot self-localization [1]. These solutions are in general more accurate than the analytical ones, but they are more complex and computationally demanding, then their application in real-time contexts is difficult, if not impossible.

Finally the affine solutions are based on the use of linearized camera models (weak-perspective and paraperspective camera models), avoiding the intrinsic non-linearity of the geometrical constraints that arises when the imaging process is modeled as a perspective transformation. These techniques are simple, efficient and, contrary to the analytical solutions, they work for scenes with an arbitrary number of features. The POSIT algorithm [8] falls within these techniques. It is based on geometric considerations that combine perspective projection and scaled orthographic projection (SOP) applied to feature points. The idea POSIT bases on is firstly to calculate the pose with a weak-perspective camera model (SOP), and then to refine the obtained pose in a loop to go towards a full-perspective pose estimation. The main advantage of this method consists of its very low computational cost, then it is well suited for real-time applications. However, linearized camera models are approximations whose validation is sometimes questionable in practice in terms of accuracy and robustness of pose estimation.

In this paper we present a new formulation of POSIT algorithm. It has been deeply studied and analyzed considering our application field of interest: the mobile robot navigation domain. In this context our concern is the estimation of the pose of mobile robot with respect to a target object in its workspace environment. The original formulation of POSIT algorithm has been tested on synthetic data first. Then both algorithms, POSIT and the proposed new formulation, have been compared by using real images acquired by the on-board camera of the Khepera mini robot. Our aim is to estimate the pose of the robot by means of an algorithm offering improved performance in terms of efficiency, robustness and precision. Even though the significant lens distortion caused by the poor quality of the camera mounted on the Khepera robot and the bounded workspace the real experimentation has proved the reliability and the accuracy of the new formulation of the algorithm.

In particular the main contribution of our work concerns the new definition of the SOP scaling factor. It has been defined in a way that allows one to apply

the method to uncalibrated real images. This is a fundamental property since in real-time robotics applications the calibration process is often tedious or unfeasible. Finally due to the peculiarity of the bounded experimental workspace we have developed a new formulation of the translation vector which joins the robot position to the reference frame of the target object. In addition to improve efficiency, accuracy and robustness of the robot task we have inserted the new formulated method in a control system made up of two closed loops.

The rest of the paper is organized as follows. In Section 2, the original formulation of POSIT algorithm is presented and the results obtained in simulation with synthetic data are shown. Section 3 discusses the modifications and the improvements to the algorithm in order to adapt it to the indoor navigation context. In Section 4 experimental results on real images and the evaluation of the performance of the new method in comparison with the original one are illustrated. Section 5 presents the developed control scheme of the robot for reaching a goal position and the relative experimental results. Finally some conclusions are drown at the end of the paper.

2. POSIT algorithm

In this section we firstly introduce the formulation of the original version of the POSIT algorithm proposed by DeMenthon and, successively, we show the validation of the method using synthetic data.

2.1. Original POSIT algorithm formulation

The POSIT algorithm finds the pose of an object from a single image, on the base of the 3D target model in the object coordinate frame of reference. Necessary conditions are the extraction of at least four non-coplanar points and the matching of the extracted feature with the corresponding model points. It combines two algorithms. The first, called POS (Pose from Orthography and Scaling), approximates the perspective projection with a scaled orthographic projection and finds the rotation matrix and the translation vector of the robot coordinate frame with respect to the target coordinate frame of reference by solving a linear system. The second algorithm, POSIT (POS with

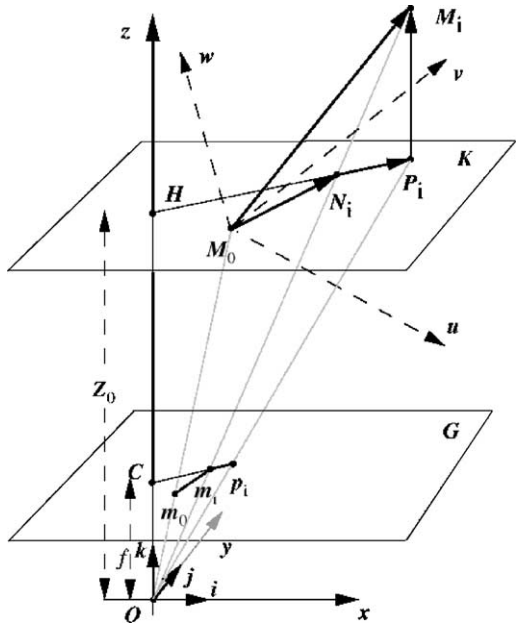


Fig. 1. SOP (p_i) and perspective projection (m_i) of an object point M_i and a reference point M_0.

Iterations), is an iterative method that applies POS to the approximate pose found in the previous step in order to compute better scaled orthographic projections of the feature points.

In a pin-hole camera model, as shown in Fig. 1, a target object, with feature points $M_0, \ldots, M_i, \ldots, M_n$, is positioned in the camera field of view. The focal length f, the M_i coordinates in the object coordinate frame of reference point M_0 and the image points m_i having image coordinate (x_i, y_i) are all known. The coordinates (X_i, Y_i, Z_i) of the points M_i in the camera coordinate system are unknown. The goal is to compute the rotation matrix and the translation vector of the object directly, without solving explicitly for the coordinates (X_i, Y_i, Z_i).

The rotation matrix is the matrix whose rows are the coordinates of the unit vectors **i**, **j** and **k** of the camera coordinate in the object coordinate frame. The translation vector **T** is the vector $\mathbf{OM}_0$ between the centre of projection, O, and the reference point M_0, the origin of the object coordinate frame of reference. Therefore the coordinates of the translation vector are X_0, Y_0, Z_0. If this point M_0 has been chosen to be a visible feature point for which the image is a point m_0, this translation vector **T** is aligned with vector

$\mathbf{Om}_0$ and is equal to $Z_0/f\ \mathbf{Om}_0$. Since the vector $\mathbf{k}$ is obtained by the cross-product $\mathbf{i} \times \mathbf{j}$, the object pose is fully defined once $\mathbf{i}$; $\mathbf{j}$ and Z_0 are computed.

After the problem has been defined, the fundamental equations of Scaled Ortografic Projection (SOP) and Perspective Projection can be introduced. SOP is an approximation of the true perspective projection. In this approximation, all M_i points have the same depth Z_0 as the reference point M_0. In SOP, then, the image point $\mathrm{p_i}$ of object point M_i has the following coordinates:

$$x'_i = f\frac{X_i}{Z_0}, \qquad y'_i = f\frac{Y_i}{Z_0}. \tag{1}$$

In the perspective projection the image point m_i of the same point M_i has coordinates

$$x_i = f\frac{X_i}{Z_i}, \qquad y_i = f\frac{Y_i}{Z_i}. \tag{2}$$

The ratio $s = f/Z_0$ is the scaling factor of the SOP.

From Fig. 1 it can be observed that vector $\mathbf{M_0M_i}$ is the sum of three vectors:

$$\mathbf{M_0M}_i = \mathbf{M_0N}_i + \mathbf{N}_i\mathbf{P}_i + \mathbf{P}_i\mathbf{M}_i. \tag{3}$$

This equation can be early expressed using the triangle similitude as

$$\mathbf{M_0M}_i = \frac{\mathrm{Z_0}}{f}\mathbf{m_0m}_i + \frac{\mathbf{M_0M}_i \cdot \mathbf{K}}{f}\mathbf{Cm}_i + \mathbf{P}_i\mathbf{M}_i. \tag{4}$$

The dot product of Eq. (4) with the unit vectors $\mathbf{i}$ and $\mathbf{j}$ gives the following equations:

$$\mathbf{M_0M}_i \cdot \frac{f}{Z_0}\mathbf{i} = x_i(1+\varepsilon_i) - x_0; \tag{5}$$

$$\mathbf{M_0M}_i \cdot \frac{f}{Z_0}\mathbf{j} = \mathrm{y}_i(1+\varepsilon_i) - y_0; \tag{6}$$

where ε_i is defined as

$$\varepsilon_i = \frac{1}{Z_0}\mathbf{M_0M}_i \cdot \mathbf{k}. \tag{7}$$

Using SOP the vector $\mathbf{M_0M_i}$ is the sum of two vectors $\mathbf{M_0P_i}$ and $\mathbf{P_iM_i}$. In this case, Eq. (4) can be written as

$$\mathbf{M_0M}_i = \frac{Z_0}{f}\mathbf{m_0m}_i + \mathbf{P}_i\mathbf{M}_i. \tag{8}$$

