Sensors **2010**, *10*, 3261-3279; doi:10.3390/s100403261

OPEN ACCESS

sensors

ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Multi-Camera Sensor System for 3D Segmentation and Localization of Multiple Mobile Robots

Cristina Losada *, Manuel Mazo, Sira Palazuelos, Daniel Pizarro and Marta Marr ón

Electronics Department, University of Alcal á, Campus Universitario s/n, 28805, Alcal áde Henares, Madrid. Spain. E-Mails: mazo@depeca.uah.es (M.M.); sira@depeca.uah.es (S.P.); pizarro@depeca.uah.es (D.P.); marta@depeca.uah.es (M.M).

* Author to whom correspondence should be addressed; E-Mail: losada@depeca.uah.es; Tel.: +34-918-856-592; Fax: +34-918-856-591.

Received: 5 February 2010; in revised form: 23 March 2010 / Accepted: 26 March 2010 / Published: 1 April 2010

Abstract: This paper presents a method for obtaining the motion segmentation and 3D localization of multiple mobile robots in an intelligent space using a multi-camera sensor system. The set of calibrated and synchronized cameras are placed in fixed positions within the environment (intelligent space). The proposed algorithm for motion segmentation and 3D localization is based on the minimization of an objective function. This function includes information from all the cameras, and it does not rely on previous knowledge or invasive landmarks on board the robots. The proposed objective function depends on three groups of variables: the segmentation boundaries, the motion parameters and the depth. For the objective function minimization, we use a greedy iterative algorithm with three steps that, after initialization of segmentation boundaries and depth, are repeated until convergence.

Keywords: multi-camera sensor; intelligent space; motion segmentation, 3D positioning; mobile robots

1. Introduction

A common problem in the field of autonomous robots is how to obtain the position and orientation of the robots within the environment with sufficient accuracy. Several methods have been developed to carry out this task. The localization methods can be classified into two groups: those that require sensors onboard the robots [1] and those that incorporate sensors within the work environment [2].

Although the use of sensors within the environment requires the installation of an infrastructure of sensors and processing nodes, it presents several advantages, it allows reducing the complexity of the electronic onboard the robots and facilitates simultaneous navigation of multiple mobile robots within the same environment without increasing the complexity of the infrastructure. Moreover, the information obtained from the robots movement is more complete, thereby it is possible to obtain information about the position of all of the robots, facilitating cooperation between them. This alternative includes "intelligent environments" [3,4] characterized by the use of an array of sensors located in fixed positions and distributed strategically to cover the entire field of movement of the robots. The information provided by the sensors should allow the localization of the robots and other mobile objects accurately.

The sensor system in this work is based on an array of calibrated and synchronized cameras. There are several methods to locate mobile robots using an external camera array. The most significant approaches can be divided into two groups. The first group includes those works that make use of strong prior knowledge by using artificial landmarks attached to the robots [5,6]. The second group includes the works that use the natural appearance of the robots and the camera geometry to obtain the positions [2]. Intelligent spaces have a wide range of applications, especially in indoor environments such as homes, offices, hospital or industrial environments, where sensors and processing nodes are easy to install.

The proposal presented in this paper is included in the second group. It uses a set of calibrated cameras, placed in fixed positions within the environment to obtain the position of the robots and their orientation. This proposal does not rely on previous knowledge or invasive landmarks. Robots segmentation and position are obtained through the minimization of an objective function. There are many works that use an objective function [7,8]. However, the works in [7,8] present several disadvantages such as high computational cost or dependence on the initial values of the variables. Moreover, these methods are not robust because they use information from a single camera.

It is noteworthy that, although the proposal in this work has been evaluated in a small space (ISPACE-UAH), it can be easily extended to a larger number of rooms, corridors, *etc.* It allows covering a wider area, by adding more cameras to the environment and properly dimensioning the image processing hardware.

2. Multi-Camera Sensor System

The sensor system used in this work is based on a set of calibrated and synchronized cameras placed in fixed positions within the environment (Intelligent Space of University of Alcalá, ISPACE-UAH). These cameras are distributed strategically to cover the entire field of movement of the robots. As has been explained in the introduction, the use of sensors within the environment presents several advantages, it allows reducing the complexity of the electronic onboard the robots and facilitates simultaneous navigation of multiple mobile robots within the same environment without increasing the complexity of the infrastructure. Moreover, the information obtained from the movement of the robots is more complete, thereby it is possible to obtain information about the position of all of the robots, facilitating the cooperation between them.

2.1. Hardware Architecture

The hardware deployed in the ISPACE-UAH consists basically of a set of cameras with external trigger synchronization, a set of acquisition and processing nodes, mobile robots and a Local Area Network (LAN) infrastructure, that includes a wireless channel that the robots use to provide information from their internal sensors and to receive motion commands. All the cameras are built with a CCD sensor with a resolution of 640 × 480 and a size of 1/2" (8mm diagonal). The optical system is chosen with a focal length of 6.5 mm which gives about 45 °of Field of View (FOV). Each camera is connected to a processing node through a Firewire (IEEE1394) local bus, which allows 25 fps RGB image acquisition speed and control of several camera parameters such as the exposure, gain or trigger mode.

The processing nodes are general purpose multi-core PC platforms with Firewire ports and Gigabit Ethernet hardware which allows them to connect to the LAN network. Each node has the capability of controlling and processing the information from one or several cameras. In the present paper each node is connected to a single camera.

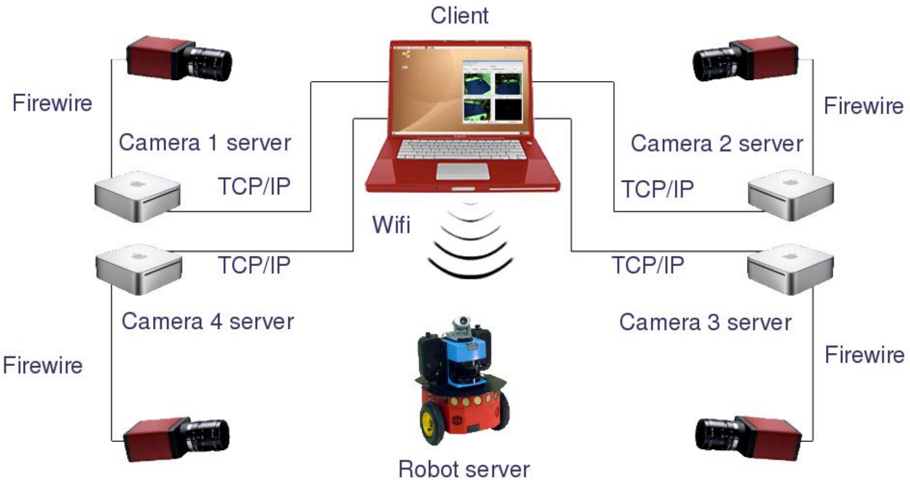
The robotic platforms used in all experiments are provided by "Active Media Robotics". More specifically, the model used is the P3-DX, which is a differential wheeled robot of dimensions 44.5 × 40 × 24.5 cm, equipped with low level controllers for each wheel, odometry systems and an embedded PC platform with IEEE 802.11 wireless network hardware.

