Localization and Geometric Reconstruction of Mobile Robots Using a Camera Ring

Daniel Pizarro, *Student Member, IEEE*, Manuel Mazo, *Member, IEEE*, Enrique Santiso, Marta Marron, *Member, IEEE*, and Ignacio Fernandez

***Abstract*—In this paper, a system capable of obtaining the 3-D pose of a mobile robot using a ring of calibrated cameras attached to the environment is proposed. The system robustly tracks point fiducials in the image plane of the set of cameras generated by the robot's rigid shape in motion. Each fiducial is identified with a point belonging to a sparse 3-D geometrical model of the robot's structure. Such a model allows direct pose estimation from image measurements, and it can easily be enriched at each iteration with new points as the robot motion evolves. The process is divided into an initialization step, where the structure of the robot is obtained, and an online step, which is solved using sequential Bayesian inference. The approach allows the proper modeling of uncertainty in measurements and estimations, and at the same time, it serves as a regularization step in pose estimation. The proposed system is verified using simulated and real data.**

***Index Terms*—Computer vision, indoor localization, intelligent spaces, robotics, sensor networks.**

I. Introduction

LOCALIZATION of mobile robots in indoor environments using a sensor network still remains to be a hot topic. The short distances involved in the localization, jointly with the structural elements found inside buildings, avoid in most of the cases to adapt the same radio technology that successfully made it possible to partially solve outdoor localization. Instead, the special conditions of indoor localization require different approaches, as it fits better with short-range sensors such as vision, ultrasounds, or recently ultrawideband (UWB) technology.

We propose in this paper a method to retrieve the pose of a mobile robot using vision sensors that are attached to the indoor environment. The cameras form part of a sensor network known as "Intelligent Space" [1]–[3]. The idea behind is to cover a bounded area with sensors connected to a centralized system, which analyzes the information and makes decisions. A set of "agents," such as robots, display screens, or any other electronic device, are remotely controlled by the environment to accomplish a certain task. Knowing where such agents are, particularly mobile robots, with enough accuracy and robustly is quite important for almost any oriented application of "Intelligent Spaces" like human assistance, robot cleaning, surveillance, and more.

A. Previous Works

Despite the potential of using camera networks to localize robots, there are relatively few publications on this area compared with those in which the camera is uniquely inside the robot [4], [5]. Some examples of robot localization with camera networks can be found in the literature, where the robot is equipped with artificial landmarks, either active [6], [7] or passive [8], [9]. In other works, a model of the robot, either geometrical or of appearance [10], [11], is previously learnt to the tracking task. In [12] and [13], the position of static and dynamic objects is obtained by multiple camera fusion inside an occupancy grid. An appearance model is used afterward to ascertain which object is in 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 previously obtained to the tracking in a supervised task.

B. Localization Based on Natural Appearance

In this paper, we present a localization system that does not necessary rely on invasive beaconing or previous supervised learning tasks. The proposed approach only needs as prior information the rigidity assumption in the geometry of the object to track and the calibration parameters from the set of cameras.

Obtaining the pose of a mobile robot using calibrated cameras, in the absence of other information, requires to define a common coordinate origin attached to the robot's volume from which to refer the pose. As a consequence, in general terms, the pose cannot be recovered without also recovering the geometrical information that defines the robot's coordinate origin. In most of the cases, the robot's geometry is easily observed in the images as points, lines, or any other tractable entity whose 3-D equivalent is possible to be inferred from image projections. Therefore, in this paper, the robot's pose is jointly obtained with a set of 3-D points from the robot's structure.

The computer vision community has developed a set of widely accepted solutions for the problem of obtaining rigid structure from motion (SFM). There are many publications of both sequential [5], [14] and batch approaches [15]–[17], and it is considered a mainly solved problem. Most of these methods are focused on scene reconstruction using a moving camera, so the geometry completely surrounds it, instead of occupying a

Manuscript received January 30, 2008; revised June 27, 2008. First published May 26, 2009; current version published July 17, 2009. This work was supported by the Ministry of Science and Technology under RESELAI Project TIN2006-14896-C02-01. The Associate Editor coordinating the review process for this paper was Dr. Jesús Ureña.

The authors are with the Department of Electronics, Universidad de Alcala, Alcala de Henares 28871, Spain (e-mail: pizarro@depeca.uah.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2009.2016380

small amount on its field of view. The main efforts are spent at the moment on the creation of unsupervised methods for reconstruction, which are able to manage with high amount of information (thousand of points in hundreds of different views) or incomplete data sets.

Usually, online methods can be split up into two parts. First, an unsupervised initialization algorithm is used to set up geometry from motion using a metric reference. Using autocalibration techniques [18], the camera parameters are obtained in the case they are unknown. The second step, which is online, combines the previous time estimation to obtain object pose given the geometry [17]. The intention of this paper is to show how to adapt such approaches to compute the pose and structure of the robot.

In [19], a system with similar objectives to this paper, which performs robot localization, but using a single camera, is proposed. In such a proposal, the initialization is solved by using a "bundle adjustment" approach, which needs the odometry information from the robot to serve as metric information. The online solution incorporates a robust method that allows the system to work under occlusions and false matchings. This paper extends the proposal made in [19] to work with several cameras, exploring the especial assumptions necessary in such a case. The statistical approach is maintained in this paper as the basis to achieve the online pose and structure of the robot.

This paper is organized as follows. In Section II, the objectives and a general schema of the proposal is presented. The problem of measuring information with the camera is explained in Section III. The initialization of pose and geometry of the robot from several cameras is presented in Section IV. In Section V, the online algorithm, which obtains the robot's pose given image measurements, is explained.

Finally, in Sections VI and VII, both the experimental results and the conclusions of this paper are discussed.

II. Objectives and Proposed Approach

The objective of this paper is to obtain the pose of a mobile robot, which is seen by a set of calibrated cameras fixed to the environment. The pose of the robot and the extrinsic parameters of the cameras are referred to a global coordinate origin O_W whose position is physically known. Camera calibration is performed previously and independently of the localization system using a classical approach that is based on calibration patterns and Tsai's method. Therefore, the intrinsic and extrinsic parameters of the set of cameras are supposed to be known at any stage of the algorithm.

This paper proposes to localize a robot by using as information a geometric model of the robot's structure and its image identification in the set of cameras. The geometric model is built using an offline initialization step, which obtains a set of 3-D points referred from a common coordinate origin, fixed to the robot's real volume, and whose position and orientation with respect to O_W define the pose. Once the 3-D model is known, an online algorithm, which obtains the pose, is proposed afterward using a probabilistic approach.

The proposal made in this paper is designed to tackle with wide-baseline arrangements of cameras, which includes situations where different cameras watch complementary views of the robot, and the size of the robot is small compared to the distance to the cameras. The system is, thus, prepared for situations in which correspondences between cameras are not necessary at any specific stage of the algorithm. Our proposal consists of a series of different blocks specialized in retrieving and filtering the information available from the cameras to obtain the pose of the robot. The processes include those which compute the pose of the robot online and those contributing to set up the information required by the online algorithm (initialization processes).

1) Initialization of Pose and Geometry: Initially, neither the pose nor the structure of the robot are known, so a method to ascertain both is proposed. A batch reconstruction algorithm is used, which takes image measurements of the robot's geometry while a motion trajectory is performed by the robot. A bundle adjustment approach obtains the 3-D model of the robot and the set of poses it occupies in the trajectory.
2) Online Pose Computation: The online computation assumes that the initialization step was successful, so the 3-D model and the last pose of the robot are both available. A probabilistic approach is used to solve the sequential update of the robot's pose given the last time estimation and the image measurements.
3) Measurement Process: The measurements, which were taken from the set of cameras, provide the positions of the known structure points of the robot in the image plane. A method of natural marker tracking is proposed, which combines an interest point detector with a point tracker.

A. *Definitions and Notation Used*

The robot's pose at time k is described by a vector X_k. We suppose that the robot's motion lies on the plane $z = 0$ referred from O_W, so the vector pose X_k is described by three components (x_k, y_k, and α_k). 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 pose X_{k-1} and the input U_k given by odometry (i.e., angular speed and linear speed of the robot), if available. The kind of motion model used is tightly coupled with the robot's hardware and odometry system. In our proposal, it is not essential to exactly know how the robot moves, but if available, it can be useful to increase the convergence of the algorithms.

