Sensors **2010**, *10*, 3655 - 3680; doi:10.3390/s100403655

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Localization of Mobile Robots Using Odometry and an External Vision Sensor

Daniel Pizarro *, Manuel Mazo, Enrique Santiso, Marta Marron, David Jimenez, Santiago Cobreces and Cristina Losada

Department of Electronics, University of Alcala, NII km 33,600, Alcala de Henares, Spain;
E-Mails: mazo@depeca.uah.es (M.M.); santiso@depeca.uah.es (E.S.); marta@depeca.uah.es (M.M.); david.jimenez@depeca.uah.es (D.J.); cobreces@depeca.uah.es (S.C.); losada@depeca.uah.es (C.L.)

* Author to whom correspondence should be addressed: E-Mail: pizarro@depeca.uah.es;
Tel.: +34-918-856-582. Fax: +34-918-856-591.

Received: 15 January 2010; in revised form: 3 March 2010 / Accepted: 31 March 2010 / Published: 13 April 2010

Abstract: This paper presents a sensor system for robot localization based on the information obtained from a single camera attached in a fixed place external to the robot. Our approach firstly obtains the 3D geometrical model of the robot based on the projection of its natural appearance in the camera while the robot performs an initialization trajectory. This paper proposes a structure-from-motion solution that uses the odometry sensors inside the robot as a metric reference. Secondly, an online localization method based on a sequential Bayesian inference is proposed, which uses the geometrical model of the robot as a link between image measurements and pose estimation. The online approach is resistant to hard occlusions and the experimental setup proposed in this paper shows its effectiveness in real situations. The proposed approach has many applications in both the industrial and service robot fields.

Keywords: robotics; vision sensor; intelligent spaces

1. Introduction

This paper presents a sensor system composed of a single camera attached to a fixed position and orientation in a bounded environment (indoor workplace) which is observing a mobile robot. The aim of the sensor system is to obtain the orientation and position (*i.e.*, pose) of a mobile robot using both visual information retrieved by the camera and relative odometry readings obtained from the internal sensors of the robot. The camera acquisition and image processing tasks are executed in a specialized hardware, which also controls the behavior and internal sensors of the mobile robot through a wireless channel. The proposed schema allows the robot to perform complex tasks without requiring dedicated processing hardware on it. This approach is sustained in the Intelligent Space [1] concept and it can be equally applied to multiple scenarios, specially both in the industrial field (e.g., automatic crane positioning, autonomous car parking) and in the service fields (e.g., wheelchair positioning in medical environments or autonomous driving of mobile platforms). The single camera solution presented in this paper allows to cover large areas with less cameras compared to multiple camera configurations where overlapped areas are mandatory. This feature reduces the cost and improves the reliability of the intelligent space philosophy.

In this paper, we suppose that the camera is correctly calibrated and thus the parameters governing the projection model of the camera are previously known. To connect the pose of the robot with information found in the image plane of the camera, we propose to obtain a 3D geometric model of the mobile robot. Such model is composed of several sparse points whose coordinates represent some well-localized points belonging to the physical structure of the robot. These points are determined by image measurements, called image features, which usually correspond to corner-like points due to texture changes or geometry changes such as 3D vertexes. Usually in industrial fields the image features are obtained by including some kind of artificial marker on the structure of the robot (infrared markers or color bands). These methods are very robust and can be used to recognize human activity and models with high degrees of freedom (AICON 3D online [2] or ViconPeak online systems [3]). However, in this paper we want to minimize the required “a priori” knowledge of the robot, so that it is not necessary to place artificial markers or beacons on it to detect its structure in the images.

Independent of the nature of the image features, the information obtained from a camera is naturally ambiguous and thus some sort of extra metric information are required in order to solve such ambiguity. In this work, we propose to use the odometry sensors inside the robot (*i.e.*, wheel velocities) to act as the metric reference.

The general diagram of the algorithm proposed in this paper is shown in Figure 1. It shows a clear division of the processes involved in obtaining the pose of the robot: first we denote as “Initialization of Pose and Geometry” to those processes necessary to start up the system, such as the 3D model of the robot and the initial pose it occupies. The initialization consists of a batch processing algorithm where the robot is commanded to follow a certain trajectory so that the camera is able to track some points of the robot’s structure under different viewpoints jointly with the recording of the odometry information. All this information is combined to give the 3D model of the robot and the initial pose it occupies.

Given the initialization information, the second group of processes, named “Sequential Localization”, provides the pose of the robot in a sequential manner. It is composed of a pose estimator, given odometry readings and a pose correction block which combines the estimation of the pose with image measurements to accurately give a coherent pose with the measurements. This algorithm operates entirely on-line and thus the pose is available at each time sample.

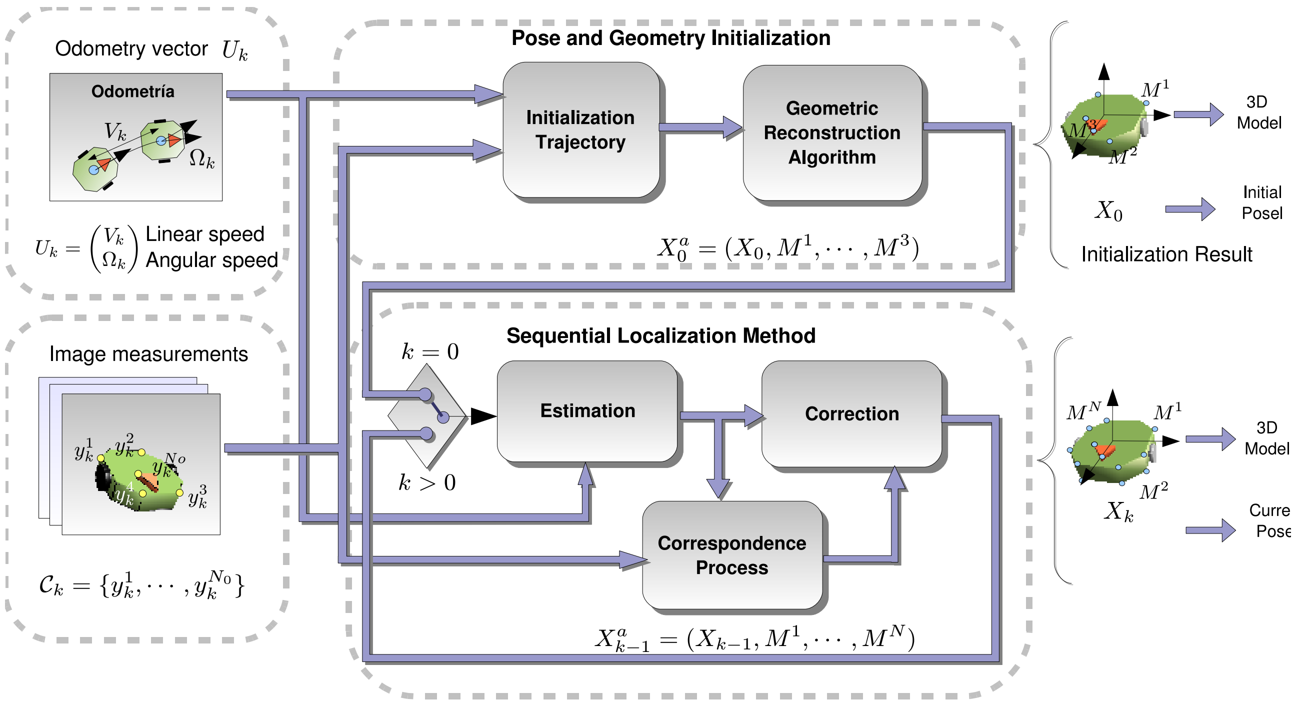
Both group of processes are supplied with two main sources of information:

1. Image measurements: they consist of the projection in the camera’s image plane of certain

points of the robot's 3D structure. The measurement process is in charge of searching coherent correspondences through images with different perspective changes due to the movement of the robot.

2. Motion estimation of the robot: The odometry sensors built on-board the robot supply the localization system with an accurate motion estimation in short trajectories but that is prone to accumulative errors in large ones.

Figure 1. General diagram of the proposed localization system using a vision sensor and odometry readings.


Odometry vector U_k
Odometría
V_k
Ω_k
$U_k = \begin{pmatrix} V_k \\ \Omega_k \end{pmatrix}$ Linear speed Angular speed
Pose and Geometry Initialization
Initialization Trajectory
Geometric Reconstruction Algorithm
$X_0^a = (X_0, M^1, \cdots, M^3)$
M^1
M^3
M^2
3D Model
X_0
Initial Posel
Initialization Result
Image measurements
y_k^1
y_k^2
y_k^{No}
y_k^3
$\mathcal{C}_k = \{y_k^1, \cdots, y_k^{No}\}$
Sequential Localization Method
$k = 0$
$k > 0$
Estimation
Correction
Correspondence Process
$X_{k-1}^a = (X_{k-1}, M^1, \cdots, M^N)$
M^N
M^1
M^2
X_k
3D Model
Curre Pose

1.1. Previous Works

Despite the inherent potential of using external cameras to localize robots, there are relative few attempts to solve it compared to the approach that consider the camera on-board the robot [4, 5]. However, some examples of robot localization with cameras can be found in the literature, where the robot is equipped with artificial landmarks, either active [6, 7] or passive ones [8, 9]. In other works a model of the robot, either geometrical or of appearance [10, 11], is learnt previously to the tracking task. In [12, 13], the position of static and dynamic objects is obtained by multiple camera fusion inside an occupancy grid. An appearance model is used afterwards to ascertain which object is each robot. Despite the technique used for tracking, the common point of many of the proposals found in the topic comes from the fact that rich knowledge is obtained previously to the tracking, in a supervised task.

In this paper the localization of the robot is obtained in terms of its natural appearance. We propose a full metric and accurate system based on identifying natural features belonging to the geometric structure of the robot. On most natural objects we can find points whose image projection can be tracked in the image plane independently of the position the object occupies and based on local properties found in the image (*i.e.*, lines, corners or color blobs). Those points are considered natural markers, as they serve as reference points in the image plane that can be easily relate with their three-dimensional counterparts. The set of methods focused on tracking natural markers have become a very successful and deeply studied topic in the literature [14, 15], as they represent the basic measurements of most of existing reconstruction methods.