2.2. Software Architecture

The software architecture chosen is a client-server system using common TCP/IP connections, where some servers (*i.e.,* processing nodes and robots internal PCs) receive commands and requests from a client (*i.e.,* computer or data storage device for batch tests).

Each processing node acts as a server that preprocesses the images and sends the results to the client platform. The preprocessing task of the servers consists of operations that can be clearly developed separately for each camera, such as image segmentation, image warping for computing occupancy grids, compression or filtering. The internal PC in each robot acts as a server which allows receiving control commands from a client and sending back the odometry readings obtained from its internal sensors. On the other hand, the client is in charge of performing data fusion using all information provided by the servers in order to achieve a certain task. In the case of the application proposed in this paper, the client receives robots odometry information, 3D occupancy grid representation of the scene and the client itself assures synchronization of the odometry values with the camera acquisition. In Figure 1, a general diagram of the proposed hardware/software architecture is shown.

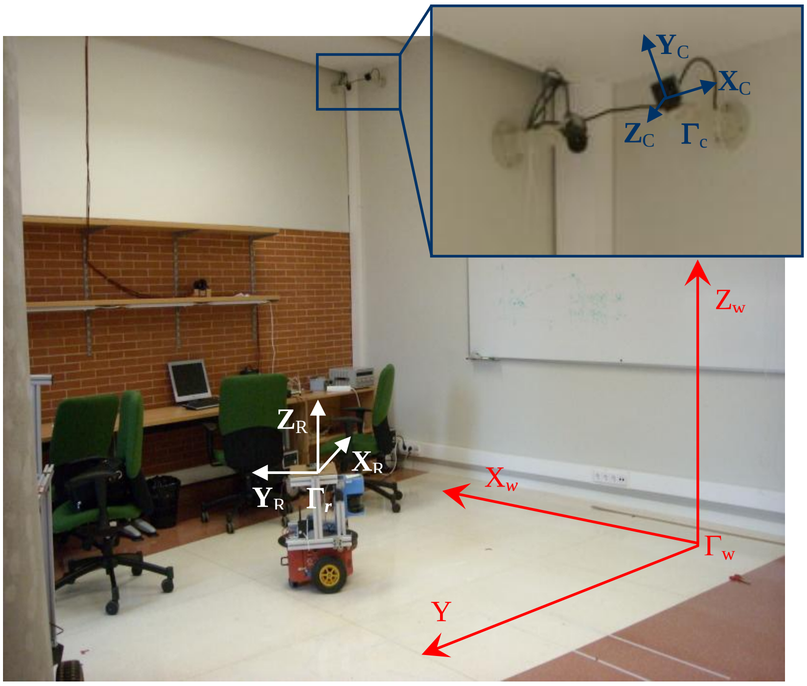
Figure 1. General diagram of the hardware/software architecture in the ISPACE-UAH.



2.3. Reference Systems in the Intelligent Space

Before presenting the proposed algorithm for motion segmentation and 3D positioning of multiple mobile robots using an array of cameras, it is important to define the different coordinate systems used in this work. In the intelligent space, the 3D coordinates of a point $\mathbf{P} = (X, Y, Z)^T$ can be expressed in different coordinate systems. There is a global reference system named "world coordinate system" and represented by Γ_w. There is also a local reference system associated with each camera (Γ_{ci}, $i = 1,\ldots,n_c$) whose origin is located in the center of projection. These coordinate systems are represented in Figure 2, where world coordinate system (Γ_w) has been represented in red color and the coordinate systems associated to the cameras (Γ_{ci}) have been represented in blue color.

Figure 2. Reference systems in the intelligent space (ISPACE-UAH): World coordinate system (Γ_w) in red color. Camera coordinate system (Γ_{ci} $i = 1,2,\ldots,n_c$) in blue color.



The cameras used in this work are placed in fixed positions within the environment (ISPACE-UAH). These cameras are distributed strategically to cover the entire field of movement of the robots. Figure 3 shows the spatial distribution, and the area covered by the cameras used in this work.

Cameras are modeled as pinhole cameras. This is a simple model that describes the mathematical relationship between the coordinates of a 3D point in the camera coordinate system (Γ_c) and its projection onto the image plane in an ideal camera without lenses through the expressions in Equation (1) where f_x, f_y are the camera focal lengths along x and y axis:

$$x = f_x \frac{X_c}{Z_c}, \quad y = f_y \frac{Y_c}{Z_c} \tag{1}$$

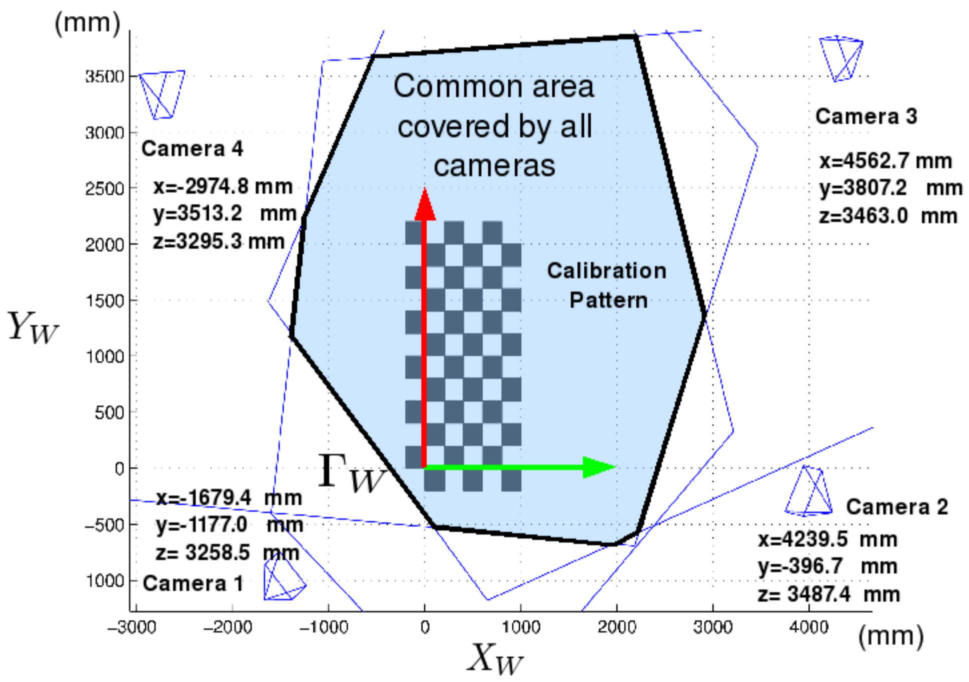
If the origin of the image coordinate system is not in the center of the image plane, the displacement (s_1,s_2) from the origin to the center of the image plane is included in the projection equations, obtaining the perspective projection Equation (2):

$$x = f_x \frac{X_c}{Z_c} + s_1, \quad y = f_y \frac{Y_c}{Z_c} + s_2 \tag{2}$$

These equations can be expressed using homogeneous coordinates, as shown in Equation (3):