The dot product of Eq. (8) with the unit vectors $\mathbf{i}$ and $\mathbf{j}$ gives the following equations:

$$\mathbf{M_0M}_i \cdot \frac{f}{Z_0}\mathbf{i} = x'_i - x_0; \tag{9}$$

$$\mathbf{M_0M}_i \cdot \frac{f}{Z_0}\mathbf{j} = y'_i - y_0. \tag{10}$$

Comparing these equations with Eqs. (5) and (6), it can be observed that it is possible to write p_i coordinates as

$$x'_i = x_i(1+\varepsilon_i); \tag{11}$$

$$y'_i = y_i(1+\varepsilon_i). \tag{12}$$

Eqs. (5) and (6) can also be written as

$$\mathbf{M_0M}_i \cdot \mathbf{I} = x_i(1+\varepsilon_i) - x_0 \tag{13}$$

$$\mathbf{M_0M}_i \cdot \mathbf{J} = y_i(1+\varepsilon_i) - y_0; \tag{14}$$

where $\mathbf{I} = (f/Z_0)\mathbf{i}$ and $\mathbf{J} = (f/Z_0)\mathbf{j}$.

If ε_i is known, Eqs. (13) and (14) provide a linear system of equations in which the only unknowns are $\mathbf{I}$ and $\mathbf{J}$. Once $\mathbf{I}$ and $\mathbf{J}$ have been computed, the scaling factors $s_1 = (\mathbf{I}\cdot\mathbf{I})^{1/2}$, $s_2 = (\mathbf{J}\cdot\mathbf{J})^{1/2}$ and $s = (s_1+s_2)/2$ are obtained. The unit vectors $\mathbf{i}$ and $\mathbf{j}$ derive from the normalization of $\mathbf{I}$ and $\mathbf{J}$ by means of s_1 and s_2 in the following way: $\mathbf{i} = \mathbf{I}/s_1$; $\mathbf{j} = \mathbf{J}/s_2$.

The POS algorithm finds the pose for which the point $\mathbf{M}_i$ has, as scaled orthographic projections, the image point $\mathbf{p}_i$. In this case ε_i is equal to zero.

This solution is an approximation because ε_i is not exact. Once $\mathbf{i}$ and $\mathbf{j}$ have been computed, a more exact ε_i, can be computed in the POSIT algorithm, and the equations can be solved again with these better values. The new ε_i is obtained from Eq. (7) computing the unit vector $\mathbf{k}$ and the z coordinate Z_0 of the translation vector as $Z_0 = f/s$.

By iterating these steps, the method converges to an accurate SOP image and an accurate pose. The method is stopped when the scaled orthographic image points are the same as those found at the previous step.

In this case, output pose using values found at the last iteration are the full translation vector $\mathbf{OM}_0$ and the rotation matrix with row vectors $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ defined in the following way:

$$\mathbf{OM}_0 = \frac{\mathbf{Om}_0}{s}; \tag{15}$$

$$R = \begin{pmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{pmatrix}. \tag{16}$$

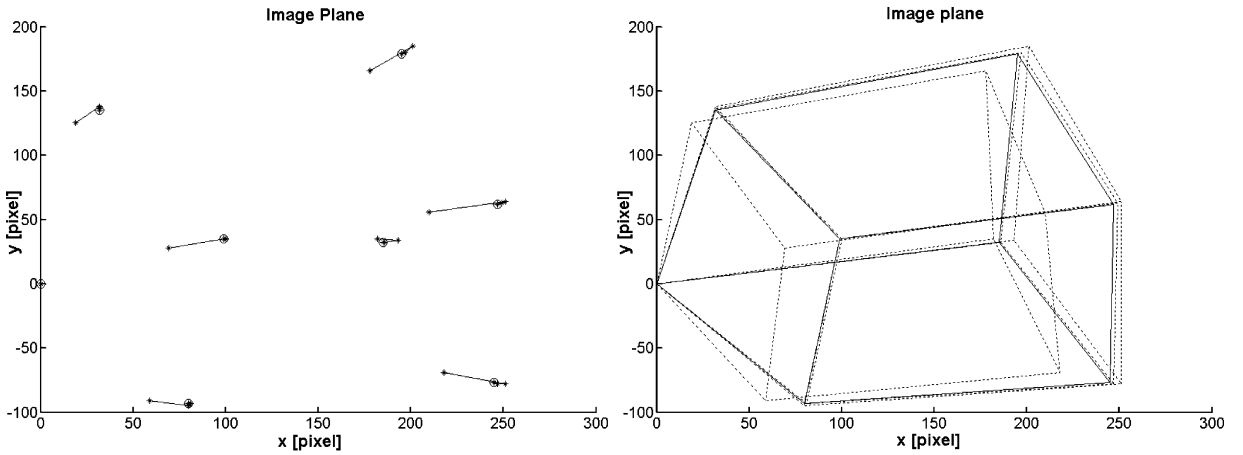


Fig. 2. Iterations of the POSIT algorithm on the image plane. (a) Iterations of image points on the image plane: (*) image points during each iteration of the POSIT algorithm; (○) convergence of the image points. (b) Iterations of the cube on the image plane: (- - -) cube on the image plane during each iteration of the POSIT algorithm; (—) cube on the image plane at the convergence point.

2.2. Evaluation of the original method using synthetic images

In order to determine the performance of the developed POSIT algorithm, experimental tests using synthetically generated images have been executed. This procedure provides exact reference values to which the evaluated measures can be compared.

The algorithm has been evaluated using as input data a list of 2D image points and a list of corresponding 3D object points reported by DeMenthon in the presentation of the original method [8]. The target object is a cube and the reference point (one of the cube corners) is positioned on the optical axis so the translation vector is made up only of the component along the optical axis. The algorithm successively converges, as it is shown in Fig. 2. In Fig. 2(a) image points at each iteration are plotted. Fig. 2(b) shows the perspective projection on the image plane of the cube used to perform the experiment during each iteration of the method after the application of the rotation matrix and the translation vector.

Moreover, we have analyzed the pose parameter errors with respect to the number of the considered feature points, as it is shown in Fig. 3.

It is possible to notice that the pose parameters errors decrease when the number of feature points increases. Nonetheless, even in the worst case (four non-coplanar points) the difference with respect to the other cases is negligible.

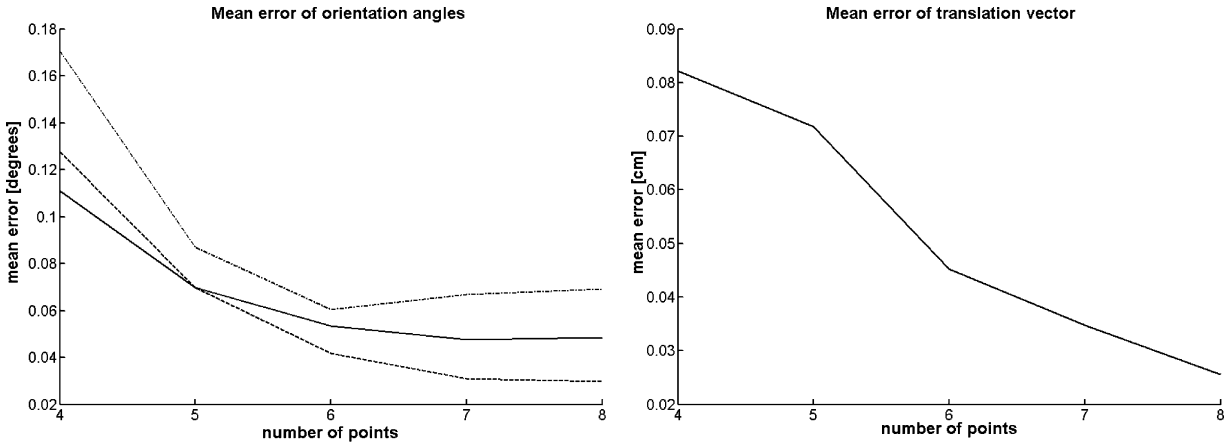
Successively, we have changed the 3D object using the parallelepiped shown in Fig. 4 and we have chosen an arbitrary pose, composed by

- Translation vector (cm): $[t_x, t_y, t_z] = [4, 5, 20]$;
- Euler angles (°): $[\theta, \varphi, \phi] = [30, 40, 50]$;

where the zeta component of the translation vector is along the optical axis.