Scene reconstruction from image measurements is a classic and mature discipline in the computer vision field. Among the wide amount of proposals it can be highlighted those grouped under the name "Bundle Adjustment" [16, 17]. Their aim is essentially to estimate, jointly and optimally, the 3D structure of a scene and the camera parameters from a set of images taken under some kind of motion. (*i.e.*, it can be the camera that moves or equally some part of the scene w.r.t. the camera).

In general terms, Bundle Adjustment reconstruction methods are based in iterative optimization methods which try to minimize the image discrepancy between the measured positions of the 3D model and the expected ones using the last iteration solution. The discrepancy balances the contribution of the measurements into the final solution and plays an important role in this paper. Our main contribution is a redefinition of the discrepancy function using a Maximum Likelihood approach which takes into account the statistical distribution of the error. This distribution is especially affected by the odometry errors which are accumulative in long trajectories.

On the other hand, once a geometric model is obtained using a structure-from-motion approach, its pose with respect to a global coordinate origin can be easily retrieved by measuring the projection of the model in the image plane. This problem, commonly known as the Perspective n Point Problem (PnP), has received considerable attention in the literature, where some accurate solutions are found such as [18] or the recent global solution proposed in [19]. In this paper we instead follow a filtering approach, where not only image measurements but also last time pose information and odometry information are used to obtain the pose. This approach, which is based on the use of a Kalman Filter, is much more regular than solving the PnP problem at each time instant and will be described in this paper.

The paper is organized as follows. In Section 2. the notation and mathematical elements used in the paper are described. In Section 3. the description of the initialization algorithms is given. The online Kalman loop is explained in Section 4. and some results in a real environment are shown in Section 5. Conclusions are in Section 6.

2. Definitions and Notation

This section presents a description of the symbols and mathematical models used in the rest of the paper.

2.1. *Robot and Image Measurements*

The robot's pose at time k is described by a vector X_k. We suppose that the robot's motion lie on the plane $z = 0$ referred from the world coordinate origin O_W (See Figure 2). The pose vector X_k is described by 3 components (x_k, y_k, α_k), corresponding to two position coordinates x_k, y_k and the rotation angle α_k in the z axis. The motion model $X_k = g(X_{k-1}, U_k)$ defines the relationship between the current pose X_k with respect to the previous time one X_{k-1} and the input U_k given by odometry sensors inside the robot. In our proposal the robot used is a differential wheeled platform and thus $U_k = (Vl_k, \Omega_k)^T$, where Vl_k and Ω_k describe the linear and angular speed of the robot's center of rotation (O_R in Figure 2) respectively. The motion model g is then very simple in function of the discretized linear speed Vl_k and the angular speed Ω_k.

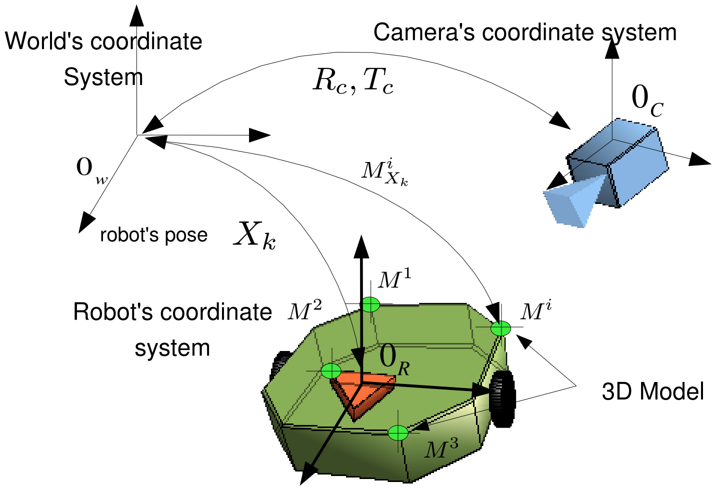
The robot's geometry is composed of a sparse set of N 3D points $\mathcal{M} = \{M^1, \cdots, M^N\}$ referred from the local coordinate origin O_R described by robot's pose X_k. The points M^i are static in time due to robot's rigidness, and thus, no temporal subindex is required for them. Function $M^i_{X_k} = t(X_k, M^i)$ uses actual pose X_k to express M^i in the global coordinate origin O_W that X_k is referred to (see Figure 2):

$$M^i_{X_k} = t(X_k, M^i) = R_k M^i + T_k \tag{1}$$

where:

$$R_k = \begin{pmatrix} \cos(\alpha_k) & \sin(\alpha_k) & 0 \\ -\sin(\alpha_k) & cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad T_k = \begin{pmatrix} x_k \\ y_k \\ 0 \end{pmatrix} \tag{2}$$

Figure 2. Spatial relationship between world's coordinate origin O_W, robot's coordinate origin O_R and camera's coordinate origin O_C.



The augmented vector X^a_k, which is the state vector of the system, is defined as the concatenation in one column vector of both the pose X_k and the set of static structure points $\mathcal{M}$:

$$X^a_k = \left((X_k)^T \quad (M^1)^T \quad \cdots \quad (M^N)^T \right)^T \tag{3}$$

An augmented motion model g^a is defined as the transition of the state vector:

$$X^a_k = g^a(X^a_{k-1}, U_k) = \left((g(X_{k-1}, U_k))^T \quad (M^1)^T \quad \cdots \quad (M^N)^T \right)^T \tag{4}$$

It must be noticed that the motion model g^a leaves the structure points contained in the state vector untouched, as we suppose that the object is rigid.

The whole set of measurements obtained in the image plane of the camera at time k is denoted by Y_k. Each measurement inside Y_k is defined by a two-dimensional vector $y_k^i = \begin{pmatrix} u_k^i & v_k^i \end{pmatrix}^T$, describing the projection of each point inside $\mathcal{M}$ in the image plane, where i stands for the correspondence with the point M^i. It must be remarked that, in general, Y_k does not contain the image projection of every point in the set M as some of them can be occluded depending on the situation.

The camera is modeled with a zero-skew "pin-hole" model (see [17] for more details), with the following matrix K_c containing the intrinsic parameters:

$$K_c = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

The extrinsic parameters of the camera (*i.e.*, position and orientation of the camera w.r.t. O_W) are described using the rigid transformation R_c, T_c (*i.e.*, rotation matrix and translation vector). The matrix R_c is defined by three Euler angles $\alpha_c, \beta_c, \gamma_c$. The vector T_c can be decomposed into its three spatial coordinates T_x, T_y, T_z. All calibration parameters can be grouped inside vector P:

$$P = (f_u, f_v, u_0, v_0, \alpha_c, \beta_c, \gamma_c, T_x, T_y, T_z)^T \tag{6}$$

Given a single measurement y_k^i from the set Y_k, its 2D coordinates can be expressed using the "pin-hole" model and the aforementioned calibration parameters:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda_k^i K_c \left(R_c M_{X_k}^i + T_c\right) = \lambda_k^i K_c \left(R_c R_k M^i + R_c T_k + T_c\right) \tag{7}$$

where λ_k^i represents a projective scale factor that can be obtained so that the third element of the right part of Equation (7) is equal to one. It is important to remark that the projection model, although simple, depends in a non-linear fashion w.r.t. M^i due to the factor λ_k^i. For compactness the projection model of Equation (7) can be described as the function h:

$$y_k^i = h(X_k, M^i, P) \tag{8}$$

In the same way, the whole vector Y_k can be expressed with the following function:

$$Y_k = h^a(X_k^a, P) = \begin{pmatrix} h(X_k, M^1, P)^T & \cdots & h(X_k, M^N, P)^T \end{pmatrix}^T \tag{9}$$

2.2. *Random Processes*

This paper explicitly deals with uncertainties by describing every process as a random variable with a Gaussian distribution whose covariance matrix stands for its uncertainty. The random processes are expressed in bold typography (*i.e.*, $\mathbf{X_k}$ is the random process describing the pose and X_k is a realization of it) and each of them are defined by a mean vector and a covariance matrix. Therefore $\mathbf{X_k^a}$ is defined by its mean $\hat{X}_k^a$ and covariance Σ_k^a (or simplifying $\mathcal{N}(\hat{X}_k^a, \Sigma_k^a)$). The following processes are considered in the paper:

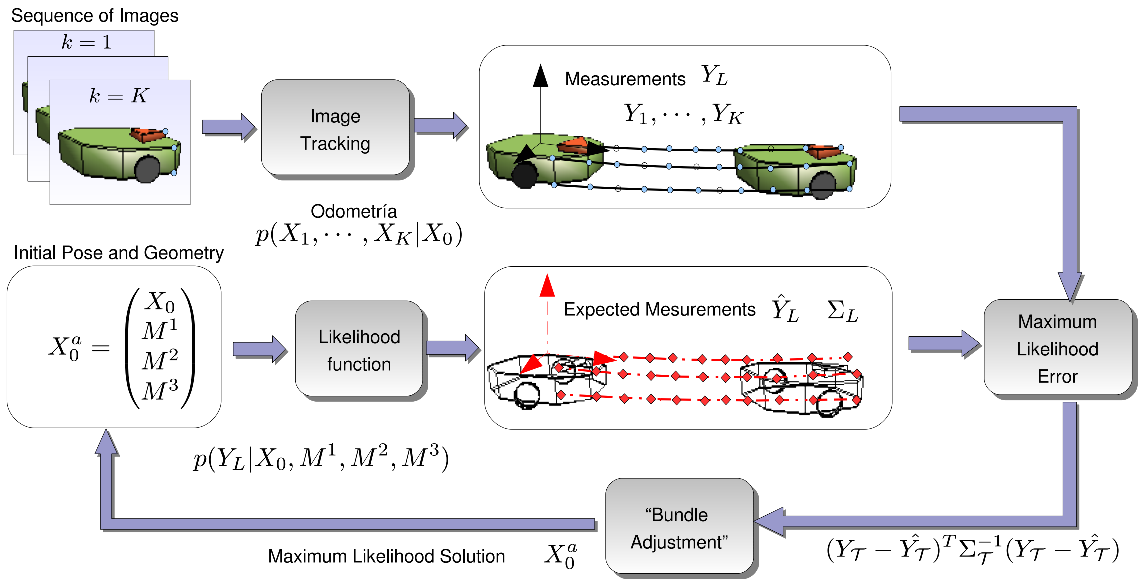
- Pose $\mathbf{X_k} = \mathcal{N}(\hat{X}_k, \Sigma_k)$ and 3D model $\mathbf{M} = \mathcal{N}(\hat{M}, \Sigma_M)$ processes. Its joint distribution is encoded in $\mathbf{X_k^a} = \mathcal{N}(\hat{X}_k^a, \Sigma_k^a)$.
- Measurement process $\mathbf{Y_k} = \mathcal{N}(\hat{Y}_k, \Sigma_{Y_k})$, whose uncertainty comes from errors in image detection.
- Odometry input values $\mathbf{U_k} = \mathcal{N}(\hat{U}_k, \Sigma_{U_k})$, which are polluted by random deviations.