The robot's geometry is composed by a sparse set of N 3-D points $\mathcal{M} = \{M^1, \ldots, M^N\}$ referred from a local coordinate origin described by the robot's pose X_k. The points M^i are static in time due to the robot's rigidness, and, thus, no temporal subindex is required for them. The 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 Fig. 1), e.g.,

$$\begin{aligned} M^i_{X_k} &= t(X_k, M^i) \\ &= \begin{pmatrix} \cos(\alpha_k) & \sin(\alpha_k) & 0 \\ -\sin(\alpha_k) & cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{pmatrix} M^i + \begin{pmatrix} x_k \\ y_k \\ 0 \end{pmatrix}. \end{aligned} \quad (1)$$

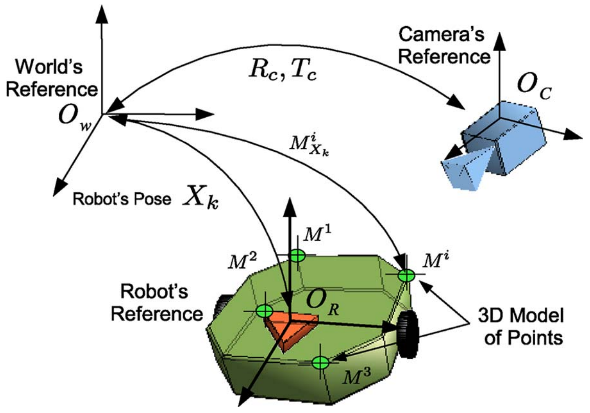


Fig. 1. Spatial relationship between the world's coordinate origin O_W, the robot's coordinate origin O_R, and the camera's coordinate origin O_C.

The augmented vector X_k^a, 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 $\mathcal{M}$, i.e.,

$$X_k^a = (X_k, M^1, \ldots, M^N). \tag{2}$$

There are N_c different cameras, modeled each by an ideal "pin-hole" model described by its 3 × 4 projection matrix P, which encodes intrinsic and extrinsic parameters. The camera projection model is expressed by the nonhomogeneous transformation $y_k^i = h(M^i_{X_k}, P)$, which converts a 3-D point $M^i_{X_k}$ expressed in a global coordinate origin O_W into its 2-D projection $y_k^i = (u_k^i, v_k^i)^T$ in the image plane using camera parameters P.

In this paper, we consider that the set of cameras are almost perfect synchronized in their acquisition. In real implementation, the cameras are synchronized by hardware using a trigger signal.

III. Measurement of Natural Markers

On most of the natural objects, we can find points whose image projection is able to independently be tracked in the image plane of the position the object occupies and based on the 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 easily be related with their 3-D counterparts. The set of methods that are focused on tracking natural markers have become a very successful and deeply studied topic in the literature [20], [21], as they represent the basic measurements of most of the existing reconstruction methods.

The process of tracking is roughly divided into two main steps: the detection of image candidates of being natural markers, and the process of their identification under different viewpoints.

A. Detection of Natural Markers

The development of stable interest point detectors has successfully been achieved since the first corner detectors where applied. The works proposed in [22] and later improved by the widely known "Harris" detector [23] have extensively been used in many vision geometry tasks. The "Harris" detector has proved to be stable enough under a variety of projective transformations, allowing its use as a reliable natural marker detector. The main drawback of the "Harris" corner comes from its sensitiveness to image scale, i.e., failing to detect a corner in objects at very different distances from the camera. In [24], a scale-invariant detector was proposed based on finding extrema in a scale space of the original image, which yields a very robust and reliable marker detector that will be used in this paper.

Given an intensity image $I(u, v)$, the multiscale detector gives a number of N_o points encoded in the set $C = (y^1, \ldots, y^{N_o})$, which are candidates of being points belonging to the robot's structure.

B. Matching of Natural Markers

The process of matching consists of describing each fiducial included in the set C such that it can be identified in the subsequent images taken at the object's different poses. In the literature, there is a vast knowledge on how to efficiently track fiducials [24]–[26] by using appearance information retrieved around the point detected (i.e., texture patch).

Among them, the most used nowadays was proposed in [24] under the acronym scale-invariant feature transform (SIFT). It has extensively been used in many tasks, such as SFM or object recognition. The SIFT method associates to each point detected a descriptor that is invariant to scale, rotation, and partially to general affine distortion. The descriptor proved to be very distinctive to match it across a large database of features.

In general, the SIFT method introduces false matchings when a set of descriptors are used to identify their new position in the input images. Those false matchings are named "outliers," and their number or identity is not available when only using the image appearance (see Fig. 2).

In the rest of this paper, the way the SIFT method is applied is left in the background with the intention of not blurring the concept. However, the measurements and their identification with the robot's structure are based on descriptor matching, and, thus, they are susceptible to contain outliers.

IV. Initialization of Pose and Geometry

The aim of this section is to describe the required set up to localize a robot from which there is no previous information about its geometry or pose. The initialization step consists of obtaining the initial pose it occupies and a 3-D model of its structure using a set of $N_c > 2$ cameras. The importance of this step is crucial, as the 3-D model obtained serves as the necessary link between the robot's pose and the image measurements. The initialization afterward allows using the online approach presented in Section V.

The initialization step presented here is based on a structure-from-motion approach, where the robot's motion is jointly used with image measurements and the calibration of the cameras to initialize both pose and geometry.

The basic idea is that each camera gives a series of tracks corresponding to the fraction of the geometry it sees, which is taken from a short sequence of the robot in motion. Each different track corresponds to the projection in the image frame of a unique point from the robot's structure that is observed under different poses.

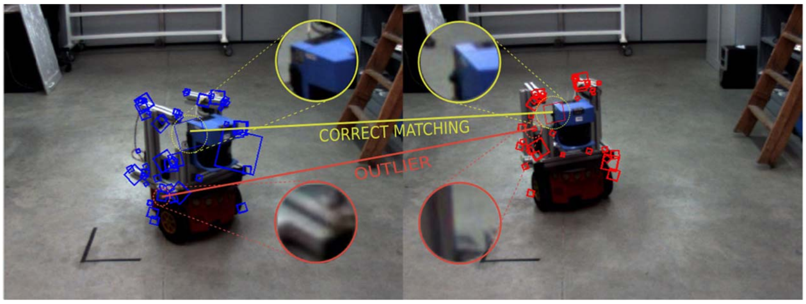


Fig. 2. Matching of natural markers, which are represented as fine lines, between two positions of the robot viewed in the same camera.

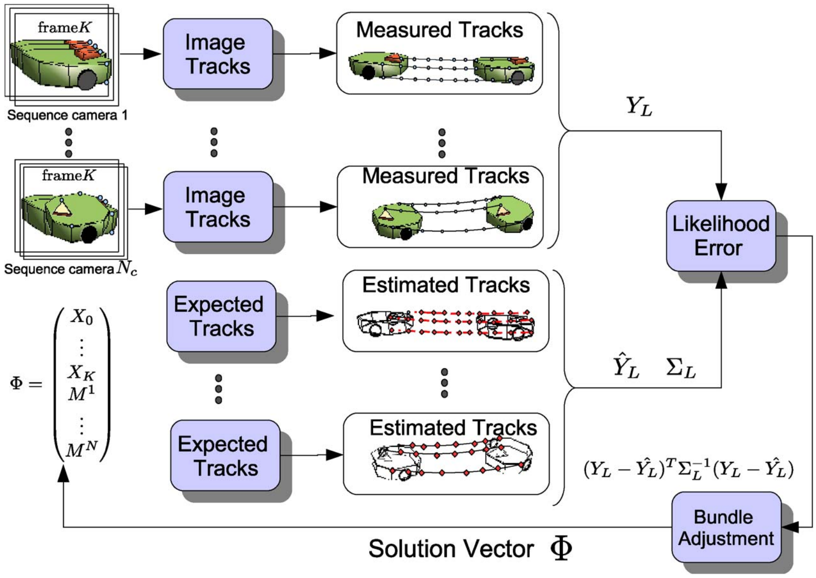


Fig. 3. Overview of the initialization process.

The set of tracks from all the cameras are combined in this step to obtain the common solution to the robot's pose and the reconstruction of the points they represent. In this approach, the matching of points through several cameras is not necessary, and it is replaced by tracks on each camera due to the robot's motion, which is generally a better posed problem. The cost of removing correspondences between images is that a structure-from-motion approach is necessary to obtain both pose and robot's geometry.

To find the solution to the structure-from-motion problem, we propose to use a "Bundle Adjustment" technique, which is able to optimally and efficiently obtain each camera viewed structure and the pose of the robot using image measurements. In Fig. 3, an overview of the initialization process is presented.

A. Preliminary Definitions

The initialization sequence consists of a set of K time samples starting from $k = 0$, where the robot is in motion. The pose vectors $X_0, \ldots, X_K$ represent the position and orientation of the robot at each time sample of the trajectory. Whenever the odometry system is available in the robot, a noisy measurement of each U_i, $i = 1, \ldots, K$, gives an estimation of the pose using the recursive motion model g and the starting pose X_0.

The robot's whole trajectory is observed by a set of $N_c > 2$ cameras. On each camera, which is identified by index n, a set of N_n points are tracked using the SIFT method commented in Section III. The number of detected measurements is different with the camera, and so, N_n is a function of the camera index.

At frame k and with the camera n, the measurement vector consists of the following:

$$Y_k^n = \left(\left(y_k^1\right)^T \quad \cdots \quad \left(y_k^{N_n}\right)^T \right)^T, \qquad y_k^i = \left(u_k^i, v_k^i\right)^T \quad (3)$$

where, for clarity, the upper index of the vector Y_k^n refers to the camera index, and that in y_k^i for the number of detected features removes the reference to the camera n that is left implicit. The set of measurements $y_0^i, \ldots, y_K^i$ describes a single track (i.e., the position of the same feature at different time instants), and the set $Y_0^n, \ldots, Y_K^n$ represents the set of tracks observed by the camera n.