We have evaluated the pose parameters errors with respect to the ratio of the distance to the camera to the object size. Five ratios have been considered: the first five powers of two. The orientation and the translation errors obtained are shown in Fig. 5. We can observe that errors, especially on the zeta component of the translation vector, increase in proportion to the ratio. In fact, the pixel quantization of the camera makes progressively difficult to appreciate the displacement of a point in the scene.

Another evaluation has been done on the influence of errors on the image points: we have tested the algorithm robustness and accuracy when noisy image points are considered to belong to the model. We have considered a range of two noise pixels around each image point. The errors modify the relative geometry and the resulting pose computation. Since the rotation and the translation vector are obtained using di-



(a) (b)

Fig. 3. Different pose parameters errors depending on the number of utilized feature points considering all possible combinations. (a) Mean error of the orientation angles: (·····) mean error of ϕ angle; (- - -) mean error of θ angle; (—) mean error of φ angle. (b) Mean error of the translation vector: (—) mean error of the translation vector component along the optical axis.

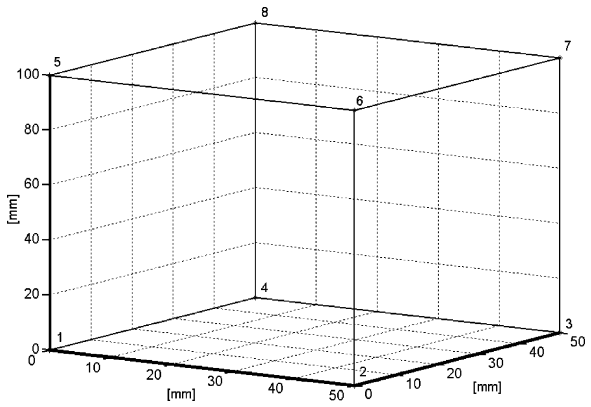


Fig. 4. Parallelepiped used in the simulation as object target.

rectly the information on the reference image point, Eqs. (15) and (16), the biggest pose estimation error is on the bounds of the noise regions placed in this point.

In Table 1 the maximum error values relative to each component, in correspondence with the noise source that has generated them, are shown in bold. As it can be seen, all maximum values are generated in the noise region placed on the reference point (point 1), except for the zeta component of the translation vector which depends only on the focal length and the scaling factor, as explained in Section 2.1.

Finally we have analyzed the accuracy of the algorithm considering the variation of all the pose parameters. The reference locations are represented by a vector of the six position and orientation parameters $[t_x, t_y, t_z, \theta, \varphi, \phi]$, uniformly distributed over the following intervals of values:

4 cm, 16 cm	for t_x;
5 cm, 20 cm	for t_y;
20 cm, 80 cm	for t_z;
20°, 40°	for θ;

Table 1
Pose estimation error depending on the presence of wrong image points

Noise source (pixel)	Error t_x (cm)	Error t_y (cm)	Error t_z (cm)	Error θ (°)	Error φ (°)	Error ϕ (°)
NO	0.000313	0.000392	0.001566	0.047043	0.490235	0.540647
Point 1 (2, 2)	0.050441	0.049901	0.010810	0.013806	1.195934	0.816510
Point 1 (−2, −2)	0.050103	0.049463	0.012809	0.093172	2.184758	1.910571
Point 1 (2, −2)	0.046296	0.060366	0.031265	0.652733	0.472263	0.570506
Point 1 (−2, 2)	0.046077	0.061021	0.033208	0.569025	0.491358	0.529962
Point 8 (2, 2)	0.018930	0.023663	0.094652	0.046421	1.135691	0.924275

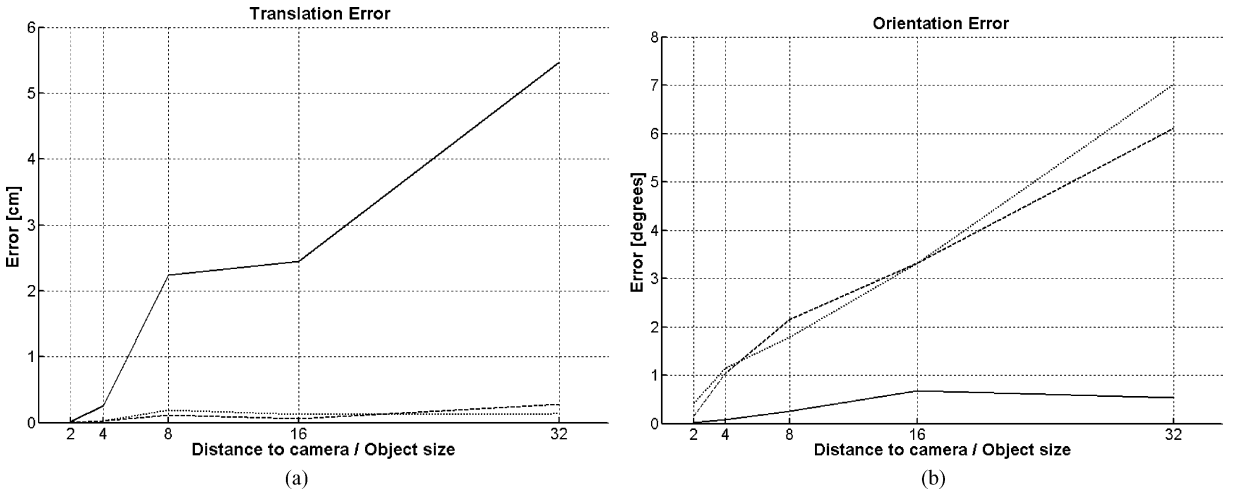


Fig. 5. Pose estimation errors depending on the ratio distance to camera-object size. (a) Translation error: (- - -) t_x component of the translation vector; (·····) t_y component of the translation vector; (—) t_z component of the translation vector. (b) Orientation error: (—) θ angle; (·····) φ angle; (- - -) ϕ angle.

30°, 50° for φ;
40°, 60° for ϕ.

Seven hundred and twenty nine different situations has been generated and the mean error e_i relative to each component i is: $[\, e_{t_x} \quad e_{t_y} \quad e_{t_z} \quad e_\theta \quad e_\varphi \quad e_\phi \,] = [\, 0.0343\,\text{cm} \quad 0.0553\,\text{cm} \quad 0.2036\,\text{cm} \quad 0.1231^\circ \quad 0.5383^\circ \quad 0.4980^\circ]$. The good results obtained in this section allowed us to test POSIT method in a practical application. In the next section we describe how we have optimized the method in order to use it in the experimental context of autonomous navigation of a mobile mini robot.

3. Optimization of POSIT algorithm in a mobile robotic indoor navigation context

Considering the specific requirements of the mobile robot navigation domain and the significant lens distortion caused by the uncalibrated camera mounted on the Khepera robot, a new formulation of POSIT algorithm has been developed.

The low performance of the original POSIT algorithm using uncalibrated real images is due to the camera perspective approximation on which this algorithm is based. To overcome this problem, we have introduced a new definition of the SOP scaling factor. On the other hand, the bounded workspace limits the robot movements control in the autonomous navigation. We have modified the formulation of the translation vector of the original algorithm, obtaining the rotation–translation sequence instead of the translation–rotation one, in order to position the robot in the environment from any point in any goal position.

3.1. Pose recovery with uncalibrated real images

Eq. (15) is valid for a camera perspective approximation, on the basis of the theoretical hypotheses of the POSIT algorithm, but, in the case of uncalibrated real images, this assumption is not valid; in fact, the poor quality of the camera mounted on the Khepera robot causes a large distortion in the image plane.

The use of a single scaling factor in the definition of the translation vector is valid only if the image plane has no distortion. Normally, the problem of lens distortion can be solved by camera calibration [24,29,30] but this causes image preprocessing and high computational cost. The idea on the base of the new version of the method is to use different scaling factors in the definition of the translation vector.