3. Initialization Process

The initialization step consists of obtaining the initial pose the robot occupies and a 3D model of its structure using visual features (*i.e.*, to obtain an initial guess of X_0^a). The importance of this step is crucial, as the robot's geometric model serves as the necessary link between robot's pose and image measurements. The initialization allows to use afterwards the online approach presented in Section 4.

A delayed initialization approach is proposed, based on collecting image measurements and the corresponding odometry position estimation along a sequence of time (*i.e.*, $k = 1, \cdots, K$). After taking a sufficient amount of measurements of the object in motion, an iterative optimization method is used to obtain the best solution for X_0^a according to a cost criterion. The odometry readings are used in this step as metric information, which allows to remove the natural ambiguity produced by measurements from a single camera. The main problem of using odometry as a metric reference for reconstruction comes from the accumulative error it presents, which is a very well-known problem in dead-reckoning tasks.

Figure 3. Maximum likelihood initialization by means of reducing the residual of the expected measurements (red diamonds) and the measured image trajectories (blue circles).


Sequence of Images
$k = 1$
$k = K$
Image Tracking
Measurements Y_L
$Y_1, \cdots, Y_K$
Odometría
$p(X_1, \cdots, X_K | X_0)$
Initial Pose and Geometry
$X_0^a = \begin{pmatrix} X_0 \\ M^1 \\ M^2 \\ M^3 \end{pmatrix}$
Likelihood function
Expected Mesurements $\hat{Y}_L$ Σ_L
Maximum Likelihood Error
$p(Y_L | X_0, M^1, M^2, M^3)$
"Bundle Adjustment"
$(Y_\mathcal{T} - \hat{Y}_\mathcal{T})^T \Sigma_\mathcal{T}^{-1} (Y_\mathcal{T} - \hat{Y}_\mathcal{T})$
Maximum Likelihood Solution
X_0^a

The initialization algorithm consists of a Maximum Likelihood (M.L.) estimation of the vector X_0^a, which does not estimate the initialization value itself but its equivalent Gaussian distribution $\mathbf{X_0^a} = \mathcal{N}(\hat{X}_0^1, \Sigma_0^a)$ by using a "Bundle Adjustment" method (See Figure 3). The M.L. approach (see [16, 17] for more details) allows to properly tackle the growing drift present in the odometry estimation by giving fair less importance or weight to those instants where the odometry is expected to be more uncertain.

In the following sections, the initialization algorithm is explained in detail.

3.1. *Odometry Estimation*

A set of odometry readings, $U_1, \cdots, U_K$ are collected over a set of K consecutive time samples $k = 1, \cdots, K$, which corresponds to the initialization trajectory performed by the robot. Using a motion model g given initial position X_0, an estimation of the robot's pose in the whole initialization sequence is obtained as follows:

$$\begin{aligned} \mathbf{X_1} &= g(\mathbf{X_0}, \mathbf{U_1}) \\ \mathbf{X_2} &= g(\mathbf{X_1}, \mathbf{U_2}) \\ &\vdots \\ \mathbf{X_K} &= g(\mathbf{X_{K-1}}, \mathbf{U_K}) \end{aligned} \tag{10}$$

where we recall that $\mathbf{X_k}$ and $\mathbf{U_k}$ denote Gaussian processes. Using expression (10), and by propagating the statistical processes through function g, the joint distribution $p(X_1, \cdots, X_K|X_0)$ can be obtained. The propagation of statistical processes through non-linear functions is in general a very complex task, as it requires to solve integral equations. However in this paper we will follow a first order approximation of the non-linear functions centered in the mean of the process (see [20] for more details). Using this technique the Gaussian processes $\mathbf{X_k}$ and $\mathbf{U_k}$ can be iteratively propagated through function g at the cost of being an approximation. This paper shows that despite being an approximation this approach end in a good estimation of the initialization vector.

By denoting as X to the joint vector containing all poses $X = (X_1^T, \cdots, X_K^T)^T$, the joint distribution of all poses, given the initial state X_0, $p(X|X_0)$ is approximated as Gaussian p.d.f. with mean $\hat{X}$ and covariance matrix Σ_X.

3.2. *Image Measurements*

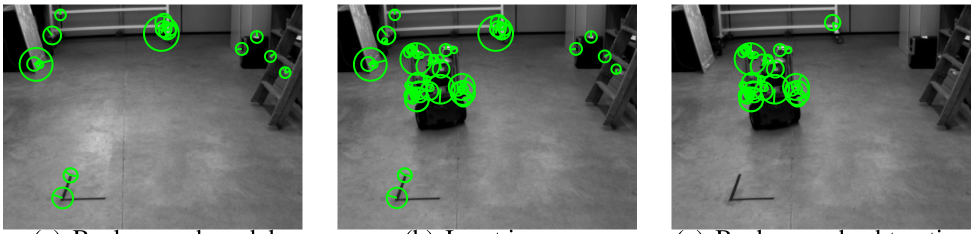
In this step we describe how to collect the position in the image plane of different points from the robot's structure during the initialization trajectory. We base our method in the work [15] where the SIFT descriptor is introduced. The process used in the initialization is composed of two steps:

- Feature-based background subtraction.

  We describe the static background of the scene using a single image, from which a set of features, $\mathcal{F}_b = \{y_b^1, \cdots, y_b^{Nb}\}$ and its correspondent SIFT descriptors $\mathcal{D}_1 = \{d_1^1, \cdots, d_1^{Nd}\}$ are obtained. The sets $\mathcal{D}_b$ and $\mathcal{F}_b$ are our feature-based background model.

  Given an input image, namely I_k, we find the sets $\mathcal{D}_k$ and $\mathcal{F}_k$. We consider that a feature $y_k^i \in \mathcal{F}_k, d_k^i \in \mathcal{D}_k$ belongs to the background if we can find a feature $y_b^j \in \mathcal{F}_b, d_b^j \in \mathcal{D}_b$ such that $|y_k^i - y_b^j| < R_{max}$ and $|d_k^i - d_b^j| < d_{max}$. This method although simple shows to be very effective and robust in real environments (See Figure 4).

Figure 4. Feature-based background subtraction method.



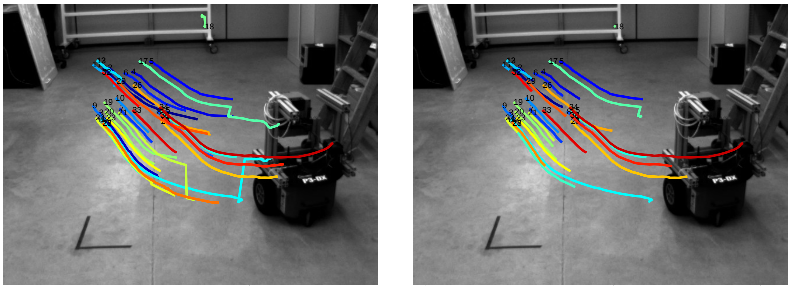
(a) Background model (b) Input image (c) Background subtraction

- Supervised tracking algorithm.

  Given the first image of the initialization trajectory, namely I_1, we obtain the set of N features $\mathcal{F}_1$, once they are cleaned by the aforementioned background subtraction method. We propose a method to track them in the whole initialization trajectory.

  By only using SIFT descriptors and its tracking in consecutive frames does not produce stable tracks, specially when dealing with highly irregular objects where many of the features are due to corners. We thus propose to use a classical feature tracking approach based on the Kanade–Lucas–Tomasi (KLT) algorithm [14]. To avoid degeneracy in the tracks, which is a very common problem in those kind of trackers, we use the SIFT descriptors to remove those segments of the tracks that clearly do not correspond to the feature tracked. This can be done easily by checking distance between the descriptors in the track. The threshold used must be chosen experimentally so that it does not eliminate useful parts of the tracks. In Figure 5a we can see the tracks obtained by the KLT tracker without degeneracy supervision. In Figure 5b the automatically removed segments are displayed.

Figure 5. Comparison between the KLT tracker and the SIFT-supervised KLT version.



(a) KLT tracking (b) SIFT-supervised tracking algorithm

3.3. *Likelihood Function*

Using the feature-based algorithm proposed before a set of measurements in the whole trajectory, $Y_1, \cdots, Y_K$, is obtained, where each vector Y_k contains the projection of N points from robot's structure at time sample k. The set of N points, $M^1, \cdots, M^N$, jointly with the initial pose X_0, represent the initialization unknowns. As the robot's motion does not depend of its structure, the following statistical independence is true:

$$p(X_1, \cdots, X_K|X_0) = p(X_1, \cdots, X_K|X_0, M^1, \cdots, M^N) = p(X_1, \cdots, X_K|X_0^a) \tag{11}$$

If all trajectories are packed into vector Y_L, the following function put in relation Y_L with the distribution of expression (11):

$$\mathbf{Y_L} = \begin{pmatrix} \mathbf{Y_1} \\ \vdots \\ \mathbf{Y_K} \end{pmatrix} = \begin{pmatrix} h^a(\mathbf{X_1^a}, P) + \mathbf{V_1} \\ \vdots \\ h^a(\mathbf{X_K^a}, P) + \mathbf{V_K} \end{pmatrix} \tag{12}$$

where $\mathbf{V_k}$ represents the uncertainty in image detection. Using distribution showed in (11), and propagating statistics through function (12), the following likelihood distribution is obtained:

$$p(Y_L|X_0, M^1, \cdots, M^N) = p(Y_L|X_0^a) \tag{13}$$

The likelihood function (13) is represented by a Gaussian distribution using a first order approximation of (12). It is thus defined by a mean $\hat{Y}_L$ and a covariance matrix Σ_L:

$$p(Y_L|X_0^a) = \frac{1}{\sqrt{|\Sigma_L|2^N}} e^{\frac{1}{2}(Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L)} \tag{14}$$

3.4. Maximum Likelihood Approach

The likelihood function (14), parametrized by its covariance matrix Σ_L and its mean $\hat{Y}_L$, is dependent of the conditional parameters and unknown values X_0^a.