Each single measurement y_k^i at time k is a function of the parameters of the camera it belongs to, the pose vector X_k, and a point M^j of the object's structure, i.e.,

$$y_k^i = h\left(t(X_k, M^j), P_n\right) + \mathbf{v_k^i}, \qquad \mathbf{v_k^i} = N(0, \Sigma_v) \quad (4)$$

where the correspondence of M^j with the ith point seen by the camera n is given by the following default ordering in the augmented vector:

$$X_k^a = \left(X_k \quad \underbrace{M^1 \quad \cdots \quad M^{N_1}}_{\text{Camera 1}} \quad \cdots \quad \underbrace{M^r \quad \cdots \quad M^{r+N_{N_c}}}_{\text{Camera } N_c} \right) \quad (5)$$

where $r = \sum_{n=1}^{N_c-1} N_n$. The augmented vector X_k^a contains all the geometry points from the robot's structure that is viewed by all the cameras from a common reference frame (robot's coordinate origin). For the moment, it is supposed that each camera observes different points from the robot's geometry, and so the direct concatenation shown in (5) does not produce multiple point repetitions.

The global measurement vector Y_L is the concatenation in a single column vector of all the measurements from the N_c cameras in the whole initialization sequence, i.e.,

$$Y_L = \Bigg(\underbrace{\left(Y_1^1 \quad \cdots \quad Y_K^1 \right)}_{\text{Camera 1}} \quad \cdots \quad \underbrace{\left(Y_1^{N_c} \quad \cdots \quad Y_K^{N_c} \right)}_{\text{Camera } N_c} \Bigg)^T. \tag{6}$$

B. Building the Cost Function

The complete set of unknowns is composed of the set of poses in the trajectory $X_0, \ldots, X_K$ and the 3-D coordinates of the points on the robot's structure M^j, $j = 1, \ldots, \sum_{n=1}^{N_c} N_n$. All the parameters are packed together into the objective vector Φ, i.e.,

$$\Phi = (X_0, \cdot, X_K, M^1, \ldots, M^N), \qquad N = \sum_{n=1}^{N_c} N_n. \tag{7}$$

A cost function is built in function of Φ to compare the error between the real measurement of vector Y_L with its estimation, which is named here as $\hat{Y}_L$, and obtained using the projection models [see (4)] as

$$\epsilon^2 = (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \tag{8}$$

where Σ_L represents the covariance matrix of the vector Y_L. The matrix Σ_L models the statistical properties of the joint vector Y_L. In this paper, we suppose that it is a diagonal block matrix, where each block represents the noise of a single measurement Σ_v so that (8) can be rewritten into the following addition of terms:

$$\epsilon^2 = \sum_{k=1}^{K} \sum_{n=1}^{N_c} \sum_{i=1}^{N_n}, \left(y_k^i - \hat{y}_k^i \right) \Sigma_v^{-1} \left(y_k^i - \hat{y}_k^i \right)^T. \tag{9}$$

The minimum of (9) with respect to Φ gives the value required to reconstruct the entire initialization trajectory. To reach the minimum, the iterative Levenberg–Mardquardt optimization method is used. The analytical expressions of the first and second derivatives of (8) with respect to the unknowns are required to compute a single step of the optimization.

In general, the matrix Σ_L is not block diagonal, and some correlation terms must be included (e.g., correlated errors in measurements on the same camera due to lens distortion or calibration error). However, the block diagonal version proposed in this paper, unless it is not accurate, has two main advantages. First, it easily allows discarding outliers using a robustified cost function, as each residual is separated in (9). Second, the Hessian of (9) has a sparse structure, which allows a quick computation of its inverse, which is required to obtain the minimum.

In [19], a similar approach is made by using a single camera; however, in that case, Σ_L models the growing error behavior of the odometry estimation, as it is included as a metric reference in the algorithm.

After the optimization of (9), the vector Φ can contain points from the robot's structure that were simultaneously seen by several cameras. These points can easily be fused together using an Euclidean distance criterion after optimization, as they are in the same coordinate origin (robot's coordinate origin).

1) Outlier Rejection: As the tracking method used in the initialization is based on image appearance matching, the presence of erroneous measurements is very probable, which are called "outliers" inside Y_L.

The "outliers" come from many different kinds of situations. Supposing that at $k = 0$ only points from the robot's structure are chosen, the SIFT method could fail into recognizing several points in the track by matching a different point from the background, other objects in motion, or another similar point in the robot's structure. False matchings belonging to the background can easily be identified (i.e., removing those SIFT candidates previously labeled as background before a robot enters the room or either using multiple camera constraints); however, the other sources of error need to include a robust geometric algorithm for their identification.

The cost function [see (8)] is designed to model possible Gaussian deviations of the solution compared to the measurements, which are suitable to contain noise. The distribution of outliers generally does not fit into such modeling, and so their presence in the optimization yields a biased result. The solution comes from using a robust cost function, which is capable of modeling the outlier distribution by removing their influence in the solution.

The equivalent robust cost function results in

$$\epsilon^2 = \sum_{k=1}^{K} \sum_{n=1}^{N_c} \sum_{i=1}^{N_n} \rho \left(\left| \epsilon_k^i \right| \right), \qquad \left(\epsilon_k^i \right)^2 = \left(y_k^i - \hat{y}_k^i \right) \Sigma_v^{-1} \left(y_k^i - \hat{y}_k^i \right)^T \tag{10}$$

where $\rho(s)$ can be any increasing function with $\rho(0) = 0$ and $(d/ds)\rho(0) = 1$ (see [27] for more details). The function ρ is also known as M-estimator in the literature.

There is a vast amount of publications dealing with the different alternatives to ρ and their influence with some kind of outlier distributions (see [27] and [28] for a survey). In this paper, we search for twice differentiable M-estimators so that the Jacobian and Hessian can be obtained for its use by a Gauss–Newton optimization algorithm. Such condition discards some effective estimators in tracking applications such as the least median square (LMedS) cost function, which is nondifferentiable.

Among the twice differentiable cost functions, it is desirable to choose one that preserves the convexity of the cost function as much as possible, such as the "Huber" cost function. However, in experimental tests, we have found that nonconvex cost functions such as "Cauchy" are easier to tune for a real tracking situation. The only condition is to bring the algorithm an initial solution that is close to the minimum so that the M-estimator does not substantially affect the convergence properties.

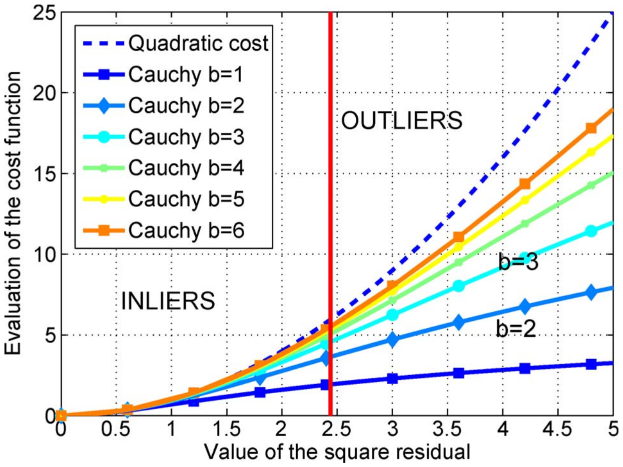


Fig. 4. Comparison of Cauchy distribution with different values of parameter *b*.

Therefore, the following ρ_i is proposed in this paper, which models the outliers as a "Cauchy"-based distribution:

$$\rho(s) = b^2 \log\left(1 + \frac{s^2}{b^2}\right) \tag{11}$$

where b^2 is used as a control parameter, which determines for which range of s^2 the function is approximated by a quadratic function and which range is considered as outliers.

The choice of parameter b is done by assuming that each single measurement y_k^i, which is considered as inlier, is described as a Gaussian distribution with covariance Σ_v, and, thus, each residual $(\epsilon_k^i)^2$ is a χ^2 random variable with two degrees of freedom. The value of the residual $|\epsilon_k^i|$ that represents the 95% of the cumulative χ^2 distribution is 2.44, so residuals with a bigger value are suitable of being considered as outliers as their probability of belonging to y_k^i is less than 0.05. In Fig. 4, the evaluation of the Cauchy robust function is given for several values of parameter b and compared to a simple quadratic cost. A value of $b \in [2, 3]$ will behave like a quadratic cost function with values of $|\epsilon_k^i| < 2.44$ and will attenuate large outliers due to correspondence errors. The tails of the Cauchy cost function tend to mitigate the effect of the outliers compared to that corresponding to the quadratic residual, which is derived from a Gaussian distribution.

2) Initialization Before Optimization: The optimization method to minimize the cost function requires a guess value for the solution Φ from which to start iterating. It is very important to choose a value as close as possible to the real solution, so that the probability to reach the global minimum increases.

A very simple proposal is made in this paper to get an initial guess of the solution. It consists of the following steps.