The three scaling factor s_1, s_2, and s, defined in Section 2.1, all converge to a single value on synthetic data obtained by simulating an image plane,

since there is not lens distortion. This does not happen in the real case.

Since the scaling factors s_1 and s_2 are defined from the normalization of the unit vectors of the image plane, it is possible to use these scaling factors obtaining a 'scaling factor vector' $\mathbf{s}_V$. In this case the new translation vector is

$$\mathbf{T} = \begin{bmatrix} \frac{x_0}{s_x} \\ \frac{y_0}{s_y} \\ \frac{f}{s_z} \end{bmatrix}, \tag{17}$$

where $\mathbf{s}_V = [\, s_x \quad s_y \quad s_z \,] = [\, s_1 \quad s_2 \quad s \,]$.

Using this method we have obtained better results with respect to the original method, as will be shown in Section 4.

3.2. Pose control for autonomous robot navigation in a bounded workspace

Since the workspace dimensions are limited, we have modified the original version of the algorithm in order to obtain the rotation–translation sequence instead of the translation–rotation one. This allows a better control of the robot position.

Given two frames 1 and 2, the rotation matrix that represents the orientation of the frame 2 with respect to frame 1 is denoted by $\mathbf{R}_{12}$. The location of the origin of the frame 2 with respect to the frame 1 is denoted by the vector $\mathbf{T}_{12}$. If we are given $\boldsymbol{P}_2$, (the coordinates of the point relative to the frame 2) we obtain the coordinates of the point with respect to the frame 1 ($\mathbf{P}_1$) by applying the coordinate transformation rule:

$$\mathbf{P}_1 = \mathbf{R}_{12}\mathbf{P}_2 + \mathbf{T}_{12}. \tag{18}$$

Eq. (18) is used for the translation–rotation sequence in the original POSIT algorithm. Unfortunately, the application of this sequence can cause the robot to move out of the environment bounds. To get over this problem, we have used a new equation to obtain a rotation–translation sequence:

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{R}_{12}(\mathbf{P}_2 + \mathbf{T}_{12}) = \mathbf{R}_{12}\mathbf{P}_2 + \mathbf{R}_{12}\mathbf{T}_{12} \\ &= \mathbf{R}_{12}\mathbf{P}_2 + \mathbf{T}'_{12}. \end{aligned} \tag{19}$$

In order to obtain the real translation vector $\mathbf{T}_{12}$ we have applied the following equation:

$$\mathbf{T}_{12} = \mathbf{R}_{12}^{-1}\mathbf{T}'_{12}. \tag{20}$$

Using the new formulation of the translation vector the robot is able to reach a goal position in a bounded indoor workspace as will be shown in Section 5.

4. Evaluation of the method using real images

The goal of the test described in this section is to confirm the applicability of the POSIT algorithm and of its new version in a real-time application using a Khepera miniaturized robot. A color CCD camera (PC-215 PAL) has been mounted on it. The target object is made up of two black regions: a 'door' region and a 'small area' region. The black door-area region is in a white closed environment as shown in Fig. 6.

The method finds the rotation matrix and the translation vector of the robot coordinate frame with respect to the object coordinate frame of reference. The reference point of the target is the door corner represented with a white cross marker (+) in Fig. 6; the y-direction is along the intersection between the door plane and the floor; the z-direction is along the normal to the door plane. This robot has only three degrees of freedom: it can move only in the z–y plane and can rotate around its vertical axis. According to this, we have defined three position parameters: ϕ, the rotation angle between the optical axis and z-axis; T_y, the distance between the current position and the reference point along the y-axis; T_z, the distance between the current position and the reference point along the z-axis.

The poor quality of the camera mounted on the robot causes a significant lens distortion in the image plane. For this reason we have modified the original version of the POSIT algorithm as described in Section 3.1. In this case only the y component of the translation vector has a new formulation in the modified version.

For the validation of the final results we have collected different data set uniformly distributed on the workspace, as shown in Fig. 7. The camera orientation is set so that the whole target is in the field of view of the camera. For any point we have carried out 500 measures.

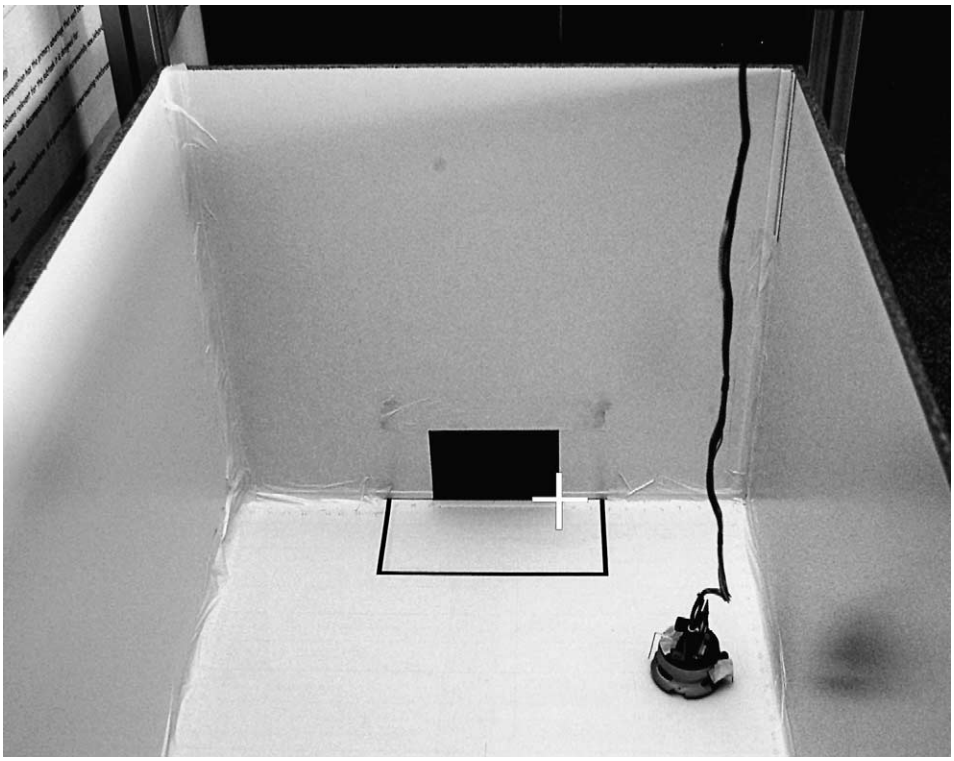


Fig. 6. Bounded workspace made up of a black door-area target in a white environment and the Khepera robot used in our real-time application.

Fig. 8 shows the comparison between the original version of the POSIT algorithm and its modified version in terms of the median error and the standard deviation of the y component of the translation vector. These values have been evaluated on the sample of 500 measures for each point. In Fig. 8, for each method, three graphics corresponding to the median value and to the extremes of the interval determined by the corresponding standard deviation are drawn: the latter two graphics mostly overlap the one of the median error, showing the robustness of the algorithms, since the standard deviation of the error for both methods is very small. It is possible to notice the improvement of the modified version with respect to the original algorithm by comparing the median values.

This improvement is much more evident by analyzing the comparison between the mean e_{my}, the integral e_{iy} and the standard deviation σ_y of the median error values of the y component evaluated on the whole set of points on the workspace calculated for the POSIT algorithm and its modified version as shown in Table 2. Notice that the error values and the standard deviation relative to the POSIT algorithm are worse than those relative to the modified POSIT.

For the sake of completeness the same error analysis has been carried out on the z component and the ϕ component, too, by using the same real image. Fig. 9 shows the obtained plots respectively.