The "Maximum Likelihood" estimation consists of finding the values for X_0^a that maximize the likelihood function:

$$\max_{X_0, M^1, \cdots, M^N} p(Y_L|X_0, M^1, \cdots, M^N) \tag{15}$$

In its Gaussian approximation and by taking logarithms, it is converted into the following minimization problem:

$$\min_{X_0, M^1, \cdots, M^N} (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \tag{16}$$

where Y_L is the realization of the process (*i.e.*, the set of measurements from the image) and $\hat{Y}_L$ are the expected measurements given a value of the parameters $X_0, M^1, \cdots, M^N$.

The configuration of Σ_L is ruled by the expected uncertainty in the measurements and the statistical relation between them. Usually all cross-correlated terms of Σ_L are non-zero, which has an important effect in the sparsity of the Hessian used inside the optimization algorithm and consequently in the computational complexity of the problem (see [21] for more details).

The covariance matrix Σ_L can be approximated assuming either independence between time samples (discarding cross-correlation terms between time indexes) or total independence between time and each measurement (discarding all cross-correlation terms except the 2×2 boxes belonging to a single measurement). In Table 1, the different cost functions are shown depending on the discarded terms of Σ_L. The different approximations of Σ_L have a direct impact in the accuracy and the reconstruction error, and the results will be shown in Section 5.

Table 1. Different cost functions depending on the approximations of Σ_L.

Type of Approximation	cost function
Complete Correlated matrix Σ_L (M.C.)	$(Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L)$
$2N \times 2N$ block approximation of Σ_L (M.B.)	$\sum_{k=1}^{K} (Y_k - \hat{Y}_k)^T \Sigma_{Y_k}^{-1} (Y_k - \hat{Y}_k)$
2×2 block approximation of Σ_L (M.D.)	$\sum_{k=1}^{K} \sum_{i=1}^{N} (y_k^i - \hat{y}_k^i)^T \Sigma_{y_k^i}^{-1} (y_k^i - \hat{y}_k^i)$
Identity approximation of Σ_L (M.I.)	$\sum_{k=1}^{K} \sum_{i=1}^{N} (y_k^i - \hat{y}_k^i)^T (y_k^i - \hat{y}_k^i)$

Intuitively, if Σ_L results to be a identity matrix, the cost function is reduced to a simple image residual minimization extensively used in Bundle Adjustment techniques, where in principle all cost differences $y_k^i - \hat{y}_k^i$ have equal importance:

$$\min_{X_0^a} \sum_{i=1}^{N} \sum_{k=1}^{K_i} |(y_k^i - \hat{y}_k^i)|^2 \tag{17}$$

The result of minimizing Equation (16) instead of the usual (17) show significant improvements in the reconstruction error. In Section 5. a comparison is shown between the initialization accuracy under the different assumptions of the matrix Σ_L, from its diagonal version to the complete correlated form. The minimum of (16) is obtained using the Levenberg–Mardquardt [17] iterative optimization method.

We suppose that during the measurement step there is a low probability of encountering outliers in the features. This argument can be very optimistic in some real configurations where more objects appear in the scene and produce occlusions or false matchings. For those cases, all cost functions presented in this section can be modified to be robust against outliers by using M-Estimators. We refer the reader to [17] for more details.

3.5. Initialization before Optimization

The use of an iterative optimization method for obtaining the minimum of (16) requires a setup value from which to start iterating. An initial estimation of X_0^a close to its correct value reduces the probability to fall in a local extrema of the cost function.

The method proposed in this paper to give an initial value for X_0^a consists of a non-iterative and exact method, which gives directly an accurate solution for X_0^a in absence of noise in odometry values. This method is based on the assumption that the robot moves in a plane and thus only the angle over its Z axis is needed. As explained below, this assumption allows to solve the problem with a rank deficient Linear Least Squares approximation, which is solved using the Singular Value Decomposition of the system's matrix and imposing a constraint that ensures a valid rotation. Its development is briefly introduced next.

For a point M^i of robot's model and at time k of the initialization trajectory, the image measurement

y_k^i results from the following projection:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda_k^i K_c (R_c (R_k^\Delta R_0 M^i + T_0 + R_0 T_k^\Delta) + T_c) \tag{18}$$

where the matrix R_k^Δ and vector T_k^Δ represent the rotation and position of the robot in the floor plane given by the odometry supposing that $X_0 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$. The rotation matrix R_0 and the offset T_0 correspond with the initial pose $X_0 = (x_0^1, x_0^2, \alpha_0)$ in form of rigid transformation:

$$R_0 = \begin{pmatrix} cos(\alpha_0) & -sin(\alpha_0) & 0 \\ sin(\alpha_0) & cos(\alpha_0) & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad T_k = \begin{pmatrix} x_0^1 \\ x_0^2 \\ 0 \end{pmatrix} \tag{19}$$

where R_0 is a non-linear function of the orientation angle. The expression (18) depends non-linearly of vector X_0^a and so a new parametrization is proposed jointly with a transformation which removes the projective parameter λ_k^i.

The point M^i is replaced by the rotated $M_{X_0}^i = R_0 M^i$, removing thus the product between unknowns. The orientation angle in the pose is substituted by parameters $a = cos(\alpha_0)$ and $b = sin(\alpha_0)$, with the constraint $a^2 + b^2 = 1$. Using the new parameterization the expression (18) is transformed in the following:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda K_c (R_c (R_k^\Delta M_{X_0}^i + T_0 + L_{T_k^\Delta} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}) + T_c) \tag{20}$$

with $L_{T_k^\Delta}$:

$$L_{T_k^\Delta} = \begin{pmatrix} x_k^{1,\Delta} & -x_k^{2,\Delta} & 0 \\ -x_k^{2,\Delta} & x_k^{1,\Delta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{21}$$

where $x_k^{1,\Delta}$ y $x_k^{2,\Delta}$ are the two first coordinates of T_k^Δ.

The new unknown vector, which correspond to the new parametrization of X_0^a, is:

$$\Phi = \begin{pmatrix} x_0^1 & x_0^2 & a & b & (M_{X_0}^1)^T, \cdots, (M_{X_0}^N)^T \end{pmatrix}^T \tag{22}$$

The expression (20) is decomposed in the following elements:

$$\begin{pmatrix} u_k^i \\ v_k^i \end{pmatrix} = y_k^i \qquad \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} = K(R_c (R_k^\Delta M_{X_0}^i + T_0 + L_{T_k^\Delta}) + T_c) \tag{23}$$

where U_k^i, V_k^i y S_k^i are linear in terms of Φ but not in terms of y_k^i.

$$U_k^i = L_{U_k}^i \Phi + b_{U_k}^i \qquad V_k^i = L_{V_k}^i \Phi + b_{V_k}^i \qquad S_k^i = L_{S_k}^i \Phi + b_{S_k}^i \tag{24}$$

The projective scale is removed from the transformation by using vector product properties:

$$\begin{pmatrix} u_k^i \\ v_k^i \\ 1 \end{pmatrix} = \lambda_k^i \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} \rightarrow \begin{pmatrix} u_k^i \\ v_k^i \\ 1 \end{pmatrix} \times \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{25}$$

where two lineally independent equations are obtained for Φ.

$$\begin{cases} v_k^i S_k^i - V_k^i = 0 \\ -u_k^i S_k^i + U_k^i = 0 \end{cases} \rightarrow \begin{cases} (v_k^i L_{S_k}^i - L_{V_k}^i)\Phi = v_k^i b_{S_k}^i - b_{V_k}^i \\ (-u_k^i L_{S_k}^i + L_{U_k}^i)\Phi = -u_k^i b_{S_k}^i + b_{U_k}^i \end{cases} \tag{26}$$

Using all measurements inside Y_L, a lineal system of equations is obtained in terms of Φ:

$$A\Phi = B \qquad A = \begin{pmatrix} (v_1^1 L_{S_1}^1 - L_{V_1}^1) \\ (-u_k^1 L_{S_1}^1 + L_{U_1}^1) \\ \vdots \\ (v_K^N L_{S_K}^N - L_{V_K}^N) \\ (-u_K^N L_{S_K}^N + L_{U_K}^N) \end{pmatrix} \qquad B = \begin{pmatrix} v_1^1 b_{S_1}^1 - b_{V_1}^1 \\ -u_1^1 b_{S_1}^1 + b_{U_1}^1 \\ \vdots \\ v_K^N b_{S_K}^N - b_{V_K}^N \\ -u_K^N b_{S_K}^N + b_{U_K}^N \end{pmatrix} \tag{27}$$

It is straightforward to show that system of (27) has a single-parameter family of solutions. If Φ_0 is a possible solution, then $\Phi_0 + \psi\Delta$ is a solution for any $\psi \in R$, with Δ:

$$\Delta = \begin{pmatrix} T_{c,x^1} & T_{c,x^2} & a & b & \begin{pmatrix} 0 \\ 0 \\ T_{c,x^3} \end{pmatrix}^T & \cdots & \begin{pmatrix} 0 \\ 0 \\ T_{c,x^3} \end{pmatrix}^T \end{pmatrix} \tag{28}$$

In fact, if Δ is normalized, it matches up with the eigen-vector associated to the zero eigenvalue of matrix $A^T A$.

Using the constraint that $a^2 + b^2 = 1$, and the singular value decomposition of matrix A, an exact solution of system (20) is obtained.

Once Φ is available, the solution X_0^a is obtained by inverting the parametrization:

$$\alpha_0 = \tan^{-1}(a, b) \qquad M^i = R_0^{-1} M_0^i \qquad X_0^a = \begin{pmatrix} x_0^1 & x_0^2 & \alpha_0 & (M^1)^T & \cdots & (M^N)^T \end{pmatrix}^T \tag{29}$$

This method, although exact, is prone to error due to odometry noise and does not benefit from the Maximum Likelihood metric. However it is valid as a method to give an initial value for X_0^a before using the iterative approach.

3.6. Degenerate Configurations

The kind of trajectory performed by the robot during initialization has direct influence in the solution of X_0^a. There are three kinds of movements that yields degenerate solutions:

- Straight motion: there is no information about the center of rotation of the robot and thus the pose has multiple solutions.