1) For each time instant k and camera n, the center of mass of the points encoded in Y_k^n is obtained (μ_k^n).
2) The position of the robot (x_k, y_k), not including the orientation parameters α_k, is obtained by camera triangulation of the N_c center of mass calculated, supposing they represent the same point in the 3-D space.
3) Once the set of $K + 1$ robot's positions are available $x_0, y_0, \ldots, x_K, y_K$, the set of orientations $\alpha_0, \ldots, \alpha_K$ are set so that they follow the kind of motion we expect in the object. Generally, the objects present a nonholonomic motion, so we aligned the orientations following the curvature of the motion. In the case that the object's motion is entirely holonomic, a random value can be used instead.
4) The geometry of the robot $M^1, \ldots, M^N$ is randomly initialized around a volume bounded by a sphere of radius R, which can be obtained by calculating the minimum sphere that covers the image measurements of all the cameras at a specific frame. This step uses a very simple background extraction method so that only measurements of the robot are taken into account.

In the case that the odometry readings are available, only the initial pose X_0 and the geometry of the robot are guessed, as mentioned before. After their estimation, the motion model and the odometry readings $U_1, \ldots, U_K$ generate the rest of the poses $X_1, \ldots, X_K$.

C. *Obtaining the Gaussian Equivalent of the Solution*

Once the minimum of (8) is reached, it is desirable to obtain the covariance matrix Σ_K^a of the vector X_K^a to connect the initialization step with the online approach of the next section.

The covariance matrix Σ_Φ of the optimized parameters Φ is easily obtained by using a local approximation of the term $Y - \hat{Y}$ in the vicinity of the minimum. The resulting Σ_Φ results from the following close expression:

$$\Sigma_\Phi = \left(J^T \Sigma_L^{-1} J\right)^{-1} \tag{12}$$

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

By truncating both the solution Φ and its covariance matrix Σ_Φ, the augmented vector X_K^a and its covariance matrix Σ_K^a are obtained.

V. Online Algorithm

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

A special emphasis is given in this paper to the fact that any process handled by the system is considered a random entity; in fact, it is a Gaussian distribution defined at each case by its mean vector and covariance matrix. The problem of obtaining pose and structure, which are 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 $p(X_k^a | Y_1, \ldots, Y_k)$. That distribution gives the best estimation of X_k^a given all the past knowledge available. In Fig. 5, a brief overview of the online method is presented.

The online approach is divided into three steps.

1) Estimation Step: Using the previous pose X_{k-1}^a and the motion model, a Gaussian distribution that infers the next state is given $p(X_k | Y_1, \ldots, Y_k)$.
2) Robust Layer: The correspondence problem in this point easily fails, so for each camera, a number of unlabeled outliers pollute the measurement vector Y_k. Using a

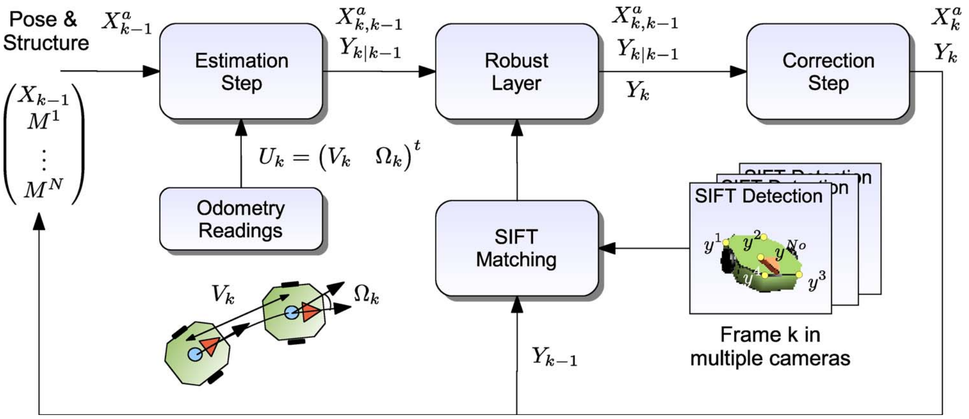


Fig. 5. Overview of the online algorithm.

robust algorithm and the information contained in the state vector, the outliers are discarded before the next step.

3) Correction Step: Using an outlier-free measurement vector, we are confident to use all the information available to obtain the target posterior distribution $p(X^a_k|Y_1,\ldots,Y_k)$.

In all three steps, we would manage the idea of propagating statistics over nonlinear functions (f and h). 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 by using extended Kalman filter (EKF) equations, which have already been implemented in problems of similar complexity such as visual simultaneous localization and mapping [5]. However, as stated in [19], there are other methods for Gaussian propagation (e.g., the unscented Kalman filter) with better statistical performance and that are less biased than the first-order expansion we show here.

A. Estimation Step

The estimation step uses the motion models available to infer the next pose of the robot. The state vector X^a_k evolves using the following augmented motion model:

$$X^a_k = g^a\left(X^a_{k-1}, U_k\right) = \begin{pmatrix} g(X_k, U_k) \\ M^1 \\ \vdots \\ M^N \end{pmatrix} \tag{13}$$

where the function $g(X_k, U_k)$ is the specific motion model used (i.e., differential model of a wheeled robot or simply a random walk model), and U_k's are the odometry readings.

The transit of (13) is often a process of uncertainty addition, as the motion information U_k is not accurately given or only linear motion models are available. It includes an update of the mean and covariance of the last pose as follows:

$$X^a_{k|k-1} = g^a\left(X^a_k, U_k\right) \tag{14}$$

$$\Sigma^a_{k|k-1} = J_X^T \Sigma^a_{k-1} J_X + J_U^T \Sigma_W J_U \tag{15}$$

where 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, so at this step the covariance matrix $\Sigma^a_{k|k-1}$ results to be bigger in terms of eigenvalues, which means uncertainty.

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

B. Correction Step

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

It is mandatory to remark that the measurement vector in the online process Y_k does not share the same structure used in the initialization processes, where each camera observes a separated set of points. Instead, due to the robot's motion, any point is suitable to be seen by any camera, and a group of cameras can see the same point.

Using the estimation shown in (14), and knowing the correspondence between measurements with the camera and the structure point of the state vector, the estimated measurement is given as

$$Y_{k|k-1} = h^a\left(X^a_k\right) \tag{16}$$

$$\Sigma_{Y_{k|k-1}} = J_h^T \Sigma^a_{k|k-1} J_h + \Sigma_V \tag{17}$$

$$\Sigma_{X^aY} = \Sigma^a_{k|k-1} J_h \tag{18}$$

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. The function h^a performs the projection in the image plane of the camera of all the visible points that form up the measurement vector Y_k. Supposing a case where all points included in X^a_k are seen by all the cameras, the following expression holds for h^a:

$$Y_k = h^a\left(X^a_k\right) = \begin{pmatrix} \left.\begin{matrix} h\left(t(X_k, M^1), P^1\right) \\ \vdots \\ h\left(t(X_k, M^N), P^1\right) \end{matrix}\right\} & \text{Camera1} \\ \vdots & \\ \left.\begin{matrix} h\left(t(X_k, M^1), P^{N_c}\right) \\ \vdots \\ h\left(t(X_k, M^N), P^{N_c}\right) \end{matrix}\right\} & \text{Camera}N_c \end{pmatrix}. \tag{19}$$

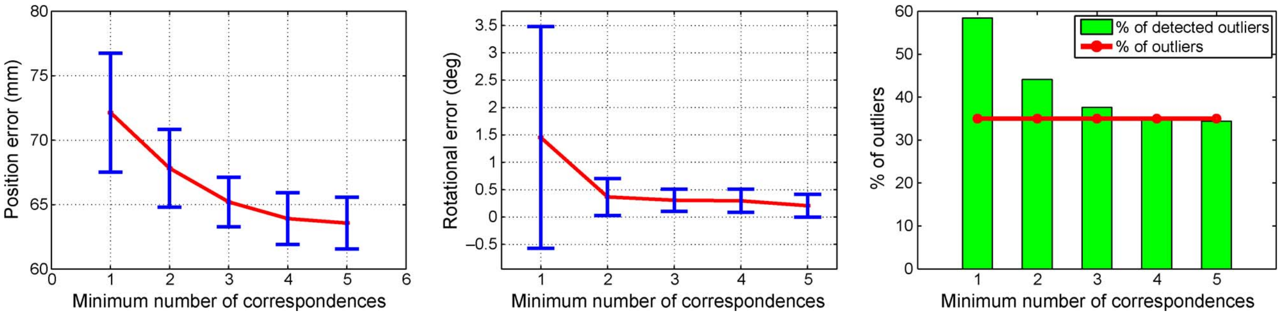


Fig. 6. Performance of the proposed algorithm in function of parameter s.

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, i.e.,

$$K_G = \Sigma_{X^aY}\Sigma^{-1}_{Y_{k|k-1}} \quad (20)$$

$$X^a_k = X^a_{k|k-1} + K_G(Y_k - Y_{k|k-1}) \quad (21)$$

$$\Sigma^a_k = \Sigma^a_{k|k-1} - K_G\Sigma^T_{X^aY}. \quad (22)$$

As stated in (22), 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."