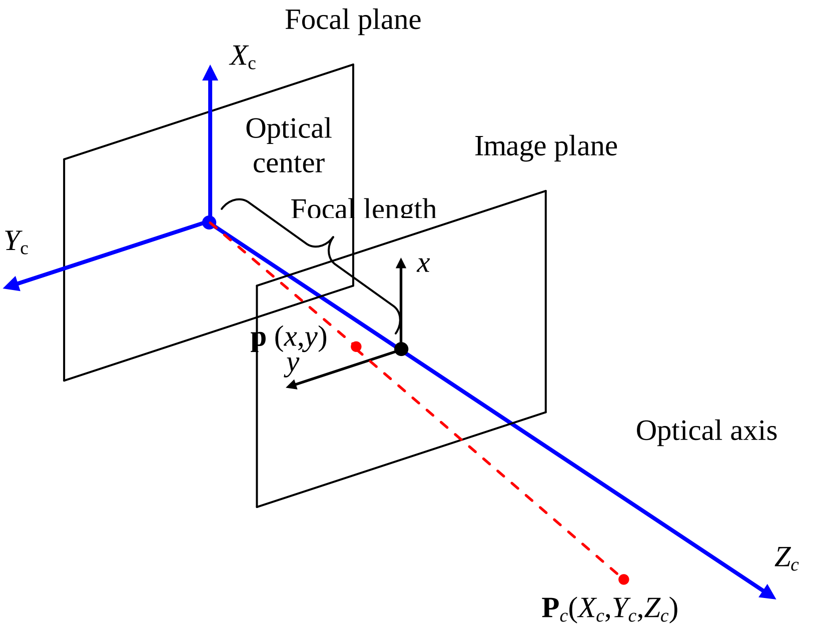
$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & s_1 \\ 0 & f_y & s_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{3}$$

Figure 3. Spatial distribution of the cameras used in the experiments.



The geometry related to the mapping of a pinhole camera is illustrated in the Figure 4.

Figure 4. Geometric model of a pinhole camera. In this model the optical center coincides with the origin of the camera coordinate system (Γ_c) represented in blue color. The image reference system (x,y) is drawn in black color.



3. Algorithm for motion segmentation and positioning

Using the work of Sekkati and Mitiche [7] as a starting point, in this work motion segmentation and 3D localization are obtained through the minimization of an objective function. The objective function proposed in [7] [and shown in Equation (4)] depends on three groups of variables: a set of curves that defines the mobile robot segmentation boundaries in the image plane $\{\gamma_k\}_{k=1}^{N-1}$, the components of linear and angular velocity of each robot $\{\mathbf{v}_{ck}\}_{k=1}^{N}$, $\{\boldsymbol{\omega}_{ck}\}_{k=1}^{N}$ and the depth. In Equation (4) λ and μ are positive, real constants. These constants weight the contribution of each term to the objective function:

$$E\left[\{\gamma_k\}_{k=1}^{N-1},\{\mathbf{v}_{ck}\}_{k=1}^{N},\{\boldsymbol{\omega}_{ck}\}_{k=1}^{N},Z\right]=\sum_{k=1}^{N}\left[\int_{\Omega_k}\psi_k^2(\mathbf{x})d\mathbf{x}+\mu\int_{\Omega_k}g(\|\nabla Z\|)d\mathbf{x}\right]+\sum_{k=1}^{N-1}\lambda\oint_{\gamma_k}ds\,,\quad \begin{matrix}\lambda,\mu\in\Re\\ \lambda,\mu>0\end{matrix} \qquad (4)$$

As can be observed in Equation (4), the objective function proposed by Sekkati and Mitiche in [7] contains three different terms. The first term measures the conformity of the 3D interpretation within each region of segmentation to the image sequence spatiotemporal variations. This measure is given by the three-dimensional brightness constraint for rigid objects proposed in [7] and shown in Equation (5). The remaining two terms in Equation (4) are regularization terms, one for depth via a boundary preserving function (g(a)) and the other one for segmentation boundaries:

$$I_t+\mathbf{s}\frac{\mathbf{v}_c}{Z_c}+\mathbf{q}\omega_c=0 \qquad (5)$$

In Equation (5), **s** and **q** are two vectors that depend on the image spatiotemporal derivatives [I_x, I_y, I_t], the coordinates of each point in the image plane (x, y) and the focal lengths f_x, f_y:

$$\mathbf{s}=\begin{pmatrix} f_x I_x \\ f_y I_y \\ -(x-s_1)I_x-(y-s_2)I_y \end{pmatrix}^T \qquad \mathbf{q}=\begin{pmatrix} -f_y I_y-\frac{y-s_2}{f_y}\left((x-s_1)I_x+(y-s_2)I_y\right) \\ -f_x I_x-\frac{x-s_1}{f_x}\left((x-s_1)I_x+(y-s_2)I_y\right) \\ -\frac{f_x}{f_y}(y-s_2)I_x+\frac{f_y}{f_x}(x-s_1)I_y \end{pmatrix}^T$$

In [7], the minimization of the objective function (4) is carried out using a greedy algorithm which consists of three iterated steps. After the initialization of the segmentation boundaries and depth, the three steps are repeated until the convergence of the algorithm. In each step, two of the three groups of variables are fixed, and the equation is solved for the remaining one. After minimization, motion segmentation of the mobile robots is obtained. However, proposal of [7] presents several disadvantages such as high computational cost, or dependence on the initial values of the variables (segmentation boundaries and depth). Moreover, this method is not robust, and it does not allow obtaining 3D position of the mobile robots because it uses information from a single camera.

Since there are multiple cameras available in the intelligent space, we have proposed a new objective function that includes information of all the cameras. The minimization of the proposed function allows us to obtain both motion segmentation and 3D position of multiple mobile robots in an intelligent space. The use of multiple cameras increases notably the robustness of the system. It also improves the accuracy of the results (segmentation and 3D positioning).

In addition, the proposed solution allows segmenting and estimating the 3D position of the mobile robots even if they are not seen by some of the cameras. Even in the worst case, if all the cameras lose some of the robots, they can still be controlled by the intelligent space. In this case, the positions of the unseen robots are estimated through the measurements of the odometry sensors they have onboard.

3.1. 3D Brightness Constraint for a Multi-camera Sensor System

Before presenting the objective function for multiple cameras, it is necessary to describe the 3D brightness constraint for multiple cameras, that is a generalization of the 3D brightness constraint for a single camera presented in [7].