The average e_m, the integral e_i and the standard deviation σ of these errors are shown in Table 3. It is possible to compare these results with the ones ob-

Table 2
Comparison between the mean e_{my}, the integral e_{iy} and the standard deviation σ_y of the median error values of the y component for each point for the POSIT algorithm and its modified version

	e_{iy} (cm)	e_{my} (cm)	σ_y (cm)
POSIT	55.006	2.292	1.398
Modified POSIT	26.143	1.089	1.302

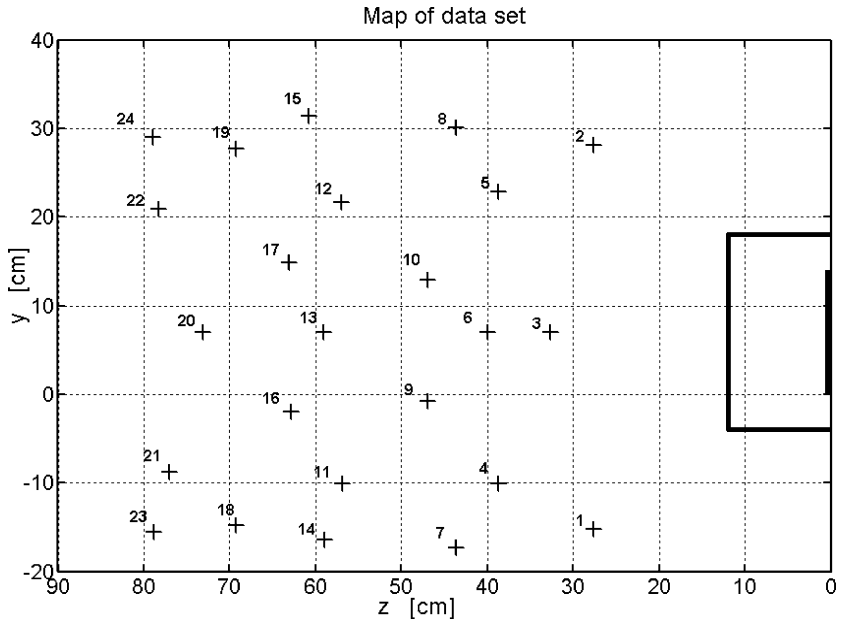


Fig. 7. Map of data set.

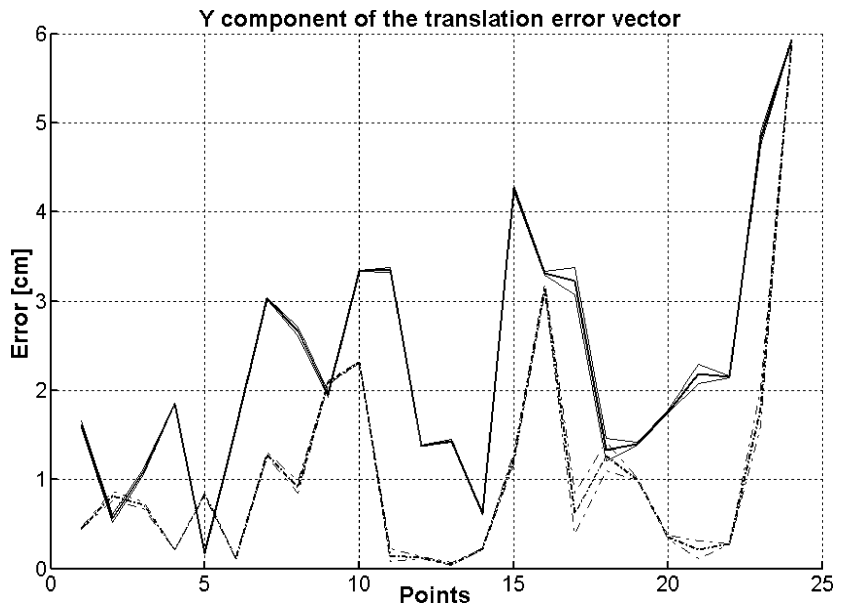


Fig. 8. Comparison between the error of the *y* component of the translation vector for the POSIT algorithm and its modified version: (—) median value of the POSIT algorithm; (—) median + standard deviation value of the POSIT algorithm; (—) median − standard deviation value of the POSIT algorithm. (-·-·-) median value of the modified POSIT algorithm; (-·-·-) median + standard deviation value of the modified POSIT algorithm; (-·-·-) median − standard deviation value of the modified POSIT algorithm.

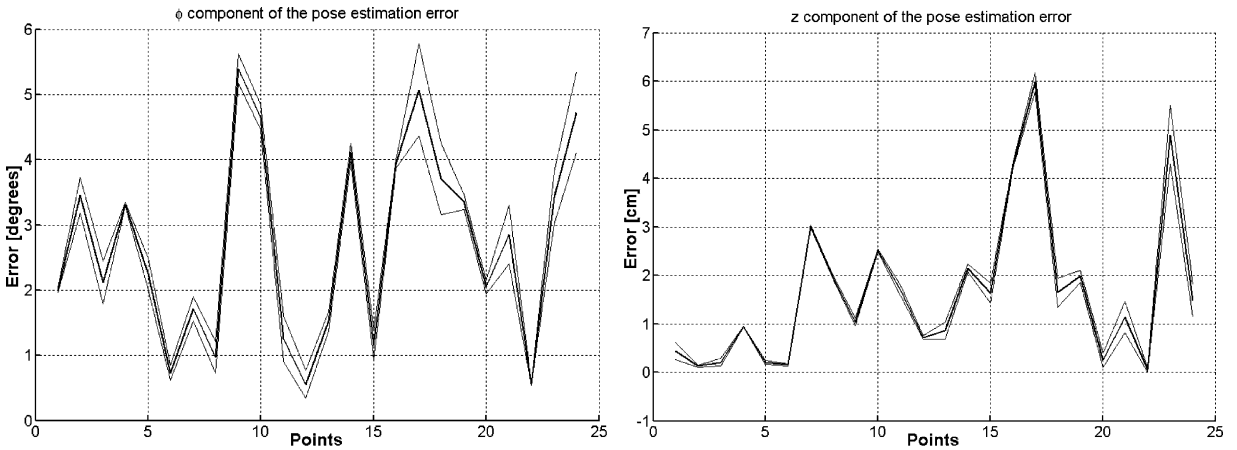


Fig. 9. Pose estimation errors. (a) z component of the translation error vector: (—) median value; (—) median + standard deviation value; (—) median − standard deviation value. (b) ϕ component of the orientation error: (—) median value; (—) median + standard deviation value; (—) median − standard deviation value.

Table 3
Mean e_{m}, integral e_{i} and the standard deviation σ of the z component and the ϕ component of the pose estimation error

	e_{i}	e_{m}	σ
ϕ component (°)	64.8692	2.7029	1.4974
z component (cm)	39.3541	1.6398	1.5685

tained in Section 2.2, even though different 3D object models have been used. The mean error relative to the components z and ϕ obtained by using real images is worse than that obtained by using synthetic images.

Actually thanks to the results obtained with uncalibrated real images, we have utilized the pose estimation method in a practical application of autonomous robot navigation in a closed environment. Since the workspace dimensions are limited we have modified the algorithm as described in Section 3.2. The application of Eq. (20) modifies the translation vector obtained by the version of POSIT algorithm discussed above. In Table 4 the average e_{m}, the integral e_{i} and the standard deviation σ_i of the error relative to the new translation vector for the POSIT algorithm and its modified version are shown. The y component of the new translation vector in the modified version of the POSIT algorithm still shows good performances. Instead the z component performances get a bit worse. On the other hand, this new formulation of the translation vector is necessary in the considered application of autonomous robot navigation due to the limitation of the workspace and the peculiarity of the mini robot Khepera.

We have finally inserted the modified version of the POSIT algorithm in a visual servoing control system. The robot task is to reach the goal pose with the desired precision after few iterations, as it is shown in the next section.