C. 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. We propose and extend the same idea proposed in [19], in which a random sample consensus (RANSAC) algorithm [29] is used between the estimation and correction steps. The general idea is to find among the measured data Y_k a set that agrees in the pose X_k it will give at the correction step.

The interest of applying RANSAC to a purely online approach resides on several reasons. First, it efficiently allows discarding outliers from Y_k, preventing the algorithm's degeneracy, which happens even if the motion model is accurate. Second, 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 experimentally observed 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 [28], 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 3-D model available in X^a_{k-1}. It needs to find a way to solve pose from measurements and the minimum number of them required.

In the literature, there is an extensive amount of publications focused on solving the problem of obtaining the pose of a 3-D model given image correspondences, which is usually known as Perspective n Point problem (PnP). In such methods [30], [31], which are exact and do not need the previous pose X_{k-1}, the minimum number of required measurements is 3, even in the generalized case of using multiple cameras. Such proposals often require to solve complex methods, which include polynomial root finding or iterative algorithms.

Instead of using the exact approaches commented before, in this paper, we propose to use Kalman equations to solve the PnP problem. For each subset of Y_k used in RANSAC, the correction step of the Kalman filter is used to obtain the most voted X_k (and, consequently, X^a_k).

The following definitions are used in the proposed implementation of RANSAC.

1) We denote as Y^s_k[1] the group of s measurements randomly taken from Y_k.
2) It is defined as $X^a_k = F(X^a_{k|k-1}, Y^s_k)$, the direct function of which computes the pose inside X^a_k using a minimum number s of measurements. This function is implemented using the correction step expressions shown in Section V-B that is reduced to use the measures grouped inside Y^s_k.
3) The function $d = D(X^a_k, y^i_k)$ gives the Mahalanobis distance d between a single measurement y^i_k taken from Y_k and its estimation $\hat{y}^i_k$ using X^a_k, i.e.,

$$d = \left(y^i_k - \hat{y}^i_k\right)\Sigma^{-1}_{y^i_k}\left(y^i_k - \hat{y}^i_k\right)^T \quad (23)$$

where $\Sigma_{y^i_k}$ is obtained by propagating the statistical properties of X^a_k through the projection model. Supposing that y^i_k belongs to camera n_i, and it is the projection of point M^{j_i}, we have

$$\Sigma_{y^i_k} = \left(J^i_h\right)^T\Sigma^a_k J^i_h + \Sigma_v, \qquad J^i_h = \frac{\partial\left(h\left(t(X_k, M^{j_i}), P^{n_i}\right)\right)}{\partial X^a_k}. \quad (24)$$

The RANSAC method obtains the solution to X^a_k that best agrees among the data Y_k, knowing the last pose information X^a_{k-1}. The measurement vector Y_k is cleaned from "outliers" after RANSAC, refining the pose using the correction step on the complete set of "inliers."

The election of parameter s or the minimum number of measurements required to obtain X^a_k is in principle arbitrary in this case, as there is no explicit restriction on the number of measurements used in the Kalman correction equations.

In Fig. 6, a comparison of the performance of the proposed method in function of s in a realistic simulation is presented. It shows that for a value of $s = 4$, the percentage of discarded outliers is nearly that artificially included in the data and consequently achieving a small pose error.

[1] The upper index s must not be misled with the notation we used in Section IV, in which the upper index was used to indicate the camera.

In Fig. 6, it is shown how the present approach can even discard outliers for $s = 2$ or even $s = 1$, although in this case the standard deviation of the error increases, and RANSAC discards many useful inliers. The proposed method addresses a simpler problem than the common PnP. It has more information in general as it assumes that the estimated pose and the corrected pose are near enough so that a simpler linear correction models its differences, and as a consequence, the problem has less free dimensions than the PnP problem.

The computational complexity of this proposal comes from solving the correction step of the Kalman filter with reduced sets of measurements. The correction step can be solved with a complexity ruled by $\mathcal{O}(Ns^2)$, so it linearly scales with the size of the model and is quadratic with s. Compared to the proposals made in the literature, which usually require to solve polynomial roots for $s = 3$ or symbolic methods for $s = 4$, the method proposed in this paper only requires regular algebra of matrices. Recently, in [32], an algebraic $\mathcal{O}(s)$ solution has been proposed for the PnP problem for large s. However, for a small s, it has a complexity of $\mathcal{O}(s^3)$ and requires several singular value decompositions (SVDs) with zero eigenvalue identification. In addition, it does not allow jointly solving the problem for several cameras.

D. Online Initialization of New Geometry

The number of points included in the augmented vector X_k^a can be enriched online by using the current estimations of pose and the measurements of new points not included in X_k^a. In the literature, there are some works dealing with the same problem from the point of view of Bayesian inference [33], [34]. The basic idea is to include the reconstruction of new points as new variables inside the posterior distribution $p(X_k^a|Y_1,\ldots,Y_k)$ so that the Kalman filter obtains new structure points and their statistical relationship with X_k^a.

From the point of view of Bayesian inference, we search for the joint distribution as

$$p\left(M^o, X_k^a|Y_1,\ldots,Y_k, y_{k^o}^o,\ldots,y_k^o\right) \tag{25}$$

where M^o's are the 3-D coordinates referred to O_R of a single new detected point. The set $y_{k^o}^o,\ldots,y_k^o$ corresponds to the measurements of the new point collected from time k^o up to the present, and the set $Y_1,\ldots,Y_k$ is the set of measurements of the points in the state vector.

VI. Results

Our proposal is tested by using synthetic generated data and real images taken in a room with four cameras.

A. Synthetic Data

The synthetic data are artificially generated according to the conditions that will be encountered in a real configuration. A maximum of $N_c = 8$ cameras are situated, approximately forming a rectangle of 2 m × 3 m, as shown in Fig. 7. The robot's geometry is formed by a set of points that are randomly distributed inside a cylindrical volume of half a meter of radius and 1-m height. The robot follows a differential motion model over the ground plane, which is ruled by its angular ω_k and linear speed v_k encoded in $U_k = (\omega_k(^o/s), v_k(mm/s))^T$. The vector U_k is affected by a motion noise with covariance $\Sigma_w = diag(\sigma_v^2 = 1, \sigma_\omega^2 = 10)$. The pose vector is thus composed of the 2-D position in the ground plane (x_k, y_k) and the single orientation angle α_k. The measurement noise is fixed to $\Sigma_V = 10 \cdot I_{2\times 2}$. The intrinsic parameters of each camera are variations of those encountered in a low-cost sensor with 640 × 480 pixels of resolution with a charge-coupled device (CCD) sensor of $1/3''$ size and an optic with a focal length of 6 mm. The trajectory described by the robot consists of a circumference of radius 2 m′, which takes place inside the common area viewed by the cameras. Both initialization and online experiments are tested using such trajectory, so that we can compare the accuracy in fair circumstances.

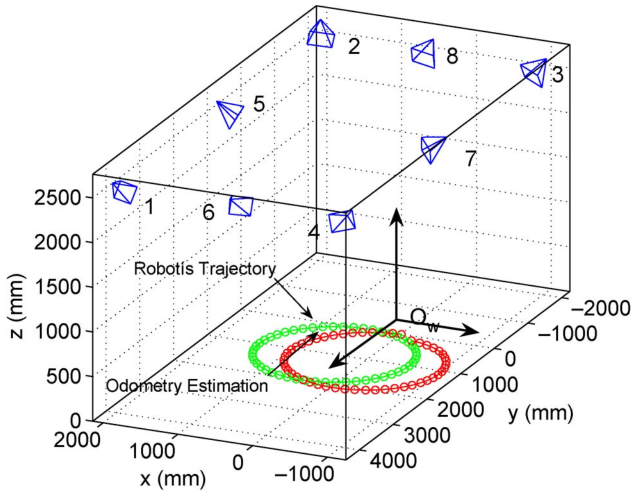


Fig. 7. Distribution of cameras and robot's trajectory used to generate synthetic data.

The experiments are divided on those dedicated to the initialization method proposed and those used to test the online algorithm.