Let $\mathbf{P}_w=(X_w, Y_w, Z_w)^T$ be the 3D coordinates of point **P** on a mobile robot related to the world coordinate system Γ_w. Let $\mathbf{v}_w=\left(v_w^x\ v_w^y\ v_w^z\right)^T$ and $\boldsymbol{\omega}_w=\left(\omega_w^x\ \omega_w^y\ \omega_w^z\right)^T$ be, respectively, the components of the linear and angular velocity of the robot motion in Γ_w. Then, the velocity of **P**, relative to Γ_w, is given by Equation (6):

$$\dot{\mathbf{P}}_w=\left(\dot{X}_w \quad \dot{Y}_w \quad \dot{Z}_w\right)^T=\mathbf{v}_w+\boldsymbol{\omega}_w\times\mathbf{P}_w \tag{6}$$

In the same way, if $\mathbf{P}_c=(X_c, Y_c, Z_c)^T$ are the coordinates of **P** relative to Γ_c and $\mathbf{v}_c=\left(v_c^x\ v_c^y\ v_c^z\right)^T$ and $\boldsymbol{\omega}_c=\left(\omega_c^x\ \omega_c^y\ \omega_c^z\right)^T$ are the components of the linear and angular velocity of the robot motion in Γ_c. The velocity of **P** relative to Γ_c is given by Equation (7):

$$\dot{\mathbf{P}}_c=\left(\dot{X}_c \quad \dot{Y}_c \quad \dot{Z}_c\right)^T=\mathbf{v}_c+\boldsymbol{\omega}_c\times\mathbf{P}_c \tag{7}$$

Let $\mathbf{R}_{wc}$ be the (3 × 3) rotation matrix and $\mathbf{T}_{wc}$ the (1 × 3) translation vector which represent the coordinate transformation from the world coordinate system (Γ_w) to the camera coordinate system (Γ_c). The coordinate transformation is carried out using the expression in Equation (8).

$$\mathbf{P}_c = \mathbf{R}_{wc}\mathbf{P}_w + \boldsymbol{T}_{wc} \tag{8}$$

Differentiating the Equation (8) with respect to time, and substituting the expressions of the velocities in Γ_w (Equation (6)) and Γ_c (Equation (7)), Equation (9) is obtained:

$$\mathbf{v}_c + \boldsymbol{\omega}_c \times \mathbf{P}_c = \mathbf{R}_{wc}\left(\mathbf{v}_w + \boldsymbol{\omega}_w \times \mathbf{P}_w\right) \tag{9}$$

Taking into account that cross product $\boldsymbol{\omega} \times \mathbf{P}$ can be expressed as a scalar product $\hat{\boldsymbol{\omega}} \cdot \mathbf{P}$, where $\hat{\boldsymbol{\omega}}$ is the following antisymmetric matrix:

$$\hat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega^z & \omega^y \\ \omega^z & 0 & -\omega^x \\ -\omega^y & \omega^x & 0 \end{pmatrix}$$

Equation (9) can be rewritten to obtain Equation (10), where the components of linear and angular velocities in Γ_c ($\mathbf{v}_c$, $\boldsymbol{\omega}_c$) are expressed as a function of the components of velocity in Γ_w ($\mathbf{v}_w$, $\boldsymbol{\omega}_w$) and the transformation matrices ($\mathbf{R}_{wc}$, $\mathbf{T}_{wc}$):

$$\begin{gathered} \mathbf{v}_c = \mathbf{R}_{wc}\mathbf{v}_w - \mathbf{R}_{wc}\boldsymbol{\omega}_w\mathbf{R}_{wc}^T\mathbf{T}_{wc} \\ \boldsymbol{\omega}_c = adj\left(\mathbf{R}_{wc}\right)\boldsymbol{\omega}_w \end{gathered} \tag{10}$$

Let (x, y) be the coordinates of the projection of a point **P** on the image plane, the derivative of the perspective projection equations (Equation (2)) with respect to time, and the subsequent substitution of the expression of the velocity components of **P** in Γ_c allows us to obtain the following equations for motion components in the image plane $(\dot{x}, \dot{y})$:

$$\dot{x} = \tfrac{1}{Z_c}\left(f_x R_{wc}^1 - x R_{wc}^3\right)\mathbf{v}_w + \mathbf{q}_u adj(\mathbf{R}_{wc})\boldsymbol{\omega}_w \tag{11}$$

$$\dot{y} = \tfrac{1}{Z_c}\left(f_y R_{wc}^2 - y R_{wc}^3\right)\mathbf{v}_w + \mathbf{q}_v adj(\mathbf{R}_{wc})\boldsymbol{\omega}_w \tag{12}$$

where R_{wc}^i is the i-th row in the rotation matrix from Γ_w to Γ_c ($\mathbf{R}_{wc}$) and $\mathbf{q}_u$, $\mathbf{q}_v$ are the following vectors:

$$\mathbf{q}_u = \left[x\left(\frac{t_{wc}^y}{Z_c} - \frac{y}{f_y}\right) \quad \left(f_x + \frac{x^2}{f_x} - \frac{1}{Z_c}\left(f_x t_{wc}^z + x t_{wc}^x\right)\right) \quad f_x\left(\frac{t_{wc}^y}{Z_c} - \frac{y}{f_y}\right) \right]$$

$$\mathbf{q}_v = -\left[\left(f_y + \frac{y^2}{f_y} - \frac{1}{Z_c}\left(f_y t_{wc}^z + y t_{wc}^y\right)\right) \quad y\left(\frac{t_{wc}^x}{Z_c} - \frac{x}{f_x}\right) \quad f_y\left(\frac{t_{wc}^x}{Z_c} - \frac{x}{f_x}\right) \right]$$

The substitution of velocity components in the image plane $(\dot{x}, \dot{y})$ in the well known brightness constraint ($I_x\dot{x} + I_y\dot{y} + I_t = 0$) allows to obtain a 3D brightness constraint for rigid objects in terms of the linear and angular velocity components in Γ_w ($\mathbf{v}_w$ and $\boldsymbol{\omega}_w$). This constraint is shown in Equation (13):

$$\Psi_k(\mathbf{x}) = I_t + \mathbf{s} \cdot \mathbf{R}_{wc}\frac{\mathbf{v}_w}{Z_c} + \mathbf{q} \cdot adj\left(\mathbf{R}_{wc}\right)\boldsymbol{\omega}_w + T_{wc}^T\mathbf{r} \cdot adj\left(\mathbf{R}_{wc}\right)\frac{\boldsymbol{\omega}_w}{Z_c} = 0 \tag{13}$$

where the matrices **s, q** and **r** in Equation (13) are given, respectively, by equations (14), (15) and (16):

$$\mathbf{s} = \begin{pmatrix} f_x I_x & f_y I_y & -\left(x I_x + y I_y\right) \end{pmatrix} \tag{14}$$

$$\mathbf{q} = \left(-f_v I_y - \tfrac{y}{f_v}\left(x \cdot I_x + y \cdot I_y\right) \quad f_u I_x + \tfrac{x}{f_u}\left(x \cdot I_x + y \cdot I_y\right) \quad -\tfrac{f_u}{f_v} y \cdot I_x + \tfrac{f_v}{f_u} x \cdot I_y \right) \tag{15}$$

$$\mathbf{r} = \begin{pmatrix} 0 & -\left(xI_x + yI_y\right) & -f_v I_y \\ \left(xI_x + yI_y\right) & 0 & f_u I_x \\ f_v I_y & -f_u I_x & 0 \end{pmatrix} \tag{16}$$

3D brightness constraint in Equation (13) must be satisfied in all of the n_c cameras. Knowing it, we define a new 3D brightness constraint for rigid objects which includes all the information provided by the n_c cameras available in the intelligent space [Equation (17)]:

$$\psi_{ki}(\mathbf{x}) = I_{ti} + \mathbf{s}_i \cdot \mathbf{R}_{wci} \cdot \frac{\mathbf{v}_{wk}}{Z_{ci}} + \mathbf{q}_i \cdot adj\left(\mathbf{R}_{wci}\right)\boldsymbol{\omega}_{wk} + \mathbf{t}_{wci}^{T}\mathbf{r}_i \cdot adj\left(\mathbf{R}_{wci}\right)\frac{\boldsymbol{\omega}_{wk}}{Z_{ci}} \qquad \begin{matrix} k = 1,2,\ldots,N \\ i = 1,2,\ldots,n_c \end{matrix} \tag{17}$$

Constraint in Equation (17) is defined for each region, in each camera. If there are *N*-1 robots in a scene, the scene is divided into *N* regions (region *N* corresponds to the background). We have added two subscripts to denote a region: subscript *k* (*k*=1,2,…,*N*), which indicates the region in each image, and subscript *i* (*i*=1,2,…,n_c) which indicates the camera. It is worth pointing out that the components of the linear and angular velocity in the world coordinate system do not include the subscript *i* to indicate the camera because these velocities are equal for the n_c cameras.