- Rotational motion around robot axis: the following one-parameter (*i.e.*, n) family of solutions gives identical measurements in the image plane:

$$M_n^i = nM^i + (n-1)\begin{pmatrix} 0 \\ 0 \\ \bar{T}_c(3) \end{pmatrix} \qquad T_{0n} = nT_0 + (n-1)\begin{pmatrix} \bar{T}_c(1) \\ \bar{T}_c(2) \\ 0 \end{pmatrix} \tag{30}$$

with $\bar{T}_c = R_c^T T_c$.

$$X_k^a(n) = \begin{pmatrix} T_{0n} & \alpha_0 & M_n^1 & \cdots & M_n^N \end{pmatrix} \tag{31}$$

- Circular trajectory: under a purely circular trajectory the following one-parameter family of initialization vectors gives identical measurements in the image plane:

$$M_n^i = nM^i + R_0^T \qquad T_{0n} = nT_0 + (n-1)R_c^T T_c \tag{32}$$

$$X_k^a(n) = \begin{pmatrix} T_{0n} & \alpha_0 & M_n^1 & \cdots & M_n^N \end{pmatrix} \tag{33}$$

In practical cases it has been proved to be effective enough to combine straight trajectories with circular motion to avoid degeneracies in the solution of X_0^a.

3.7. Obtaining the Gaussian Equivalent of the Solution

Once the minimum of (16) is reached we suppose that the resulting value of X_0^a is the mean of the distribution $\mathbf{X_0^a}$. This section describes how to also obtain the covariance matrix.

The covariance matrix Σ_0^a of the optimized parameters is easily obtained by using a local approximation of the term $Y_L - \hat{Y}_L$ in the vicinity of the minimum $\hat{X}_0^a$ using the following expression:

$$\Sigma_0^a = (J^T \Sigma_L^{-1} J)^{-1} \tag{34}$$

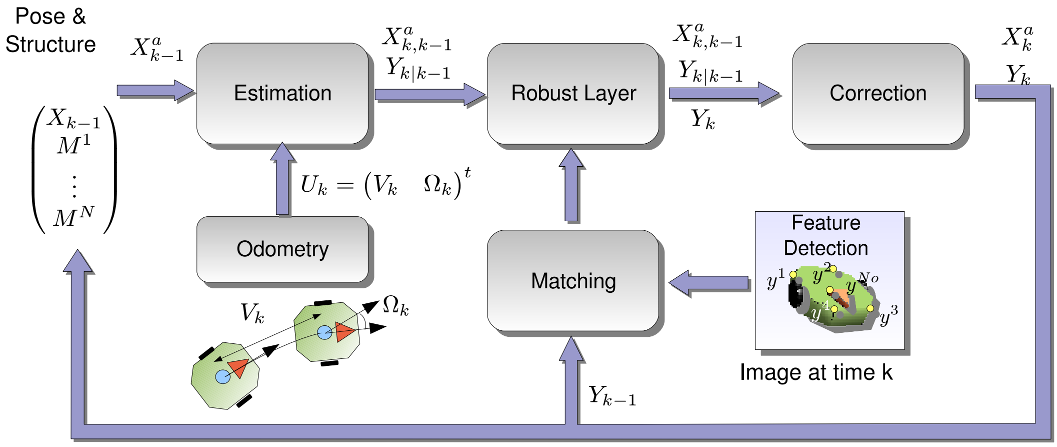
where J is the Jacobian matrix of $\hat{Y}$ with respect to parameters X_0^a. The Jacobian is available from the optimization method, in which is used to compute the iteration steps.

4. Online Robot Localization

In this section the solution to X_k^a given the last pose information is derived. The fact that last frame information is available and the assumption of soft motion between frames allows to greatly simplify the problem.

A special emphasis is given to the fact that any process handled by the system is considered a random entity, in fact a Gaussian distribution defined at each case by its mean vector and covariance matrix. The problem of obtaining pose and structure, encoded in X_k^a given image observations Y_k and the previous time estimation X_{k-1}^a is viewed from the point of view of statistical inference, which means searching for the posterior probability distribution $p(X_k^a|Y_1, \cdots, Y_k)$. That distribution gives the best estimation of X_k^a given all the past knowledge available. In Figure 6, a brief overview of the online method is presented.

Figure 6. Overview of the online algorithm.



The online approach is divided into three steps:

- **Estimation Step:** using the previous pose distribution $p(X^a_{k-1}|Y_1, \cdots, Y_{k-1})$, defined by its mean $\hat{X}^a_{k-1}$ and covariance matrix Σ^a_{k-1}, and the motion model g^a, a Gaussian distribution which infers the next state is given $p(X^a_k|Y_1, \cdots, Y_{k-1})$.
- **Robust Layer:** the correspondence between image measurements and the 3D model of the robot easily fails, so a number of wrongly correspondences or outliers pollute the measurement vector Y_k. Using a robust algorithm and the information contained in the state vector, the outliers are discarded before the next step.
- **Correction Step:** using an outlier-free measurement vector, we are confident to use all the information available to obtain the target posterior distribution $p(X^a_k|Y_1, \cdots, Y_k)$

In all three steps it is required to propagate statistic processes over non-linear functions (f and h). As was stated in the previous section we show how to face the problem using first order expansions as it offers more compactness and is more readable. As a consequence the "Estimation" and "Correction" steps are solved using the so called Extended Kalman Filter (EKF) equations, which have been already implemented in problems of similar complexity such as visual Simultaneous Localization and Mapping (SLAM) [5].

4.1. Estimation Step

The estimation step uses the motion models available to infer the next pose of the robot which implies an increment in uncertainty. Starting from the last pose distribution $p(X^a_{k-1}|Y_1, \cdots, Y_{k-1}) = N(\hat{X}^a_{k-1}, \Sigma^a_{k-1})$, the motion model g^a and the noise included in odometry values, the following update is obtained:

$$\begin{aligned} \hat{X}^a_{k|k-1} &= g^a(\hat{X}^a_k, U_k) \\ \Sigma^a_{k|k-1} &= J_x^T \Sigma^a_{k-1} J_x + J_U^T \Sigma_W J_U, \end{aligned} \qquad (35)$$

where $\hat{X}^a_{k|k-1}$ and $\Sigma^a_{k|k-1}$ are the mean and covariance matrix of distribution:

$$p(X^a_k|Y_1, \cdots, Y_{k-1})$$

The matrices J_X and J_U are the first derivatives of the function g^a with respect to X^a_{k-1} and U_k respectively. Usually J_X in odometry systems is the identity, therefore, at this step, the covariance matrix $\Sigma^a_{k|k-1}$ results to be bigger in terms of eigenvalues, which means uncertainty.

4.2. Correction Step

The correction step removes the added uncertainty in the estimation by using image measurements. It passes from the distribution $p(X^a_k|Y_1, \cdots, Y_{k-1})$ to the target distribution $p(X^a_k|Y_1, \cdots, Y_k)$, which includes the last measurement.

Using the estimation shown in (35), and knowing the correspondence between measurements with the camera and structure point of the state vector, the estimated measurement is given:

$$\begin{aligned} Y_{k|k-1} &= h^a(X^a_k) && (36) \\ \Sigma_{Y_{k|k-1}} &= J_h^T \Sigma^a_{k|k-1} J_h + \Sigma_V && (37) \\ \Sigma_{X^aY} &= \Sigma^a_{k|k-1} J_h && (38) \end{aligned}$$

where J_h is the Jacobian matrix of the function h^a with respect to X^a_k and Σ_V is block diagonal matrix with Σ_v on each block. Function h^a performs the projection in the image plane of the camera of all visible points that form up the measurement vector Y_k. The correction step itself is a linear correction of $X^a_{k|k-1}$ and $\Sigma^a_{k|k-1}$ by means of the Kalman gain K_G:

$$\begin{aligned} K_G &= \Sigma_{X^aY} \Sigma^{-1}_{Y_{k|k-1}} && (39) \\ X^a_k &= X^a_{k|k-1} + K_G(Y_k - Y_{k|k-1}) && (40) \\ \Sigma^a_k &= \Sigma^a_{k|k-1} - K_G \Sigma^T_{X^aY} && (41) \end{aligned}$$

As it is stated in (41) the resulting Σ^a_k is reduced compared to $\Sigma^a_{k|k-1}$ which means that after the correction step, the uncertainty is "smaller".

4.3. Robust Layer

The robust layer has the objective of removing bad measurements from Y_k to avoid inconsistent updates of X^a_k in the correction step. In this paper we propose to include the Random Sample Consensus (RANSAC) algorithm [22] between the estimation and correction step of the filter. The general idea is to found among the measured data Y_k a set which agrees in the pose X_k, using the 3D model obtained in X^a_{k-1}.

The interest of applying RANSAC in a sequential update approach resides on several reasons: firstly it allows to efficiently discard outliers from Y_k preventing algorithm's degeneracy, which happens even if the motion model is accurate. Secondly, compared to online robust approaches, where a robust cost function is optimized, RANSAC allows not to break the Kalman filter approach, as it only cleans the measurement vector of outliers. Furthermore we have observed experimentally that the RANSAC algorithm can be very fast between iterations, as only a few outliers are inside the data. (We use the RANSAC implementation described in [17], which implements a dynamical computation of the outlier probability).

The RANSAC method proposed in the commented framework obtains the consensus pose X_k from the set of measurements Y_k and the 3D model available in X^a_{k-1} using the algorithm presented in [18]. For a robot moving in a plane, as it is the case with the mobile robot considered in this paper, it is enough to use a minimum of 2 correspondences between the model and the measurement which makes the RANSAC very fast for removing outliers.

4.4. Image Measurements

Contrary to the initialization case, in this step we have an accurate prediction of the tracked points in the image plane at each time instant, namely vector $Y_{k|k-1}$. Using such prediction we can easily match the 3D points in the state vector with measurements taken in a measurement set $\mathcal{F}_k$, using the SIFT detector applied to current image I_k. The matching is done in terms of distance in the image plane. Let $y^i_{k|k-1}$ a feature inside vector $Y_{k|k-1}$ and y^j_k a candidate obtained using SIFT method. We conclude that they are matched if $|y^j_k - y^i_{k|k-1}|^2_M < \tau_{max}$, where $||_M$ states for the Mahalanobis distance using the covariance $\Sigma_{y_{k|k-1}}$ computed from the matrix $\Sigma_{Y_{k|k-1}}$. The Mahalanobis distance allows to take into account the uncertainty predicted for $y^i_{k|k-1}$ and also helps to select a threshold τ_{max} with a probabilistic criterion.