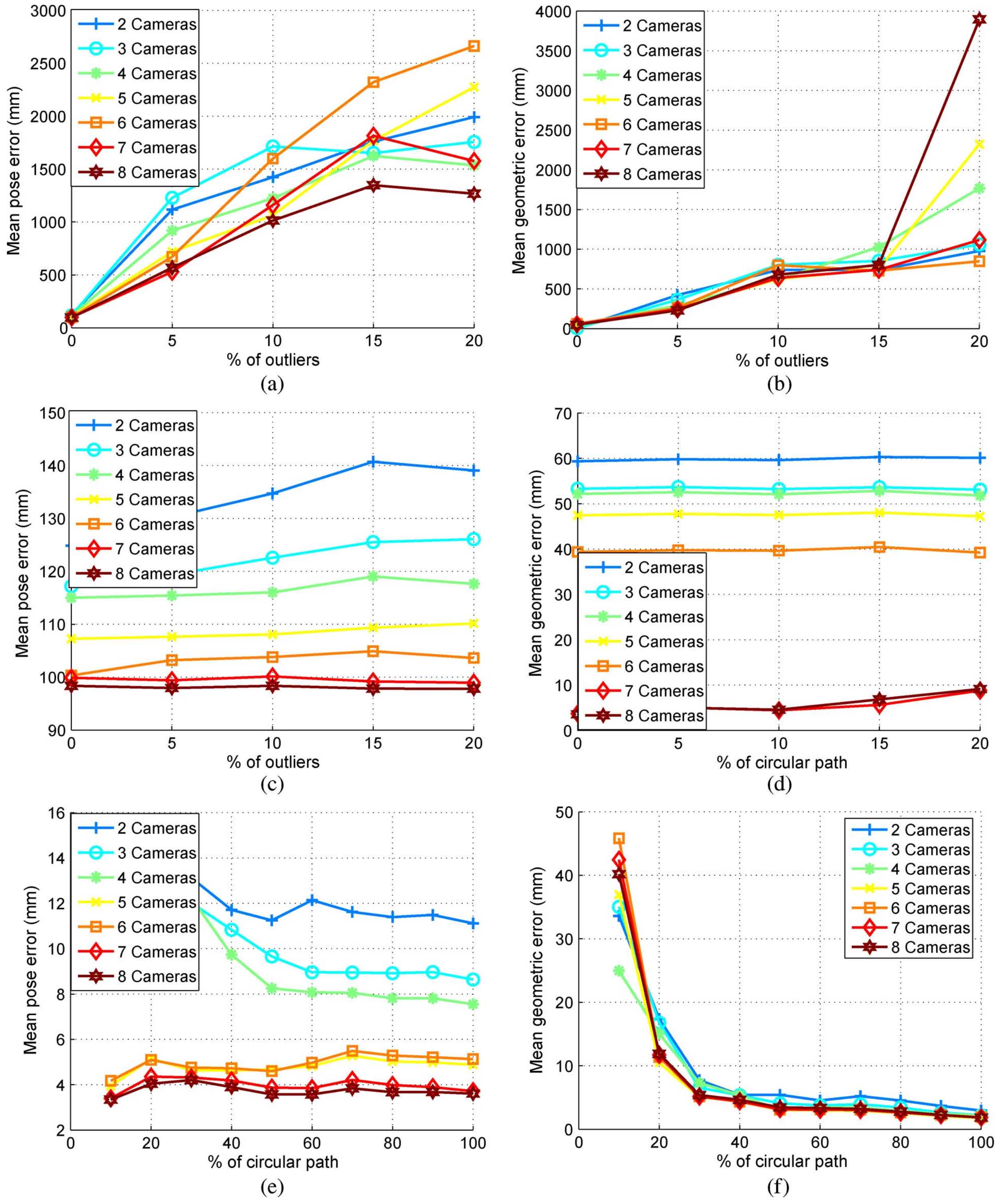
1) Initialization Method: In this experiment, we consider that each camera n observes a number $N_n = 6$ of points from the robot's structure, and, therefore, the number of points obtained after the initialization step is $N_c \times N_n$. The trajectory is downsampled, so that $K = 50$ positions. We observe that adding more time samples does not achieve better accuracy.

The following experiments are proposed.

1) Error in pose [Fig. 9(a)] and geometry [Fig. 9(b)] versus percentage of outliers in the measurements.
2) Error in pose [Fig. 9(c)] and geometry [Fig. 9(d)] versus percentage of outliers in the measurements, when the robust cost function is used.
3) Error in pose [Fig. 9(e)] and geometry versus [Fig. 9(f)] percentage of the circular path used for initialization.

The following observations have been made.

The pose and geometry error [Fig. 8(a) and (b)] of the initialization algorithm quickly grows out of a useful value even with a small portion of outliers in the measurements (5%). However, in the case of using a Cauchy robust cost function, both pose and geometry errors remain approximately constant.



(a) (b) (c) (d) (e) (f)

Fig. 8. Experiments of the initialization method using synthetic data. (a) Error in pose versus percentage of outliers. (b) Error in geometry versus percentage of outliers. (c) Error in pose versus percentage of outliers using ρ. (d) Error in geometry versus percentage of outliers using ρ. (e) Error in pose versus percentage of circumference used. (f) Error in geometry versus percentage of circumference used.

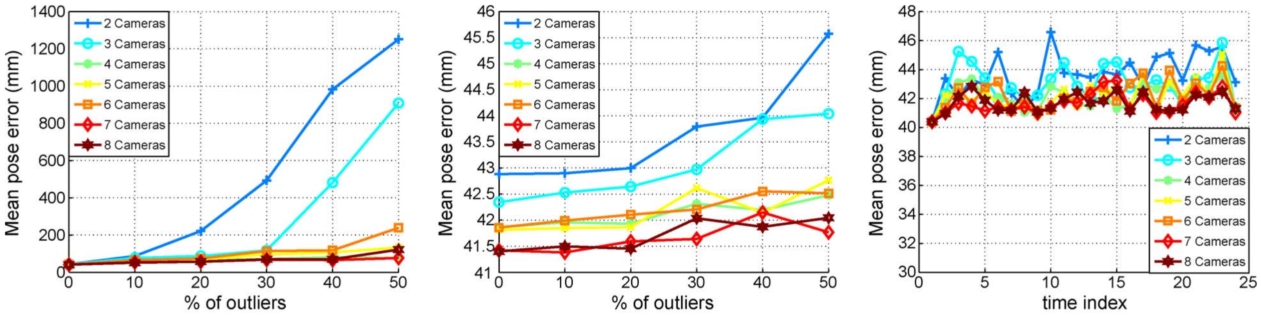
It can be observed that in [Fig. 8(c) and (d)] the pose error and geometry error decreases with the number of cameras, even taking into account that more points are included in the optimization.

In Fig. 8(e) and (f), it is clearly shown that with the 40% of the circumference of radius 2 m, it is enough for the algorithm to converge.

2) Online Method: The robust approach of the initialization algorithm is used to set up the pose and geometry X_K^a. The online algorithm covers the circular path, starting where the online algorithm ended.

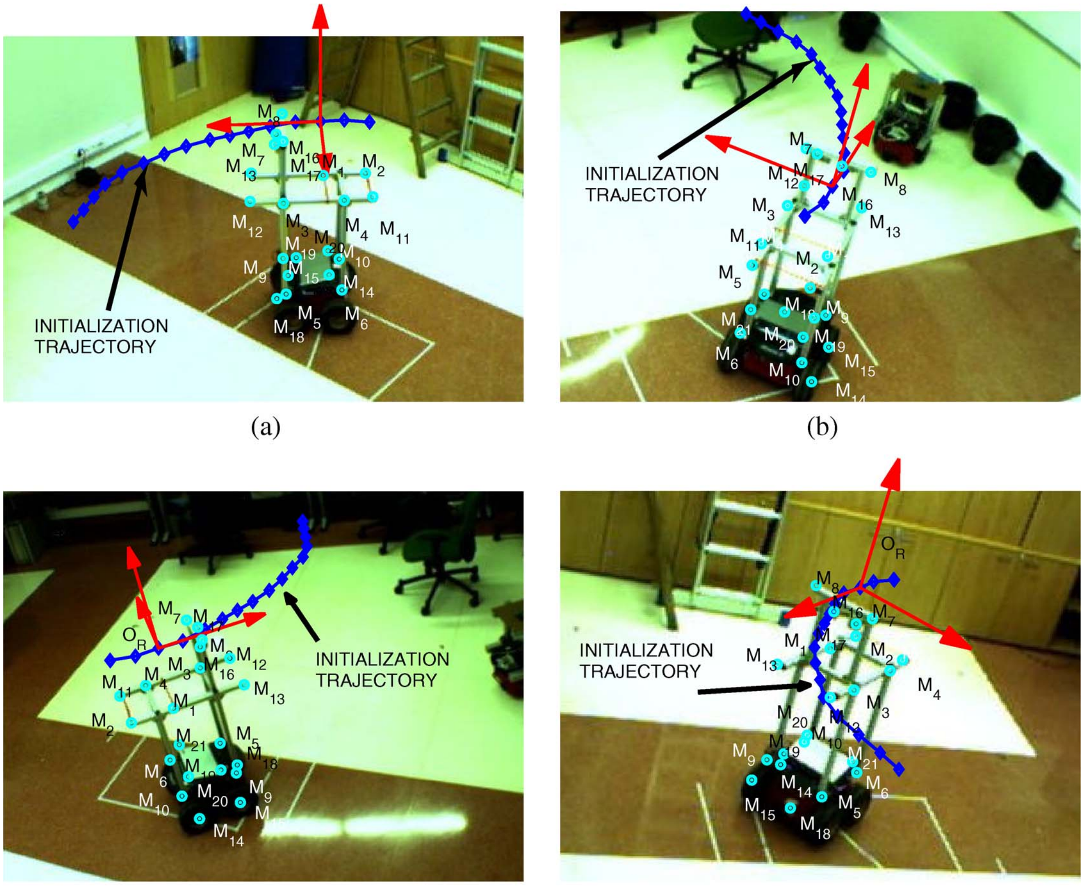
The following experiments are proposed.