3.2. Objective Function for a Multi-camera Sensor System

The objective function for a multi-camera sensor system proposed in this work, Equation (18), depends on three groups of variables:

- A set of *N*-1 curves $\left\{\gamma_{ki}\right\}_{k=1,\ldots,N-1}^{i=1,\ldots,n_c}$ that divide each image in *N* regions. These curves define the boundaries of the segmentation in the images acquired by each camera.
- The components of the linear and angular velocities $\left\{\mathbf{v}_{wk}\right\}_{k=1}^{N}$, $\left\{\boldsymbol{\omega}_{wk}\right\}_{k=1}^{N}$ of the (*N*-1) mobile robots and background. These velocities are related to the world reference system Γ_w and are equal for the n_c cameras.
- The depth (distance from each 3D point **P** to each camera). The value of depth in each point coincides with the Z_{ci} coordinate of the point **P** related to the coordinate system of the camera *i* Γ_{ci}:

$$E\left[\left\{\gamma_{ki}\right\}_{k=1,\ldots,N-1}^{i=1,\ldots,n_c}, \left\{\mathbf{T}_{wk}\right\}_{k=1}^{N}, \left\{\boldsymbol{\omega}_{wk}\right\}_{k=1}^{N}, \left\{Z_{ci}\right\}_{i=1}^{n_c}\right] = \sum_{k=1}^{N}\sum_{i=1}^{n_{ck}}\left[\int_{\Omega_{ki}} \psi_{ki}^{2}(\mathbf{x})d\mathbf{x} + \mu \int_{\Omega_{ki}} g\left(\left\|\nabla Z_{ci}\right\|\right)d\mathbf{x}\right] + \sum_{k=1}^{N-1}\sum_{i=1}^{n_{ck}} \lambda \oint_{\gamma_{ki}} ds \tag{18}$$

In Equation (18), ψ_{ki} is the 3D brightness constraint (defined in Equation (17)) for the pixels inside the curve *k* in the image acquired by the camera *i*; λ and μ are positive and real constants to weigh the contribution of the terms in the objective function (18) and $\nabla = (\partial_x, \partial_y)$ is the spatial gradient operator.

As in the objective function for one camera (Equation (4)), the first term in (18) measures the conformity of 3D interpretation to the sequence spatiotemporal variations in each region through the 3D brightness constraint for a multi-camera sensor system. The second integral is a regularization term of smoothness of depth, and the third integral is a regularization term of the *N*-1 boundaries.

The objective function in Equation (18) includes information of all the cameras in the intelligent space. In this work, objective function minimization is carried out using a greedy algorithm that, after the initialization of the variables, consists of three iterative steps. Before the minimization, it is necessary to initialize the curves that define the contours of the segmentation and depth in the images acquired by each camera. Both, the initialization process and the minimization algorithm are explained below.

3.3. Curve and Depth Initialization

The initialization process is very important due to the high dependence of the results on the initial values of the variables. This process includes three different steps: in the first step, we obtain the initial curves. Since cameras are located in fixed positions within the intelligent space, the *N*-1 initial curves are obtained using GPCA (Generalized Principal Components Analysis) [9]. Then, the initial depth (relative to each camera coordinate system Γ_{ci}) is obtained using Visual Hull 3D [10] which allows to obtain a 3D occupancy grid (composed by cubes with size ⍰ h) in Γ_w from the initial segmentation boundaries, that have been computed previously using GPCA. Finally, an extended version of the k-means algorithm is used to estimate the number of mobile robots in the scene. The three steps are described below.

As previously mentioned, GPCA [9] is used in this work to obtain a background model for each of the n_c cameras. Background modeling is carried out from a set of background images that do not contain any mobile robot. Using GPCA we obtain two transformation matrices, $\mathbf{L}_{ci}$ and $\mathbf{R}_{ci}$, for each camera. These matrices are calculated in each camera, and they represent the background model. Since the cameras are placed in fixed positions within the environment, the background modeling stage needs to be carried out only once, and it can be done off-line.

GPCA [9] is also used to initialize the segmentation boundaries by comparing each image to the background model. In this stage, each image is projected (Equation (19)) to the GPCA space using the matrices **L** and **R** (that have been obtained previously). After that, the image is reconstructed (Equation (20)). In these two equations **M** represents the mean of the N_i images that have been used to obtain the background model:

$$\mathbf{I}_T = \mathbf{L}^T(\mathbf{I} - \mathbf{M})\mathbf{R} \tag{19}$$

$$\mathbf{I}_R = \mathbf{L}\mathbf{I}_T\mathbf{R}^T + \mathbf{M} \tag{20}$$

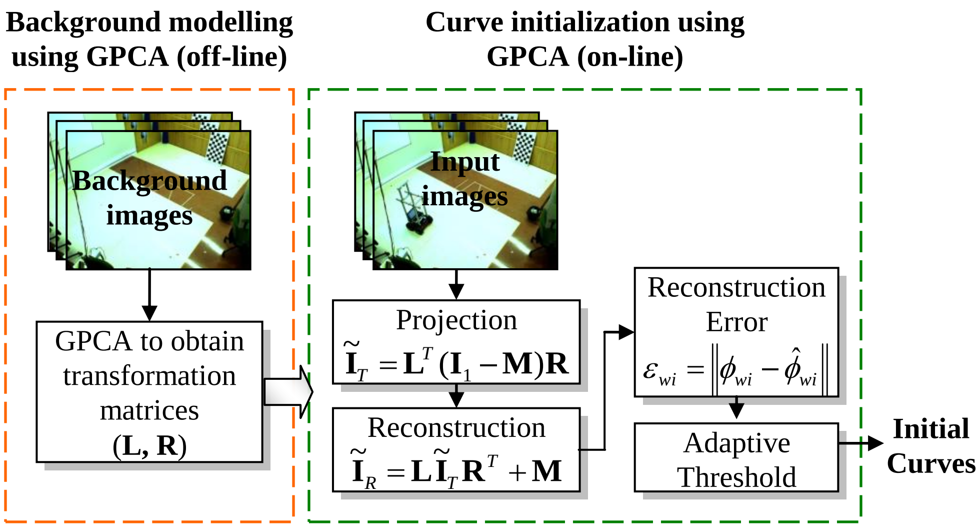
Then, the reconstruction error is computed. This error is defined as the difference between the reconstructed ($\mathbf{I}_R$) and the original (**I**) image and can be calculated subtracting the images pixel-to-pixel, but this approach is not robust against noise. Therefore, we define a set of pixels (window) around each pixel (with dimensions *q*x*q*) called $\mathbf{\Phi}_{wi}$ in the original image an $\hat{\mathbf{\Phi}}_{wi}$ in the reconstructed image, and we obtain the reconstruction error for these windows, using Equation (21):

$$\varepsilon_{wi} = \left\| \mathbf{\Phi}_{wi} - \hat{\mathbf{\Phi}}_{wi} \right\| \tag{21}$$

Pixels whose reconstruction error (calculated using Equation (21)) is higher than a threshold are candidate to belong to a mobile robot, because in those pixels there is an important difference between the current image and the background model. The value of the threshold is very important. In this work we use an adaptive threshold [11].