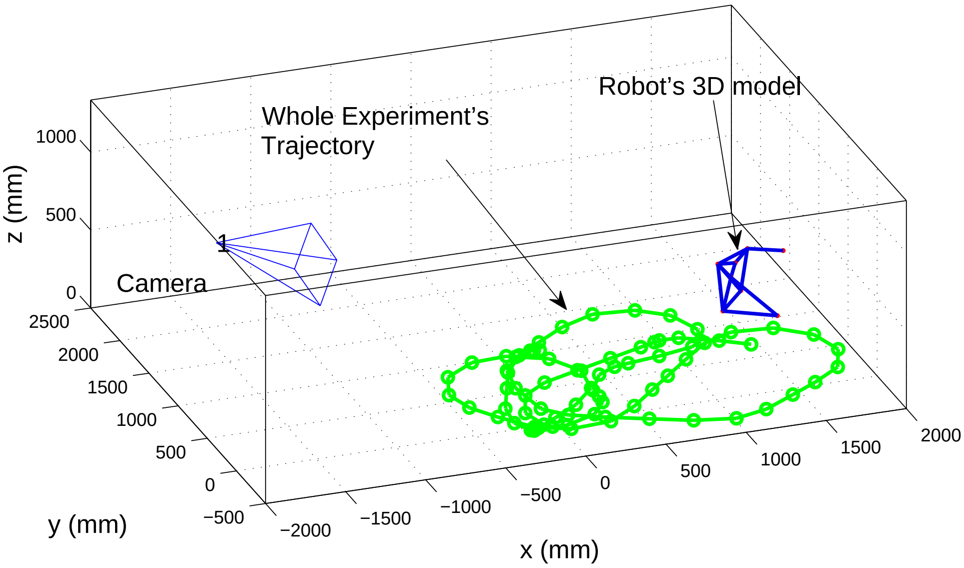
5. Results

This section describes the experimental setup developed to support the theoretical algorithms proposed in this paper. The experiments are divided in those performed over synthetic data and those run in a real implementation of the Intelligent Space in the University of Alcala (ISPACE-UAH) [23]. In both kind of experiments the same camera parameters are used, derived from the real device used in its real placement. The single camera consists of a CCD based sensor with resolution of 640 × 480 pixels and a physical size of 1/2 (around 8 mm diagonal). The optical system is chosen with a focal length of 6.5 mm which gives around 45^o of Field of View (FOV). The camera is connected to a processing and acquisition node through a IEEE1394 port, which support 15 fps in RGB image format acquisition. The intrinsic parameters are the following:

$$f_u = 636.7888 \quad f_v = 637.5610 \quad u_0 = 313.3236 \quad v_0 = 210.6894 \tag{42}$$

The camera is placed with respect a global coordinate origin, as it is displayed in Figure 7. Camera calibration is performed previously to the localization task, using checkerboards as calibration patterns and the "Matlab Calibration Toolbox" implemented by [24]. The distortion parameters of the camera are not considered in this case. As it can be shown in Figure 7, the camera is placed in oblique angle, which is specially useful for covering large areas with less FOV requirements (less distortion) compared to overhead configurations.

Figure 7. Geometric distribution of the camera and robot's trajectory.

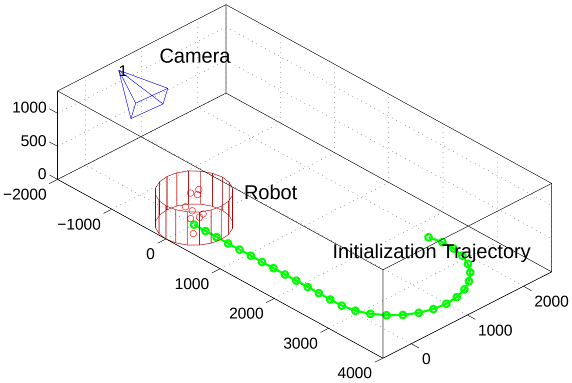


The robotic platform is connected to the same processing node controlling the camera by means of a wireless network. The camera acquisition and the odometry values obtained from the robot are synchronized. The control loop of the robot is internal, and it is prepared to follow position landmarks based on odometry feedback at faster sampling frequency than the image localization system (15 fps). Therefore, for each image acquisition cycle, the odometry system obtain several readings that are used by the internal loop control. In this paper the localization information provided by the cameras is not included in the control loop of the robot.

5.1. Simulated Data

In this experiment the robot structure is simulated with a set of $N = 10$ points distributed randomly inside a cylinder with radius $R = 0.5m$ and height $h = 1m$. The odometry system is supposed to have an uncertainty $\sigma^2_{Vl} = 10$ and $\sigma^2_{\Omega} = 1$ in linear and angular speed respectively. The initialization trajectory is shown in Figure 8. The image measurements are considered polluted with a Gaussian process of $\sigma^2_v = 10$, independently of each measurement and image coordinate.

Figure 8. Initialization trajectory of the robot in the experiment based on synthetic data.



To compare the performance of the initialization method proposed in Section 3., the following error magnitudes are described:

$$\epsilon_M = \frac{\sqrt{\sum_{i=1}^{N} ||M^i - M^i_{gth}||^2}}{\sqrt{\sum_{i=1}^{N} ||M^i_{gth}||^2}} \quad \epsilon_T = ||T_0 - T_{0,gth}|| \quad \epsilon_\alpha = ||\alpha_0 - \alpha_{0,gth}|| \tag{43}$$

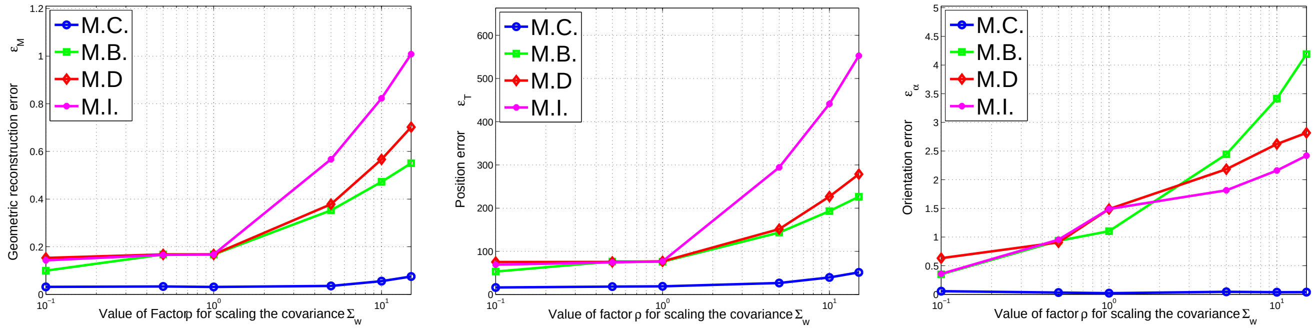
where $T_{0,gth}, \alpha_{0,gth}$ and M^i_{gth} correspond to the ground truth values of the initialization vector X^a_0. The following two experiments are proposed:

- *Initialization errors in function of the odometry error.* In this experiment the value of σ^2_{Vl} and σ^2_Ω are multiplied by the following multiplicative factor:

$$\rho \in \begin{pmatrix} 0.01 & 0.1 & 0.5 & 1 & 5 & 10 \end{pmatrix} \tag{44}$$

The different errors of Equation 43 can be viewed in Figure 9a–c in terms of ρ. As it can be observed in the results, the M.C. (Complete correlated matrix Σ_L) method outperforms the rest of approximations of Σ_L, specially the full diagonal method M.I., which means that the statistical modelling proposed in this paper is effective.

Figure 9. Experiment showing the different initialization errors in function of the amount of error in odometry readings.



(a) ϵ_M vs ρ (b) ϵ_T vs ρ (c) ϵ_α vs ρ

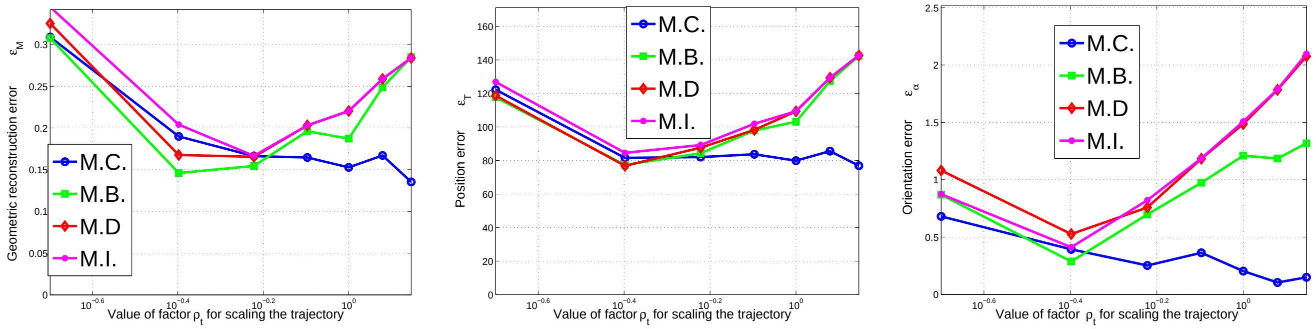
- *Initialization errors in function of the trajectory length.* The trajectory used in the experiment and displayed in Figure 8 is uniformly scaled by parameter ρ_t,

$$\rho_t \in \begin{pmatrix} 0.2 & 0.4 & 0.6 & 0.8 & 1 & 1.2 & 1.4 \end{pmatrix} \tag{45}$$

so that it can be guessed the relationship between trajectory length and initialization errors. In Figure 10 the initialization errors are displayed versus ρ_t.

In light of the results shown in Figure 10a–c, the M.C. method is capable of reducing the error no matter how large is the trajectory chosen. However, in the rest of the approximations of Σ_L there is an optimal point where the initialization errors are minimum. This results make sense, as without statistical modelling large trajectories contain accumulative errors which usually affects the final solution of X^a_0.

Figure 10. Experiment showing the different initialization errors in function of the trajectory length performed by the robot.