1) Error in pose and geometry versus percentage of outliers without robust layer [Fig. 9(a)].



(a) (b) (c)

Fig. 9. Experiments of the online method using synthetic data.



(a) (b) (c) (d)

Fig. 10. Experiments of the initialization method using real data. (a) Initialization sequence in camera 1. (b) Initialization sequence in camera 2. (c) Initialization sequence in camera 3. (d) Initialization sequence in camera 4.

2) Error in pose and geometry versus percentage of outliers [Fig. 9(b)].
3) Error in pose and geometry versus time [Fig. 9(c)].

The following observations have been made.

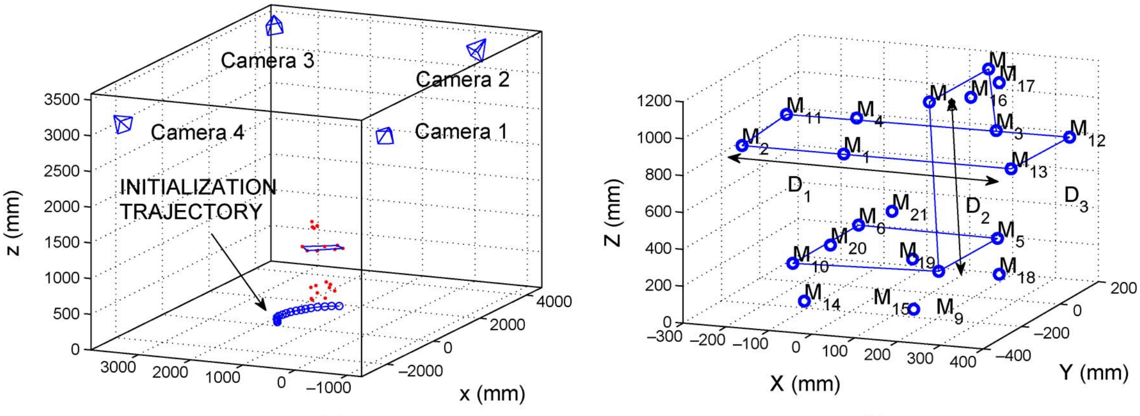
Fig. 9(a) shows that the online process without a robust layer performs good when the number of cameras increases $(N_c > 3)$. However, it has to be noted that in real experiments, it tends to be unstable. In Fig. 9(b), it can be seen that the robust step using RANSAC almost obtains similar performances with few cameras thanks to its ability to discard outliers. In Fig. 9(c), the pose error obtained in the online algorithm against the time index is shown with the intention to remark that the error remains stable and bounded during the process.

B. Real Results

The real experiment is composed of four cameras [see Fig. 11(a)], filling the same area shown in the initialization, and a mobile robot that also presents the same kind of motion model used for the synthetic data.

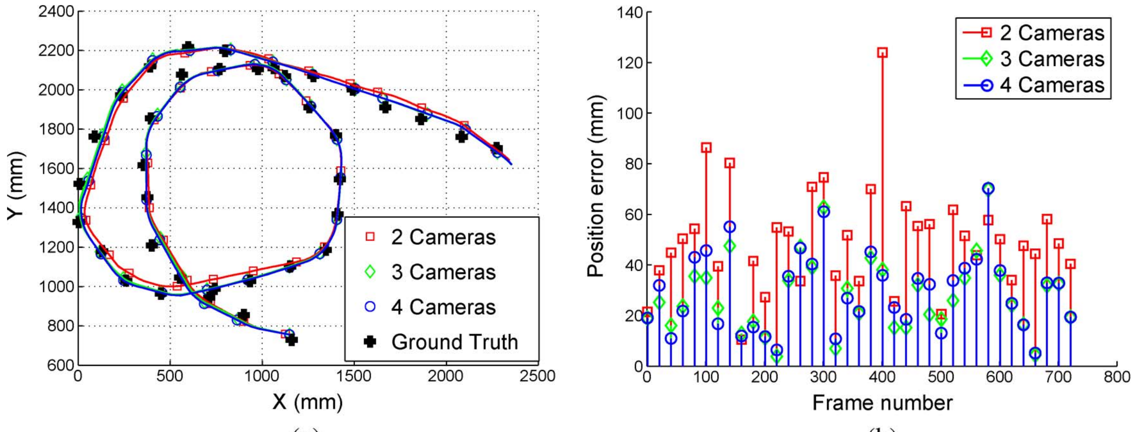
In Fig. 10(a)–(d), the projection of the 3-D model [Fig. 11(b)] obtained from the robot is shown on each of the four cameras. The model is obtained by applying the initialization process presented in this paper and using the four cameras. The initialization trajectory is shown in 3-D in Fig. 11(a).

A model composed of $N = 21$ points from the robot's structure is given after this step. The accuracy obtained in the model has been tested by measuring the relative distances between



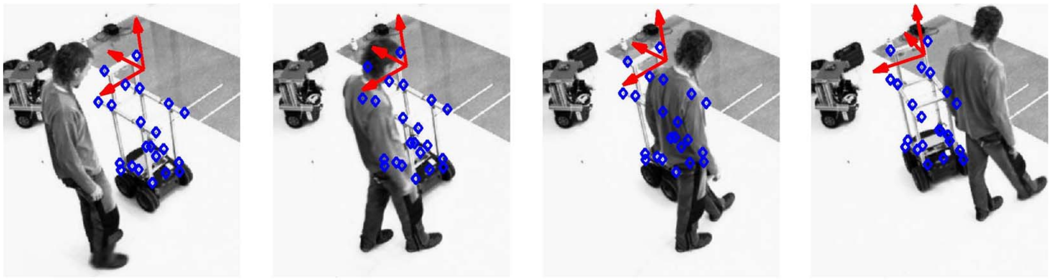
(a) (b)

Fig. 11. Experiments of the initialization method using real data. (a) Initialization sequence in 3-D. (b) Obtained 3-D model.



(a) (b)

Fig. 12. Comparison between the ground truth and the online method for several cameras. (a) Localization obtained by two, three, and four cameras versus ground truth. (b) Error in the position in function of frame number.



(a) (b) (c) (d)

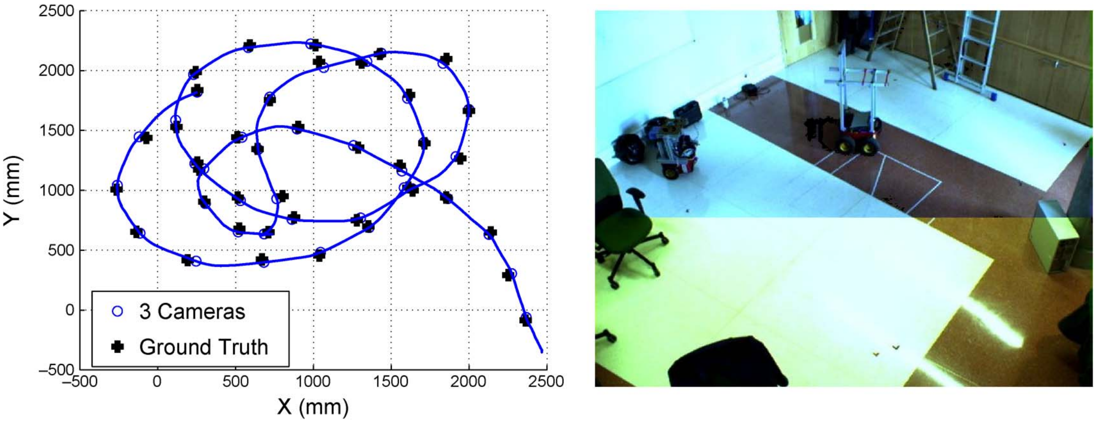
Fig. 13. Example of occlusions made by human that is crossing. (a) Frame $k = 104$. (b) Frame $k = 111$. (c) Frame $k = 120$. (d) Frame $k = 131$.

some of the points on the model in the real structure of the robot. The points used have been chosen so that they are good localized in the real structure of the robot. In Fig. 11(b), the distances measured are shown as D_1, D_2, and D_3. The average error found in the model is around 4 cm, and as can be seen in Fig. 11(b), those points that must lie on a plane (i.e., M^{12}, M^2, M^{13}, and M^{11}) are approximately coplanar.

Using distances D_1, D_2, and D_3, and by manually clicking the images (annotation procedure) over a set of identified points from the model, a ground truth measure is incorporated to the algorithm.

In Fig. 12, a comparison between the ground truth and the online algorithm is presented. The robot follows a path while a human is walking several times around it to generate occlusions, as seen in Fig. 13 for a sequence of frames. The online algorithm has been tested using four cameras, three cameras (cameras 1, 2, and 3), and two cameras (cameras 1 and 2). Their performance is compared with the ground truth in Fig. 12(a) and (b), where it showed the error in function of the frame number just to assure that the occlusions made by the human (at frames shown in the caption of Fig. 12) do not substantially affect the algorithm.