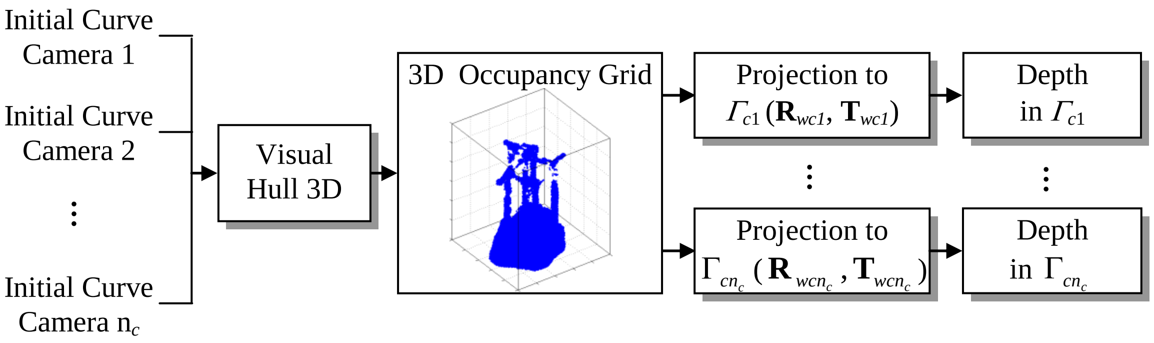
A block diagram including all the stages involved in curve initialization using GPCA is shown in Figure 5. All these stages have to be executed for each camera to obtain a set of initial curves $\{\gamma_{ki}\}_{k=1,\ldots,N-1}^{i=1,\ldots,n_c}$.

Figure 5. General block diagram of the proposed method for curve initialization using GPCA.


Background modelling using GPCA (off-line)
Curve initialization using GPCA (on-line)
Background images
GPCA to obtain transformation matrices (L, R)
Input images
Projection
$\tilde{\mathbf{I}}_T = \mathbf{L}^T(\mathbf{I}_1 - \mathbf{M})\mathbf{R}$
Reconstruction
$\tilde{\mathbf{I}}_R = \mathbf{L}\tilde{\mathbf{I}}_T\mathbf{R}^T + \mathbf{M}$
Reconstruction Error
$\varepsilon_{wi} = \left\| \phi_{wi} - \hat{\phi}_{wi} \right\|$
Adaptive Threshold
Initial Curves

After curve initialization, Visual Hull 3D [10] is used to obtain a 3D occupancy grid (composed of cubes of size Δh) in Γ_w from the initial segmentation boundaries computed previously. The 3D coordinates of the occupied cell are projected from Γ_w to each camera coordinate system Γ_{ci} (i=1,…,n_c) through the transformation matrices ($\mathbf{R}_{wci}$ and $\mathbf{T}_{wci}$) to obtain a set of points on the mobile robots in Γ_{ci}. This process provides an effective method for depth initialization in each camera. Figure 6 presents a block diagram including the main steps in the depth initialization process.

Figure 6. General block diagram of the proposed method for curve initialization using GPCA.


Initial Curve Camera 1
Initial Curve Camera 2
⋮
Initial Curve Camera n_c
Visual Hull 3D
3D Occupancy Grid
Projection to Γ_{c1} ($\mathbf{R}_{wc1}$, $\mathbf{T}_{wc1}$)
⋮
Projection to Γ_{cn_c} ($\mathbf{R}_{wcn_c}$, $\mathbf{T}_{wcn_c}$)
Depth in Γ_{c1}
⋮
Depth in Γ_{cn_c}

The algorithm used for motion segmentation and 3D positioning requires previous knowledge about the number of mobile robots. In order to estimate this value, we have included a clustering algorithm in the initialization process. In this stage, we project the coordinates of the occupied cell in the 3D occupancy grid obtained using Visual Hull 3D onto XY plane in Γ_w. Then, we cluster the 2D data using an extended version of k-means [12]. This clustering algorithm allows us to obtain a good estimation of the number of robots in the scene, and a division of the initial curves in each image.

The use of GPCA and VH3D allows obtaining a set of initial values of the variables that are close to the real ones. Using these initial values, the objective function minimization converges after a few iterations. It is noteworthy that the reduction in the number of iterations until convergence with respect to the algorithms in [7,8] decreases notably the processing time of the proposed solutions.

3.4. Objective Function Minimization

After curve and depth initialization, objective function minimization is carried out. Because the proposed objective function (defined in Equation (18)) depends on three groups of variables, a greedy algorithm, which consists of three iterated steps, is used. In each step, two of the three groups of variables are fixed, and we solve the equation for the remaining one.

In the first step, we fix segmentation boundaries and depth in each Γ_{ci}. So, the energy to minimize reduces to Equation (22):

$$E\left(\{\mathbf{v}_{wk}\}_{k=1}^{N},\{\boldsymbol{\omega}_{wk}\}_{k=1}^{N}\right)=\sum_{k=1}^{N}\sum_{i=1}^{n_{ck}}\int_{\Omega_{ki}}\psi_{ki}^{2}(\mathbf{x})d\mathbf{x} \tag{22}$$

Since the 3D brightness constraint for multiple cameras defined in Equation (17) depends linearly on the components of linear and angular velocity, 3D motion parameters in Γ_w can be obtained solving the linear equation system shown in Equation (23):

$$\begin{pmatrix} \mathbf{a}_{k1}(x_{11}) \\ \vdots \\ \mathbf{a}_{k1}(x_{p_{k1}1}) \\ \mathbf{a}_{k2}(x_{12}) \\ \vdots \\ \mathbf{a}_{kn_c}(x_{p_{kn_c n_c}}) \end{pmatrix} \begin{pmatrix} v_w^x \\ v_w^y \\ v_w^z \\ \omega_w^x \\ \omega_w^y \\ \omega_w^x \end{pmatrix} = \begin{pmatrix} -I_{t1}(x_{11}) \\ \vdots \\ -I_{t1}(x_{p_{k1}1}) \\ -I_{t2}(x_{12}) \\ \vdots \\ -I_{tn_c}(x_{p_{kn_c}n_c}) \end{pmatrix} \qquad k=1,\ldots,N \tag{23}$$

where: $\mathbf{a}_k(\mathbf{x}_j)=\left(\frac{S_{i1}}{Z_{ci}},\frac{S_{i2}}{Z_{ci}},\frac{S_{i3}}{Z_{ci}},\mathbf{Q}_{i1}+\frac{\mathbf{R}_{i1}}{Z_{ci}},\mathbf{Q}_{i2}+\frac{\mathbf{R}_{i2}}{Z_{ci}},\mathbf{Q}_{i3}+\frac{\mathbf{R}_{i3}}{Z_{ci}}\right)$.