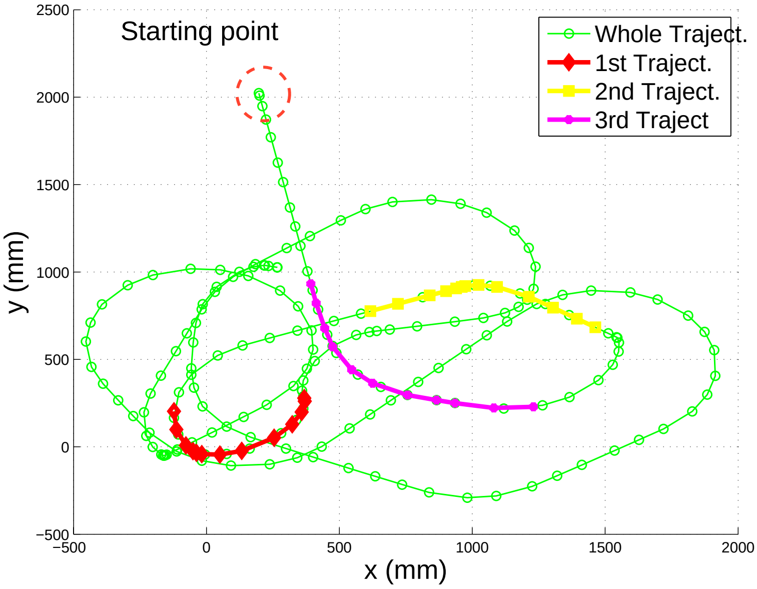
(a) ϵ_M vs ρ_t (b) ϵ_T vs ρ_t (c) ϵ_α vs ρ_t

Both experiments clearly manifest that the complete matrix Σ_L, with all its cross-correlated terms (M.C.), outperforms the rest of proposals, especially when it is compared to the case where Σ_L is the identity matrix (M.I.), which means no statistical modeling.

5.2. *Real Data*

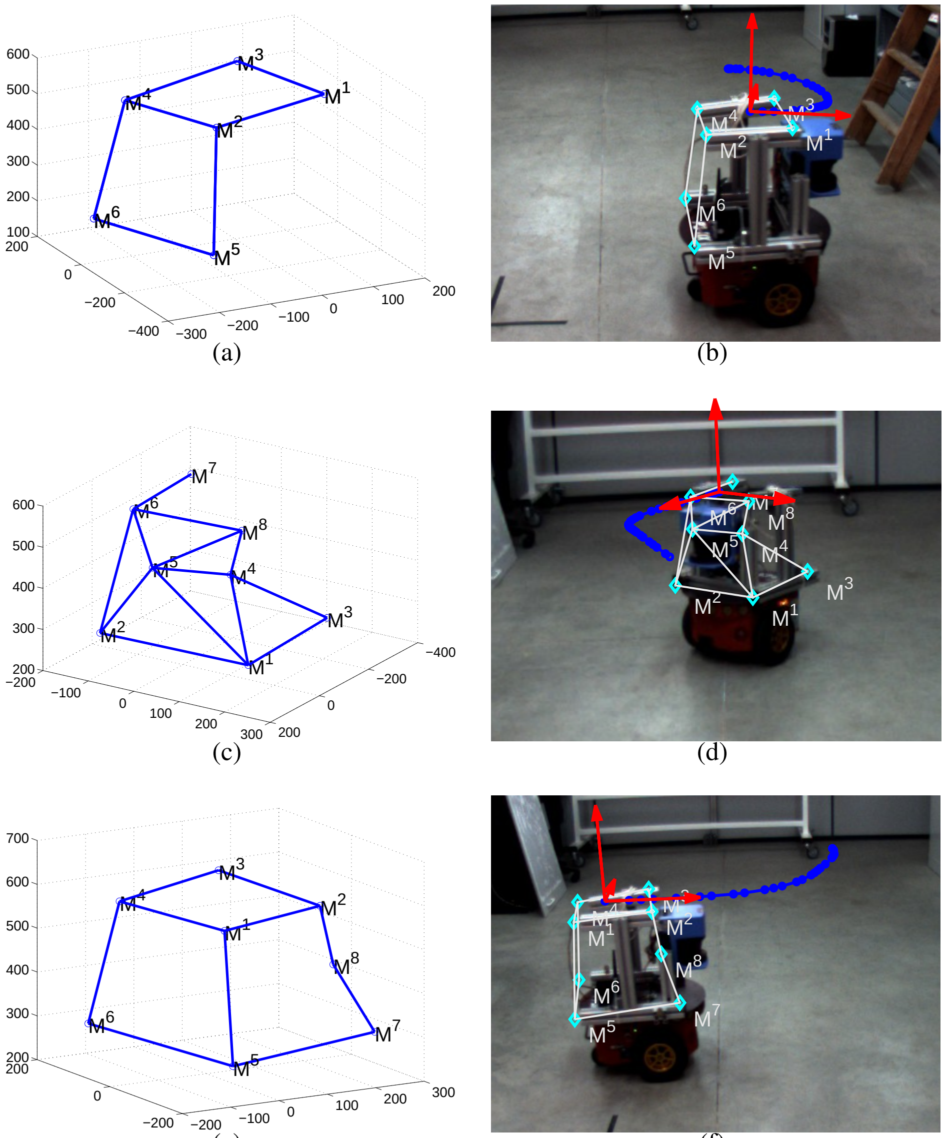
The initialization experiment proposed in this paper consists of a robot performing a short trajectory from which its 3D model and initial position is obtained using the results of Section 3.. We present a comparison of the initialization results using three different trajectories. Each one of the trajectories is displayed in Figure 11 as a colored interval of the whole trajectory performed by the robot in the experiment.

Figure 11. Intervals of robot's trajectory used for initialization.



The results of the initialization method on each of the 3 trajectories selected is shown in Figure 12. Depending on the trajectory used, the features viewed by the camera vary and thus the corresponding initialized 3D model. As it can be seen in Figure 12, the 3D model is accurate and its projection matches with points in the robot's structure in all cases. It must be remarked that on each case, the position obtained as a result of the initialization (*i.e.*, X_0) corresponds to the first position of each interval.

Figure 12. Single camera initialization results. On each row it is shown the resulting reconstruction and its projection in the image plane of the camera.



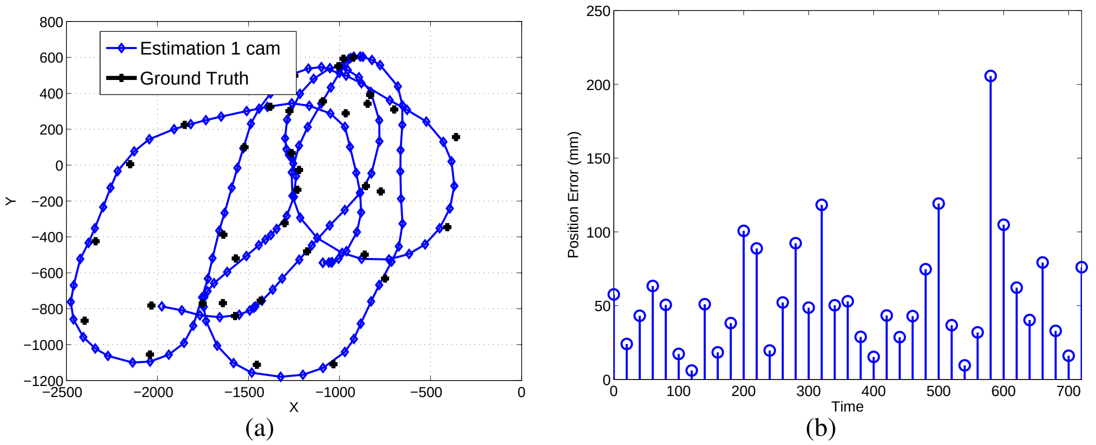
(a) (b)

(c) (d)

(e) (f)

The sequential algorithm is tested using the whole trajectory shown in Figure 11. The initial pose and 3D model are the result of the initialization results shown in Figure 12e,f. The estimated position of the robot, compared to a "ground truth" measurement, is presented in Figure 13a,b.

Figure 13. Online robot's pose compared with the "ground-truth".

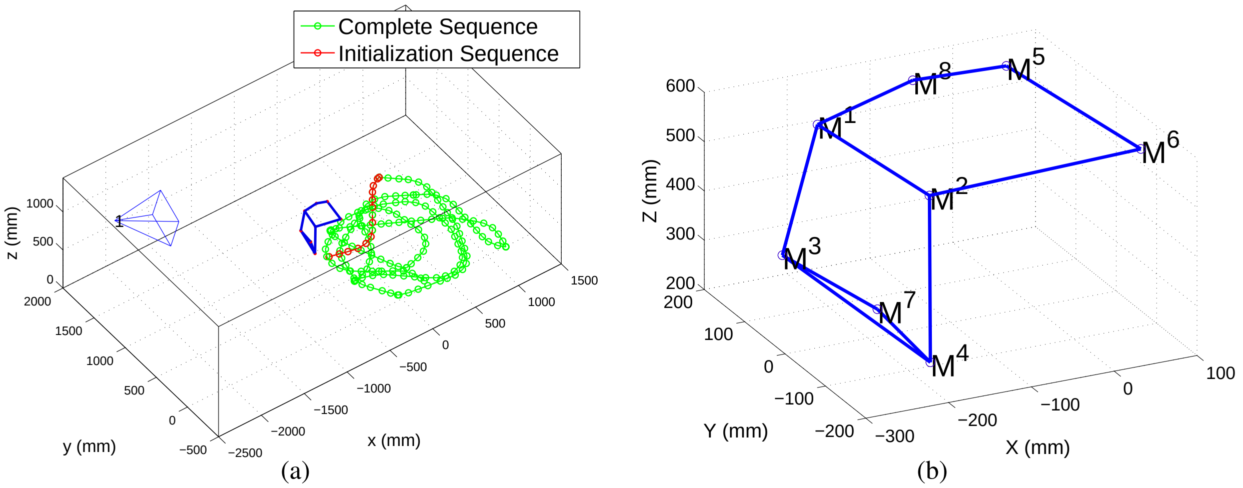


(a) (b)

The ground truth is obtained by means of a manually measured 3D model of the robot. The model is composed of 3D points, that are easily recognized by a human observer. By manually clicking points of the 3D model in the image plane, and by using the method proposed in [18], the reference pose of the robot is obtained in some of the frames of the experiment.

Another experiment is proposed to test the online algorithm with occlusions and a larger path followed by the robot. In Figure 14a it is shown the geometric placement of the camera and in Figure 14b it is shown the geometric model obtained during initialization.