5. Control system

The last version of the POSIT algorithm has been used in a practical application of autonomous robot

Table 4
Comparison between the mean e_{m}, the integral e_{i} and the standard deviation σ_i of the new translation vector for the POSIT algorithm and its modified version

	$e_{\mathrm{i}z}$ (cm)	$e_{\mathrm{m}z}$ (cm)	σ_z (cm)	$e_{\mathrm{i}y}$ (cm)	$e_{\mathrm{m}y}$ (cm)	σ_y (cm)
POSIT	32.212	1.342	0.352	85.415	3.559	3.291
Modified POSIT	41.949	1.748	1.628	59.918	2.497	2.197

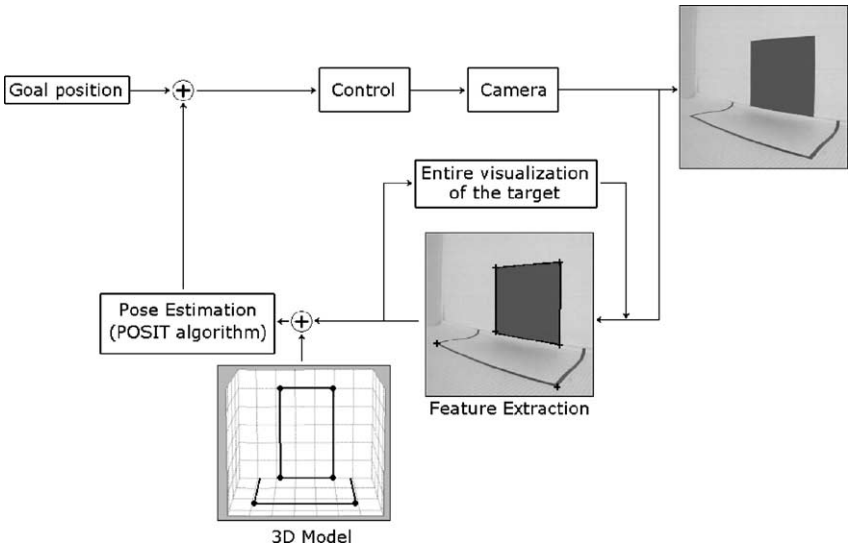


Fig. 10. Scheme of the developed visual servoing control system.

navigation for reaching a predefined position in the environment.

For this application, we have inserted the algorithm in a closed loop control system which allows to obtain a good performance in spite of the poor quality of the camera mounted on the robot. The developed control system is made up of two closed loops, as shown in Fig. 10.

The inner loop is necessary to drive the vehicle until the whole target appears in the field of view of the camera (this process produces the rotation and the translation of the camera–robot joint system). Once the desired and the current camera pose are available, the camera displacement to reach the desired position is computed and the control of the robot is performed in the principal closed loop.

In the screen output of the control system (see Fig. 11) the toolbar for the robot movement control, the current image displayed, the results of the developed feature extraction algorithm, and the output data pertaining to the extracted features and the computed robot pose parameters are shown.

The feature extraction algorithm has been developed in few steps: the first one consists in applying a threshold to the gray level input image. The image is thus transformed in a binary image where the door-area target is black and the environment is white. This binary image is further processed in order to separate the door region from the small area region, using the information about the door being the biggest region extending around the image barycentre. The door region is limited by an over-estimation of the standard deviation relative to the distribution of black pixels in the current image. Outside the limited region of the door, all the black pixels belong to the blue small area region. In the next step we have found the four sides of the door region in order to obtain four distinct point sequences. For each of these loci, the method of the regression lines has been used to compute the slope and the vertical intercept, identifying the corresponding lines. The intersection points of these four lines determine the feature points corresponding to the door corners. Since a necessary condition for the POSIT algorithm is the extraction of features corresponding to at least four non-coplanar object points, it has been necessary to use at least another feature point non-coplanar with the ones on the door. The corners of the small area has been used for this aim. This feature extraction is based on geometric considerations because the shape is quite different from the one of the door. The computation of small area corners uses the door features just extracted and the relative distances from the corners of the image. The combination of these two feature extraction techniques provides six non-coplanar feature points with an error not greater than two pixels.

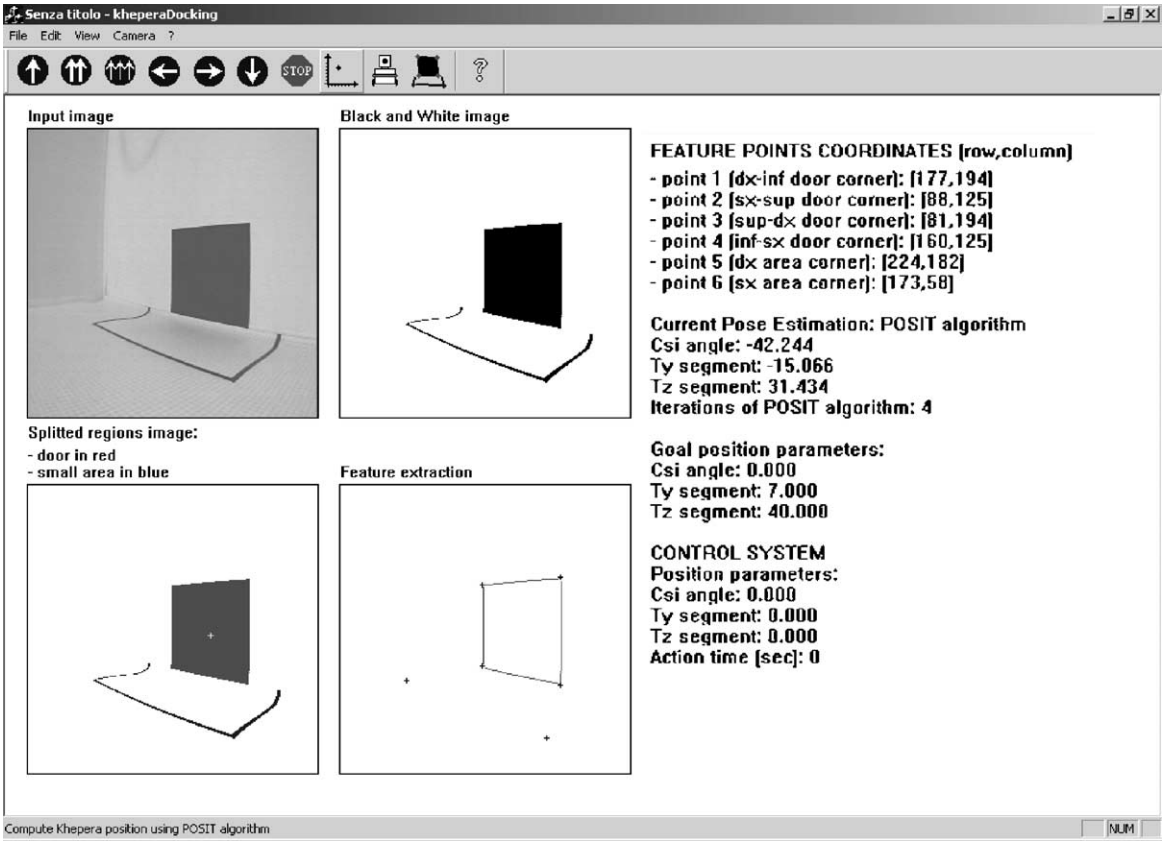


Fig. 11. Output screen of the control system.

For the validation of the autonomous robot positioning two series of tests have been planned and executed. In the former, we have evaluated the convergence neighbourhood in a fixed goal pose starting from a large set of points in the environment. The map of the starting points (Fig. 7) and the correspondent camera orientations are the same used in Section 4. Point 6 is the goal point having the following pose parameters:

$$[\phi, y, z] = [0^\circ, 7\,\text{cm}, 40\,\text{cm}].$$

For each final pose, the camera orientation, defined by ϕ parameter, is represented in Fig. 12 by the line inclination with respect to the z- (horizontal) axis. The black points and the correspondent labels represent the final and the starting points, respectively.

Fig. 12 shows that the convergence neighbourhood in the goal position is smaller than 1.5 cm in diameter and the difference between the current and the desired camera attitude is less than 5°.

An example of the application of the control system is shown in Fig. 13. In each frame of the sequence can be observed the white environment, the black door-area target, the black point marker representing the goal position, the Khepera robot and a white pointer on it indicating its heading.

In the first step (Fig. 13(a)) Khepera, from a generic point in the environment, automatically position the camera to obtain six exact feature points in the image plane. This is realized using an internal control loop, as shown in Fig. 10.