In the second step, motion parameters and segmentation boundaries are fixed. In this step, the function to minimize is shown in Equation (24). In this function, χ_{ki} is the characteristic function of region k in image i (Ω_{ki}):

$$E(Z)=\int_{\Omega}\sum_{k=1}^{N}\sum_{i=1}^{n_{ck}}\left[\chi_{ki}(\mathbf{x})\left(\psi_{ki}^{2}(\mathbf{x})+\mu g(\|\nabla Z_{ci}\|)\right)\right]d\mathbf{x} \tag{24}$$

Given a set of contours $\{\gamma_{ki}\}_{k=1,\ldots,N-1}^{i=1,\ldots,n_c}$ that divides each image in N regions (N-1 mobile robots and background), Equation (25) shows the descend equations for any region and for any camera. In this equation, τ indicates the algorithm execution time and g' is the ordinary derivative of the boundary preserving function g. The boundary preserving function used in this work is a quadratic function ($g(a)=a^2$). This is a simple function, but its effectiveness has been verified in several experiments, in which we have compared the results obtained using this quadratic function, and other boundary functions described in [13]:

$$\frac{\partial Z_{ci}}{\partial \tau} = \frac{2}{Z_{ci}^2}\left(\mathbf{S}_i \mathbf{v}_{wk} + \mathbf{R}_i \boldsymbol{\omega}_{wk}\right)\psi_{ki} + \mu div\left(\frac{g'\left(\left\|\nabla Z_{ci}\right\|\right)}{\left\|\nabla Z_{ci}\right\|}\nabla Z_{ci}\right) \qquad \begin{matrix} i = 1,\ldots,n_{ck} \\ k = 1,\ldots,N \end{matrix} \tag{25}$$

The function to minimize in the third step is shown in Equation (27), where $\xi_{ki}(\mathbf{x}) = \psi_{ki}^2(\mathbf{x}) + \mu g\left(\left\|\nabla Z_{ci}\right\|\right)$. This function is obtained after fixing depth and 3D motion parameters:

$$E\left[\left\{\gamma_{ki}\right\}_{k=1,\ldots,N-1}^{i=1,\ldots,n_c}\right] = \sum_{k=1}^{N}\sum_{i=1}^{n_{ck}}\int_{\Omega_{ki}}\xi_k(\mathbf{x})d\mathbf{x} + \lambda\sum_{k=1}^{N-1}\sum_{i=1}^{n_{ck}}\oint_{\gamma_{ki}} ds \tag{26}$$

As described in [7], for multiple region segmentation, the Euler-Lagrange descent equations shown in Equation (27) are obtained:

$$\frac{\partial \gamma_{ki}}{\partial \tau}(\tau) = -\left(\xi_{ki}(\gamma_{ki}) - \varphi_{ki}(\gamma_{ki}) + \lambda\kappa_{\gamma_{ki}}(\gamma_{ki})\right)\times \mathbf{n}_{ki}(\gamma_{ki}) \qquad \begin{matrix} i = 1,\ldots,n_{ck} \\ k = 1,\ldots,N-1 \end{matrix} \tag{27}$$

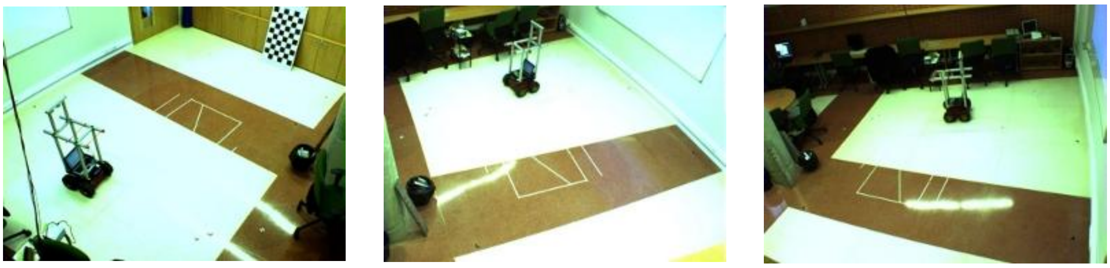
In these equations, $\kappa_{\gamma_{ki}}$ is the mean curvature and n_{ki} is the exterior, unit, normal function of the curve γ_{ki} . Functions φ_{ki} are defined as: $\varphi_{ki}(\gamma_{ki}(s)) = \min_{j \neq k} \xi_{ji}(\gamma_{ki}(s))$.

After initialization, the three described steps are repeated until the computed variables cease to evolve significantly.

4. Experimental Results

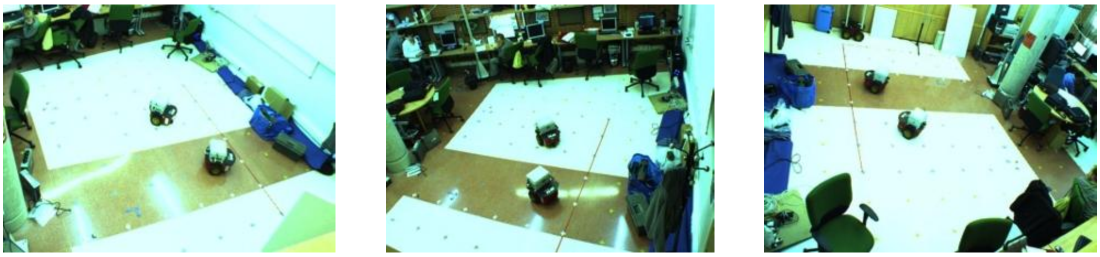
In order to validate the proposed system, several experiments have been carried out in the ISPACE-UAH. In these experiments we have used two five-hundred image sequences. These sequences have been acquired using three of the four cameras in the ISPACE-UAH. Figure 7 shows one scene belonging to each sequence. As can be noticed in Figure 7, sequence 1 contains one robot whereas sequence 2 contains two mobile robots. The proposed algorithm for motion segmentation and 3D localization using a multi-camera sensor system has been used to obtain motion segmentation and 3D position for each couple of images in each sequence. All the experiments shown in this work have been carried out on Intel® core 2, 6600 with 2.4 GHz using Matlab.

Figure 7. Images belonging to the test sequences, acquired by fixed cameras in the ISPACE-UAH. (a) Images belonging to the sequence 1 (b) Images belonging to the sequence 2.