In Fig. 14, an experiment is shown, where the robot follows a long path in which there are changes of illumination and occlusions.



(a)

(b)

Fig. 14. Example of a long experiment in which there are changes in illumination. (a) Localization obtained by three cameras versus ground truth in a long experiment. (b) Changes in the illumination of the room during the test.

The proposal has been implemented in real time using a cluster architecture of four medium-cost PC computers. The algorithm runs the online algorithm at 25 fps, with the biggest bottleneck coming from the implementation of the SIFT used. The initialization process is solved by using one computer in a few seconds if the SIFT descriptors are previously obtained.

VII. Conclusion

This paper has proposed a system that achieves robot localization using several cameras without needing invasive beaconing on the robot or supervised learning tasks. Compared to the single camera solution proposed in [19], which this paper extends, the use of several cameras allows avoiding the need of using odometry systems in the robot in any stage of the algorithm, which reduces the required knowledge from the object to localize.

The two steps that compound the system, initialization of the robot's pose and geometry, and the online process are designed so that they are robust against the inclusion of outliers in the algorithm, which is of importance to achieve a reliable solution.

The tests using synthetic data show that the more the cameras, the better in general becomes the algorithm in terms of accuracy in both geometry and pose estimation. The algorithm is tested in real conditions, i.e., performing the localization of a robot using four cameras inside a room. Using a manual procedure, it is shown how the solution is accurate even in the case of using less than three cameras. Our proposal shows promising results as not only a reliable robot localization system but also any other rigid object. The extension to tackle multiple robots is quite straightforward using our approach, as the robust layers efficiently allows removing measurements that do not behave like individual rigid objects.

Acknowledgment

This paper was based on the paper ”Pose and Sparse Structure of a Mobile Robot using an External Camera” by D. Pizarro, M. Mazo, E. Santiso, and M. Marrn, which appeared in the Proceedings of the IEEE International Symposium on Intelligent Signal Processing, 2007.

References

[1] A. Pentland, “Smart rooms,” *Sci. Amer.*, vol. 274, no. 4, pp. 68–76, Apr. 1996.

[2] H. Hashimoto, J. H. Lee, and N. Ando, “Self-identification of distributed intelligent networked device in intelligent space,” in *Proc. IEEE ICRA*, 2003, vol. 3, pp. 4172–4177.

[3] J. H. Lee and H. Hashimoto, “Controlling mobile robots in distributed intelligent sensor network,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 890–902, Oct. 2003.

[4] J. Leonard and P. Newman, “Consistent, convergent, and constant-time SLAM,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, vol. 18, pp. 1143–1150.

[5] A. J. Davison and D. W. Murray, “Simultaneous localization and map-building using active vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, 2002.

[6] Y. Hada, E. Hemeldan, K. Takase, and H. Gakuhari, “Trajectory tracking control of a nonholonomic mobile robot using IGPS and odometry,” in *Proc. IEEE Int. Conf. MFI Intell. Syst.*, Jul. 30–Aug. 1, 2003, pp. 51–57.

[7] I. Fernández, M. Mazo, J. L. Lázaro, D. Pizarro, E. Santiso, P. Martín, and C. Losada, “Guidance of a mobile robot using an array of static cameras located in the environment,” *Auton. Robots*, vol. 23, no. 4, pp. 305–324, Nov. 2007.

[8] K. Morioka, X. Mao, and H. Hashimoto, “Global color model based object matching in the multi-camera environment,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2644–2649.

[9] J. Chung, N. Kim, J. Kim, and C.-M. Park, “Postrack: A low cost real-time motion tracking system for VR application,” in *Proc. 7th Int. Conf. Virtual Syst. Multimed.*, Oct. 2001, pp. 383–392.

[10] T. Sogo, H. Ishiguro, and T. Ishida, “Acquisition of qualitative spatial representation by visual observation,” in *Proc. IJCAI*, 1999, pp. 1054–1060.

[11] E. Kruse and F. M. Wahl, “Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning,” in *Proc. ICRA*, 1998, pp. 662–667.

[12] A. Hoover and B. D. Olsen, “Sensor network perception for mobile robotics,” in *Proc. IEEE ICRA*, 2000, pp. 342–347.

[13] P. Steinhaus, M. Ehrenmann, and R. Dillmann, “MEPHISTO: A Modular and Extensible Path Planning System Using Observation,” in *Lecture Notes in Computer Science*, vol. 1542. Berlin, Germany: Springer-Verlag, 1999, pp. 361–375.

[14] D. Nister, “Preemptive RANSAC for live structure and motion estimation,” *Mach. Vis. Appl.*, vol. 16, no. 5, pp. 321–329, Dec. 2005.

[15] A. W. Fitzgibbon and A. Zisserman, “Automatic 3D model acquisition and generation of new images from video sequences,” in *Proc. Eur. Signal Process. Conf.*, 1998, pp. 1261–1269.

[16] F. Schaffalitzky and A. Zisserman, “Multi-view matching for unordered image sets, or ’how do I organize my holiday snaps?”’in *Proc. ECCV*, 2002, vol. 1, pp. 414–431.

[17] I. Gordon and D. G. Lowe, “Toward category-level object recognition,” in *What and Where: 3D Object Recognition With Accurate Pose*. Berlin, Germany: Springer-Verlag, 2006, pp. 67–82.

[18] R. Szeliski and S. B. Kang, “Recovering 3D shape and motion from image streams using nonlinear least squares,” in *Proc. IEEE Comput. Soc. Conf. CVPR*, 1993, pp. 752–753.

[19] D. Pizarro, E. Santiso, M. Mazo, and M. Marron, “Pose and sparse structure of a mobile robot using an external camera,” in *Proc. IEEE Int. Symp. Intell. Signal Process.*, 2007, pp. 389–394.

[20] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Conf. CVPR*, 1994, pp. 593–600.

[21] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[22] H. Moravec, *Robot Rover Visual Navigation*. Ann Arbor, MI: UMI Res. Press, 1981.

[23] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proc. 4th Alvey Vis. Conf.*, Manchester, U.K., 1988, pp. 189–192.

[24] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. Int. Conf. Comput. Vis.*, Kerkyra, Greece, Sep. 1999, pp. 1150–1157.

[25] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, “Fully automated and stable registration for augmented reality applications,” in *Proc. 2nd IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2003, pp. 93–102.

[26] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. DARPA Image Understanding Workshop*, 1981, pp. 121–130.

[27] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment—A modern synthesis,” in *Vision Algorithms: Theory and Practice*, vol. 1883. Berlin, Germany: Springer-Verlag, 2000, pp. 298–372.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[29] R. C. Bolles and M. A. Fischler, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[30] B. M. Haralick, C. N. Lee, K. Ottenberg, and M. Nölle, “Review and analysis of solutions of the three point perspective pose estimation problem,” *Int. J. Comput. Vis.*, vol. 13, no. 3, pp. 331–356, Dec. 1994.

[31] D. Nister, “A minimal solution to the generalised 3-point pose problem,” in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 27–Jul. 2, 2004, vol. 1, pp. I-560–I-567.

[32] F. Moreno-Noguer, V. Lepetit, and P. Fua, “Accurate non-iterative O(n) solution to the PnP problem,” in *Proc. IEEE 11th ICCV*, 2007, pp. 1–8.

[33] A. J. Davison, I. Reid, N. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[34] J. Civera, A. J. Davison, and J. M. M. Montiel, “Inverse depth to depth conversion for monocular SLAM,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2778–2783.

Daniel Pizarro (S’03) received the M.S. degree in electrical engineering and the Ph.D. degree from the University of Alcala, Alcala de Henares, Spain in 2003 and 2008, respectively.

His research interests are focused on intelligent spaces and computer vision applied to robotics.

Manuel Mazo (M’92) received the M.S. and Ph.D. degrees from the Polytechnic University of Madrid, Madrid, Spain, in 1982 and 1988, respectively.

He is currently a Full Professor with the Department of Electronics, Universidad de Alcala, Alcala de Henares, Spain. His research interests include multisensor integration (ultrasonic, infrared, and artificial vision) for indoor localization, computer vision, and electronic control systems applied to railway safety, mobile robots, assistance technologies for physically disabled people, and intelligent spaces.

Enrique Santiso received the Ph.D. degree in telecommunications from Universidad de Alcala, Alcala de Henares, Spain.

He is currently with the Department of Electronics, Universidad de Alcala. His interests are in the fields of intelligent spaces.

Marta Marron (M’04) received the M.S. degree in electrical engineering from Universidad de Alcala, Alcala de Henares, Spain, in 2000.

She was a Researcher from 1996 to 2001 and is currently an Associate Professor with the Department of Electronics, Universidad de Alcala. Her research interests include multisensor indoor localization (intelligent spaces), computer vision, probabilistic algorithms, electronics control systems and robotics in general, and personal mobile robots in particular applied to assistance technologies.

Ignacio Fernandez received the Ph.D. degree from Universidad de Alcala, Alcala de Henares, Spain, in 2005.

He is currently a Professor with the Department of Electronics, Universidad de Alcala. His research interests are focused on intelligent spaces.