The next step is to retrieve the robot pose using the modified version of the POSIT algorithm. Once the current robot pose $[\phi, T_y, T_z]$ is available, the robot displacement to reach the goal $[\phi_g, T_{yg}, T_{zg}]$ can be easily computed. Since the algorithm gives the

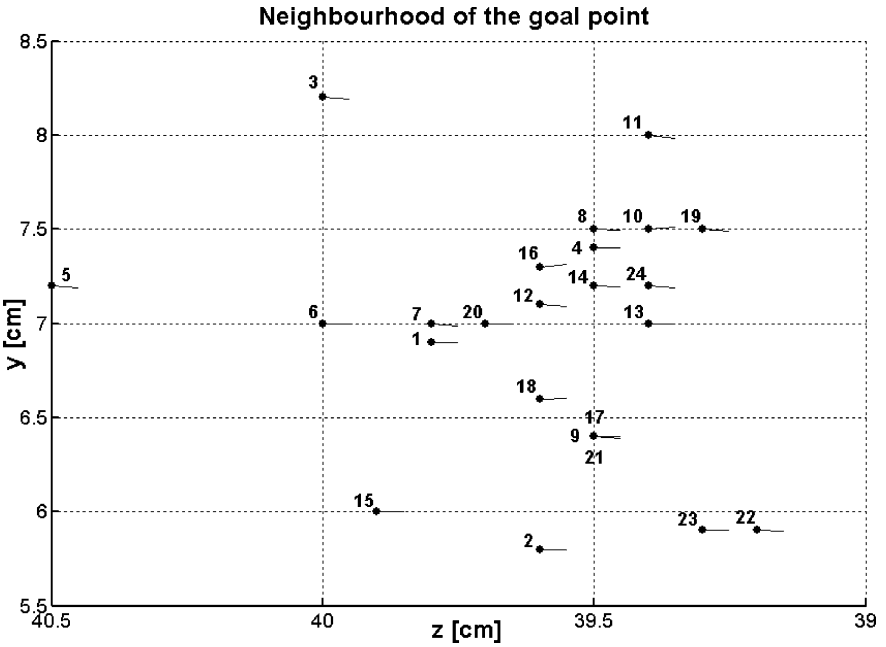


Fig. 12. Convergence neighbourhood in a fixed goal pose starting from any point in the environment.

rotation–translation sequence, the rotation command is executed before the translation one.

Khepera rotates an angle ϕ to steer itself along the z-axis (Fig. 13(b)). Then, it moves along the z-axis for a distance equal to $(T_z - T_{zg})$ (Fig. 13(c)); it rotates 90° (Fig. 13(d)) to move subsequently along the y-axis for a distance equal to $(T_y - T_{yg})$ (Fig. 13(e)). Finally, it rotates to retrieve the goal attitude (Fig. 13(f)).

It then reapplies the POSIT algorithm to obtain an estimation of the just reached position. If this pose is different from the desired one, the robot repeats the previous steps (Fig. 13(g)–(i)).

In the final test we have evaluated the convergence neighbourhood in any point in the environment starting from a fixed pose.

The map of the goal points (Fig. 7) and the correspondent camera orientations are the same used in Section 4. Point 6 is the starting point.

If we indicate with $[e_\phi, e_y, e_z]$ the difference between the final and the desired goal pose, it is possible to notice in Table 5 that the convergence neighbourhood is about 1.5 cm diameter, the difference between the current and the goal camera attitude is less than 5° and the number of iterations is less than 3.

Table 5
Difference between the final and the desired goal pose starting from a fixed point

Goal point	e_ϕ (°)	e_y (cm)	e_z (cm)	Number of iterations
1	5.5	1.6	1.0	2
2	1.9	0.5	0.9	2
3	0.5	0.6	0.9	1
4	0.1	0.0	0.2	2
5	3.7	1.1	0.9	1
6	0.0	0.0	0.0	0
7	0.9	0.7	1.8	3
8	2.9	0.1	0.5	2
9	0.0	0.2	0.0	3
10	0.6	1.6	0.1	3
11	1.8	0.5	0.4	2
12	3.0	1.2	1.0	1
13	0.2	1.1	0.6	2
14	1.0	0.8	0.6	2
15	2.3	1.3	0.2	1
16	1.2	1.3	0.2	2
17	3.7	1.1	0.9	3
18	4.2	1.0	0.3	2
19	2.3	0.7	1.6	3
20	0.5	0.6	0.3	2
21	2.9	1.0	1.4	2
22	1.5	1.1	1.5	1
23	1.4	1.4	0.2	3
24	1.1	0.9	0.4	2

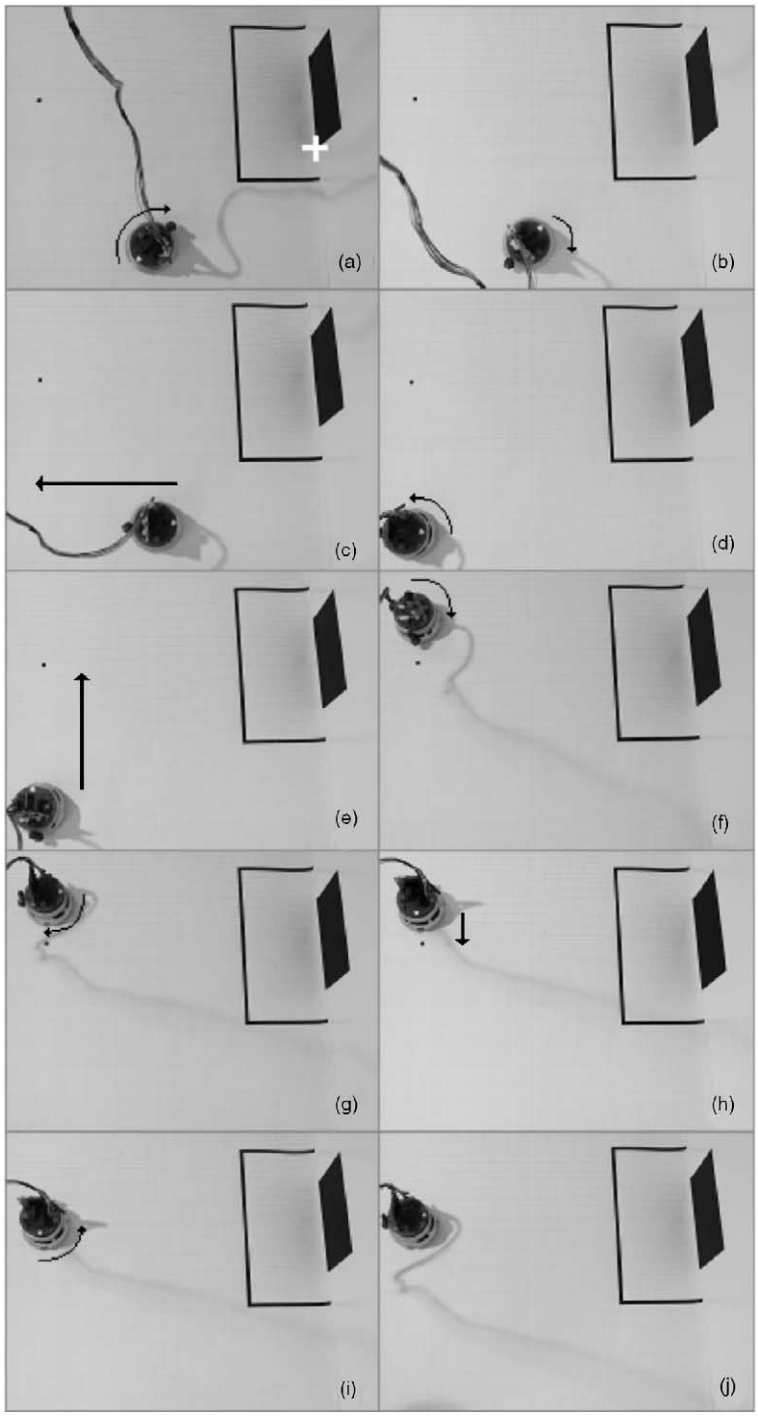


Fig. 13. A sequence showing the Khepera being controlled using the proposed approach. The white pointer shows the heading of the vehicle in each image.

Analyzing the experimental results relative to the robot position reaching, we can conclude that, starting from any point in the environment, the robot is able to reach any goal position, assuming any camera attitude.

6. Conclusion

A new formulation of the POSIT camera pose recovery method has been implemented in the contest of mobile robot indoor autonomous navigation. Due to the specificity of the domain, certain modifications were performed on the original formulation. We have introduced a new formulation of the SOP scaling factor to overcome the problem of low performance due to the camera perspective approximation relative to uncalibrated real images. Moreover, the bounded workspace limits the robot movements control in the autonomous navigation. For this reason, we have modified the formulation of the translation vector in the original algorithm, obtaining the rotation–translation sequence instead of the translation–rotation one.