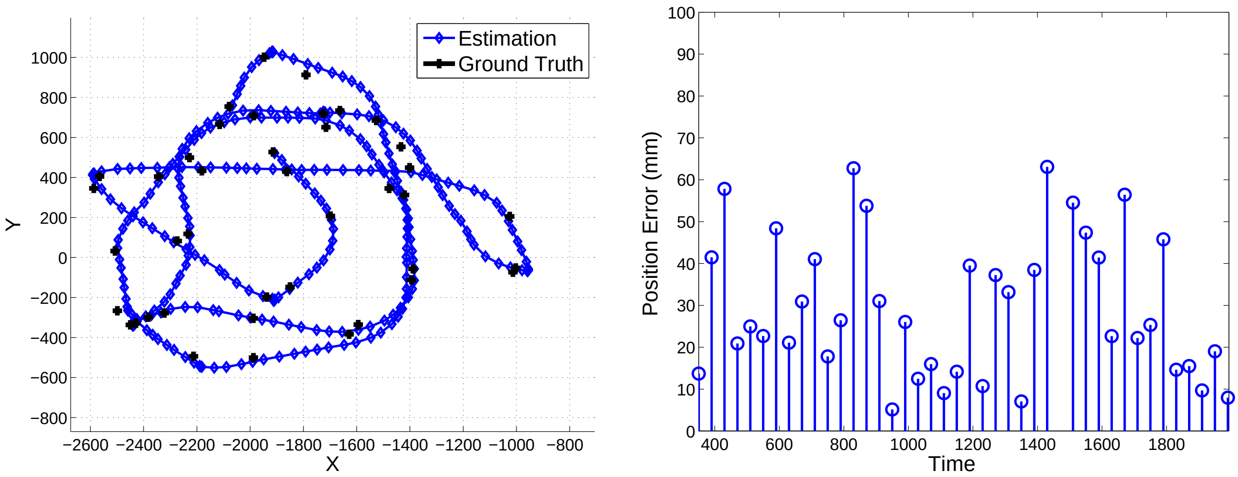
Figure 14. Scene geometry and 3D model obtained during initialization.



(a) (b)

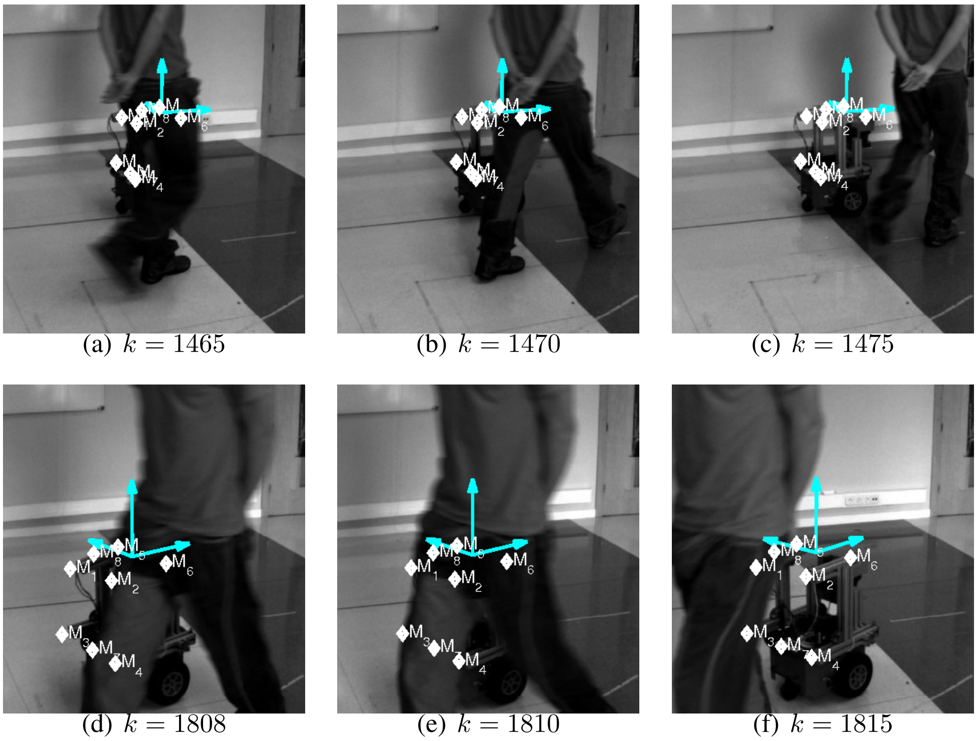
The resulting localization error is shown in Figure 15b. In Figure 16 it is shown several frames, indexed by the time sample number k, presenting hard occlusions between the camera and the robot without losing the tracking. The RANSAC method used in the Kalman loop avoid erroneous matches in the occluded parts of the object. Besides, in those frames with completely occluded features, only the estimation step of the Kalman filter is done, which is accurate enough for short periods of time.

Figure 15. Localization results in an experiment with occlusions.



(a) (b)

Figure 16. Occlusions and its influence in the localization accuracy.



(a) $k = 1465$ (b) $k = 1470$ (c) $k = 1475$

(d) $k = 1808$ (e) $k = 1810$ (f) $k = 1815$

In both sequential experiments the 3D model is composed of 8 to 10 points and no extra geometrical information is introduced in the state vector. As a future line of study a simultaneous reconstruction and localization approach can be adopted to introduce online extra 3D points in the state vector as the robot is tracked. A very similar approach is done in the Simultaneous Localization and Mapping (SLAM) problem, and some of their solutions [5] are perfectly applicable to the problem assessed in this paper.

6. Conclusions

This paper has presented a complete localization system for mobile robots inside intelligent environments with a single external camera. The objectives of obtaining the pose of the robot based on its natural appearance has been tackled in the paper using a reconstruction approach followed by a sequential localization approach.

The main contributions of this paper are summarized in the following list:

- The initialization step provides a non-supervised method for obtaining the initial pose and structure of the robot previously to its sequential localization. A Maximum Likelihood cost function is proposed, which obtains the pose and geometry given a trajectory performed by the robot. The proposal of this paper allows to compensate for the odometry drift in the solution. Also an exact initialization method has been proposed and the degenerate configurations have been identified theoretically.

- The online approach of the algorithm obtains the robot's pose by using a sequential Bayesian inference approach. A robust step, based on the RANSAC algorithm, is proposed to clean the measurement vector out of outliers.

- The results show that the proposed method is suitable to be used in real environments. The accuracy and non-drifting nature of the pose estimation have been also evaluated in a real environment.

The future research must be oriented to scale the problem for large configurations of non-overlapped cameras and multiple robots, where extra problems arise, such as to automatically detect the transitions between cameras and online refinement of the geometric models. It is important from our point of view to, in the future, combine the information given by the system proposed in this paper with information sensed onboard the robots using cameras. This approach allows to jointly build large maps attached to information given by the cameras, so that robots can be localized and controlled to perform a complex task.

Acknowledgements

This work was supported by the Ministry of Science and Technology under RESELAI project (references TIN2006-14896-C02-01) and also by the Spanish Ministry of Science and Innovation (MICINN) under the VISNU project (REF-TIN2009-08984).

References

1. Lee, J.; Hashimoto, H. Intelligent space concept and contents. *Advanced Robotics* **2002**, *16*, 265–280.
2. Aicon 3D. Available online: www.aicon.de (accessed January 2010).
3. ViconPeak. Available Online: www.vicon.com (accessed January 2010).
4. Se, S.; Lowe, D.; Little, J. Vision-based global localization and mapping for mobile robots. *IEEE Trans. Robotics Automat.* **2005**, *21*, 364–375.

5. Davison, A.J.; Murray, D.W. Simultaneous Localization and Map-Building Using Active Vision. *IEEE Trans. Patt. Anal.* **2002**, *24*, 865–880.
6. Hada, Y.; Hemeldan, E.; Takase, K.; Gakuhari, H. Trajectory tracking control of a nonholonomic mobile robot using iGPS and odometry. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Tokyo, Japan, 2003; pp. 51–57.
7. Fernandez, I.; Mazo, M.; Lazaro, J.; Pizarro, D.; Santiso, E.; Martin, P.; Losada, C. Guidance of a mobile robot using an array of static cameras located in the environment. *Auton. Robots* **2007**, *23*, 305–324.
8. Morioka, K.; Mao, X.; Hashimoto, H. Global color model based object matching in the multi-camera environment. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006; pp. 2644–2649.
9. Chung, J.; Kim, N.; Kim, J.; Park, C.M. POSTRACK: a low cost real-time motion tracking system for VR application. In *Proceedings of Seventh International Conference on Virtual Systems and Multimedia*, Berkeley, CA, USA, 2001; pp. 383–392.
10. Sogo, T.; Ishiguro, H.; Ishida, T. Acquisition of qualitative spatial representation by visual observation. In *Proceedings of International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.
11. Kruse, E.; Wahl, F.M. Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998; pp. 662–667.
12. Hoover, A.; Olsen, B.D. Sensor network perception for mobile robotics. In *Proceedings of IEEE International Conference on Robotics and Automation*, Krakow, Poland, 2000; pp. 342–347.
13. Steinhaus, P.; Ehrenmann, M.; Dillmann, R. MEPHISTO. A Modular and extensible path planning system using observation. *Lect. Notes Comput. Sci.* **1999**, *1542*, 361–375.
14. Shi, J.; Tomasi, C. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 1994; pp. 593–600.
15. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **2004**, *60*, 91–110.
16. Triggs, B.; McLauchlan, P.; Hartley, R.; Fitzgibbon, A. Bundle adjustment–a modern synthesis. *Lect. Notes Comput. Sci.* **1999**, *1883*, 298–372.
17. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
18. Moreno-Noguer, F.; Lepetit, V.; Fua, P. Accurate non-iterative o (n) solution to the PnP problem. In *Proceedings of IEEE International Conference on Computer Vision*, Rio de Janeiro, Brasil, 2007; pp. 1–8.
19. Schweighofer, G.; Pinz, A. Globally optimal o (n) solution to the pnp problem for general camera models. In *Proceedings of BMVC*, Leeds, UK, 2008.
20. Van der Merwe, R.; de Freitas, N.; Doucet, A.; Wan, E. *The Unscented Particle Filter*; Technical Report CUED/F-INFENG/TR380; Engineering Department of Cambridge University: Cambridge, UK, 2000.

21. Pizarro, D.; Mazo, M.; Santiso, E.; Hashimoto, H. Mobile robot geometry initialization from single camera. Field and Service Robotics; Springer Tracts in Advanced Robotics: Heidelberg, Germany, 2008; pp. 93–102.
22. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Match.* **1981**, *24*, 381–395.
23. GEINTRA. Group of Electronic Engineering Applied to Intelligent Spaces and Transport. Available online: http://www.geintra-uah.org/en (accessed December 2009).
24. Bouguet, J.Y. Matlab Calibration Toolbox. Available online: http://www.vision.caltech.edu/bouguetj/calib-doc (accessed December 2008).