After preliminary tests to validate the method using synthetic data, we have tested the new version with uncalibrated real images. The results show that these modifications considerably improve the performances of the original method.

Finally, an application of a visual-based control system for the autonomous positioning of a mobile robot with respect to a predefined target has been presented.

The developed system is able, starting from any point in the working space, to drive the vehicle until the whole target appears in the field of view of the camera, to extract and match the feature points to the known geometric model of the target, to calculate the current position of the robot using the modified version of the POSIT algorithm and to reach, using a closed loop control, the desired position after few iterations. The described method can be used even in the case of uncalibrated camera. We have applied two different closed loops: one to obtain the entire visualization of the target in the field of view of the camera and the other to reach a better positioning of the mobile robot. The method has proved to be effective in autonomous positioning during real tests on a Khepera robot, in spite of the poor optical properties of the camera mounted on the vehicle.

References

[1] O. Ait-Aider, P. Hoppenot, E. Colle, Adaptation of Lowe's camera pose recovery algorithm to mobile robot self-localization, Robotica 20 (2002) 385–393.
[2] N. Andreff, B. Espiau, R. Horaud, Visual Servoing from Lines, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.
[3] R. Basri, E. Rivlin, I. Shimshoni, Visual homing: surfing on the epipole, International Journal of Computer Vision 33 (2) (1999) 117–137.
[4] M. Betke, L. Gurvits, Mobile robot localization using landmarks, IEEE Transactions On Robotics And Automation 13 (2) (1997) 251–263.
[5] H.H. Chen, Pose Determination from Line-to-Plane Correspondences: Existence Condition and Closed-Form Solutions, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6) (1991) 530–541.
[6] T. D'Orazio, F.P. Lovergine, M. Ianigro, E. Stella, A. Distante, Mobile robot position determination using visual landmarks, IEEE Transactions on Industrial Electronics 41 (6) (1994) 654–662.
[7] A.J. Davison, Real-time Simultaneous Localisation and Mapping with a Single Camera, in: Proceedings of the International Conference on Computer Vision, Nice, 2003.
[8] D.F. DeMenthon, L.S. Davis, Model-based object pose in 25 lines of code, International Journal of Computer Vision 15 (2) (1995) 123–141.
[9] G.N. DeSouza, A.C. Kak, Vision for mobile robot navigation: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2) (2002) 237–267.
[10] M. Dhome, M. Richetin, J.T. Lapresté, G. Rives, Determination of the attitude of 3D objects from single perspective view, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (12) (1989) 1265–1278.
[11] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.
[12] E. Foxlin, Generalized Architecture for Simultaneous Localization, Auto-Calibration, and Map-building, in: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2–4 October 2002.
[13] T. Gramegna, F.L. Di Alessio, Vision-based robot control, Technical report RI 03/2003 ISSIA-CNR, Bari, Italy, July 2003.
[14] K. Hashimoto (Ed.), Visual Servoing, ISBN: 981-02-1364-6, World Scientific, 1993.
[15] D. Jacobs, R. Basri, 3-D to 2-D pose determination with regions, International Journal of Computer Vision 34 (2/3) (1999) 123–145.
[16] E. Krotkov, Mobile Robot Localization Using a Single Image, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, 1989, pp. 978–983.
[17] J. Lessard, D. Laurendeau, Estimation of the position of a robot using computer vision for a live-line maintenance task,

in: Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, NC, 1987, pp. 1203–1208.
[18] Y. Liu, T.S. Huang, O.D. Faugeras, Determination of camera location from 2-D to 3-D line and point correspondences, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1) (1990) 28–37.
[19] I. Lopez, A. Frangi, B. van der Wijst, H. Broers, Pose estimation from 2D to 3D for computer vision in an assembly node, Technical Report CTB500-02-0000, Philips CFT, Eindhoven, The Netherlands.
[20] D.G. Lowe, Fitting parameterized three-dimensional models to images, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (5) (1991) 441–450.
[21] D.G. Lowe, Perceptual Organization and Visual Recognition, Kluwer Academic Publishers, Boston, MA, 1985.
[22] D.G. Lowe, Three-dimensional object recognition from single two dimensional images, Artificial Intelligence 31 (3) (1987) 355–395.
[23] E. Malis, Survey of vision-based robot control, ENSIETA European Naval Ship Design Short Course, Brest, France, 2002.
[24] P. Puget, T. Skordas, Calibrating a mobile camera, Image and Vision Computing 8 (4) (1990) 341–348.
[25] L. Quan, Z. Lan, Linear N-point camera pose determination, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (8) (1999) 774–780.
[26] E. Stella, A. Distante, Self-location of a mobile robot by estimation of camera parameters, Robotics and Autonomous Systems 15 (1995) 179–187.
[27] K. Sugihara, Some location problems for robot navigation using a single camera, Computer Vision Graphics and Image Processing 42 (1988) 112–129.
[28] R. Talluri, J.K. Aggarwal, Position Estimation Techniques for an Autonomous Mobile Robot—A Review, in: C.H. Chen, L.F. Pau, P.S.P. Wang (Eds.), The Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Co., Singapore, 1993, pp. 769–801.
[29] R.Y. Tsai, A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE Journal of Robotics and Automation 3 (4) (1987) 323–344.
[30] R.Y. Tsai, R.K. Lenz, A new technique for fully autonomous and efficient 3D robotics hand-eye calibration, in: Proceedings of the Fourth International Symposium on Robotics Research, vol. 4, 1987, pp. 287–297.
[31] L. Venturino, T. Gramegna, G. Cicirelli, G. Attolico, Autonomous robot positioning by visual servoing, in: Proceedings of the Third IASTED International Conference on Artificial Intelligence and Applications, Benalmádena, Spain, 8–10 September 2003.
[32] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, Technical Report 2273, INRIA France, May 1994.

Tommaso Gramegna was born in Bari in 1973. He received the degree in Electronic Engineering at Politecnico di Bari (Bari, Italy) in 2001. From 2001 to 2002 he was contract researcher at the Politecnico di Bari. He is currently researcher at the Institute on Intelligent Systems for Automation of the National Research Council. His research interests are focused on computer vision, image analysis and 3D modelling with application to robotics.

Lea Venturino was born in Bari, Italy in 1976. She received the degree in Electronic Engineering at Politecnico di Bari (Bari, Italy) in 2001. From February 2002 she works as researcher at Institute of Intelligent Systems for Automation (ISSIA) of the Italian National Research Council (CNR). Her research interests are focused on autonomous robot navigation, image analysis and application of computer vision to robotics.

Grazia Cicirelli obtained the degree in Computer Science from the University of Bari (Italy) in 1994. From 1995 to 2001 she has been a young researcher at the Institute for Signal and Image Processing in Bari, under grant of the Italian National Research Council (CNR). She is currently a technologist researcher at the Institute of Intelligent Systems for Automation of the CNR in Bari. Her current research interests cover vision-based robotics, pattern recognition, neural network and reinforcement learning.

Giovanni Attolico was born in Bari in 1961. He received the degree in Computer Science at University of Bari (Italy) in 1986. He is currently senior researcher at the Institute on Intelligent Systems for Automation of the National Research Council. His main interests are in Image Analysis, Computer Vision and 3D Modelling with application to robotics, visual inspection and image retrieval. He is a member of the IAPR.

Arcangelo Distante received the degree in Computer Science from the University of Bari (Italy) in 1976. He joined the National Nuclear Physics Institute until 1983 where he worked on various theoretical and computation aspects of 3D reconstruction and pattern recognition of nuclear events. Since 1984 he has been working with the Institute for Signal and Image Processing (IESI) of Italian National Research Council (CNR). Currently he is the coordinator of the Robot Vision Group and the Director of the Institute of Intelligent System for Automation (ISSIA—CNR). In 1996 he joined the University of Lecce where he is associate professor in Theory and Practice of Image Processing at the Faculty of Engineering. His current research deals with computer vision, pattern recognition, machine learning, neural computation, robot navigation and architectures for computer vision. Dr. Arcangelo Distante is a member of IAPR and SPIE.