(a)

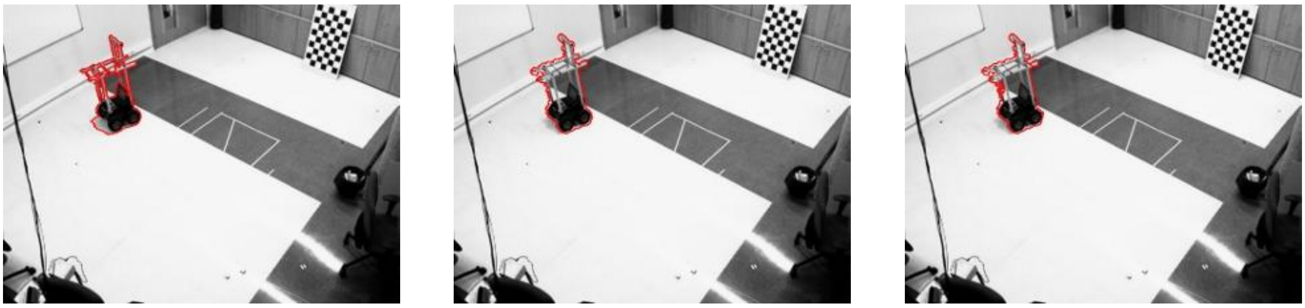
Figure 7. *Cont*.



(b)

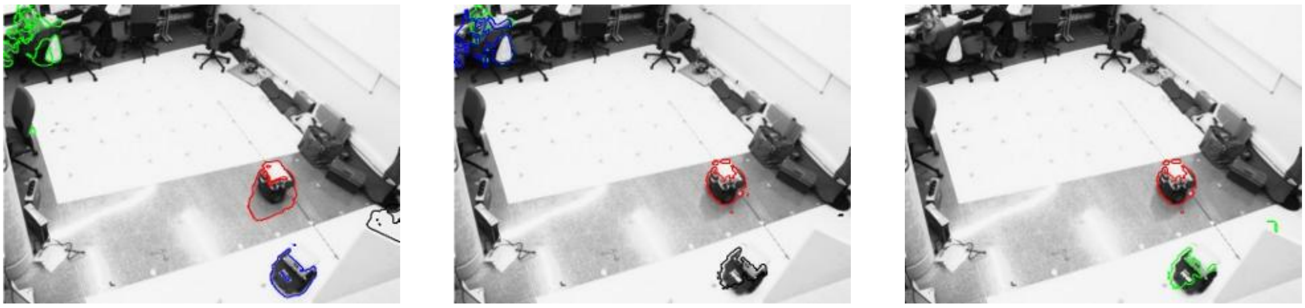
To start with the results, the boundaries of the motion segmentation in one image belonging to the sequence 1 [Figure 7(a)] and the sequence 2 [Figure 7(b)], respectively, are shown in Figure 8 and Figure 9 respectively.

Figure 8. Boundaries of the segmentation obtained after the objective function minimization for one image belonging to the sequence 1 (Figure 7(a)). (a) Curves obtained using one camera (b) Curves obtained using two cameras (c) Curves obtained using three cameras.



(a) (b) (c)

Figure 9. Boundaries of the segmentation obtained after the objective function minimization for one image belonging to the sequence 2 (Figure 7(b)). Each detected object is shown in a different colour (a) Curves obtained using one camera (b) Curves obtained using two cameras (c) Curves obtained using three cameras.



(a) (b) (c)

Figure 8 shows the boundaries of the segmentation obtained for one image belonging to the sequence 1 [Figure 7(a)] that contains only one robot. In this figure we can observe that the result of the motion segmentation is similar regardless of the number of cameras considered. In all the images, the segmentation boundary is close to the real contour of the mobile robot in the image plane.

However, the boundaries obtained for an image belonging to the sequence 2 [Figure 7(b)] are notably different for 1, 2 or 3 cameras, as can be noticed in Figure 9. If the segmentation is carried out from the images acquired using one [Figure 9(a)] or two [Figure 9(b)] cameras, the person in the background of the scene is considered as a mobile robot but, if the images from three cameras are used, this person is not detected.

The computational time depends on both, the number of cameras and the number of robots detected in the scene. If the number of detected robots remains constant (as in sequence 1, where only one robot is detected for 1, 2 or 3 cameras) the processing time increases with the number of cameras. It can be observed in Table 1, where the average value of the computation time of each couple of images in the image sequences 1 and 2 is shown. In Table 1 we can observe that, for the images belonging to the sequence 1 (with only 1 robot) computation time increases with the number of cameras.

On the other hand, the number of objects detected as mobile robots has a bigger impact in the computation time than the number of cameras, as can be noticed in Table 1. The sequence 1, used to obtain the results in Table 1, contains only one robot whereas the sequence 2 includes two robots. Comparing the results obtained for the sequence 1 and the sequence 2, it can be noticed that, regardless of the number of cameras, the processing time obtained for the sequence 2 (with two robots) is higher than the processing time obtained for the sequence 1 (including only one robot). Moreover, in case of the sequence 2, the computation time using two cameras is bigger than using three cameras. The reason is that the number of objects that have been segmented with 2 cameras is bigger.

Table 1. Average value of the computation time (in seconds) of each couple of the 500 images belonging to each test sequence for 1, 2 and 3 cameras.

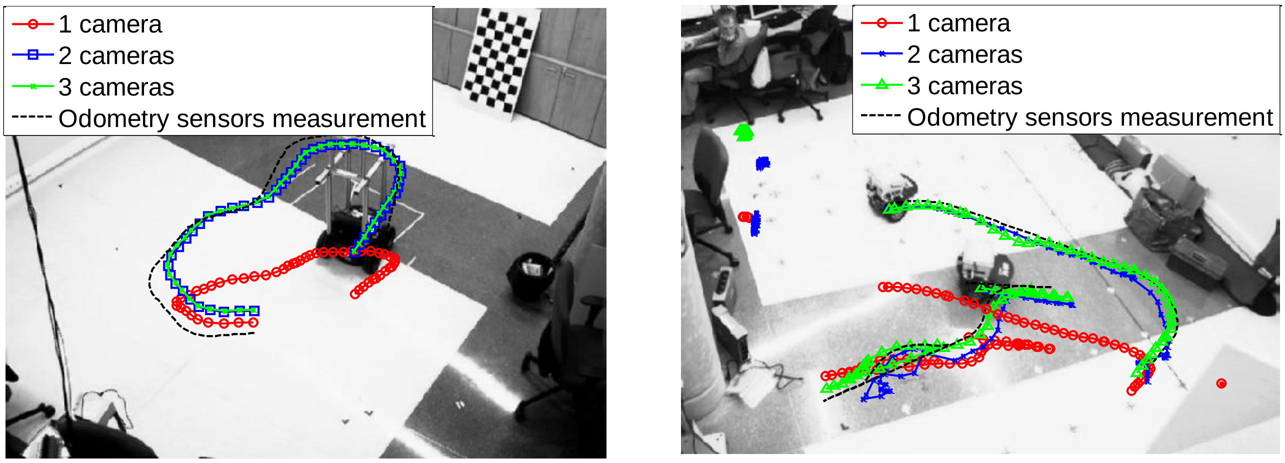
	Sequence 1 (contains one robot)			**Sequence 2 (contains two robots)**		
	1 camera	**2 cameras**	**3 cameras**	**1 camera**	**2 cameras**	**3 cameras**
Initialization	0.2910	2.8353	3.3234	0.3410	5.1390	4.0753
Minimization	0.8758	2.8247	4.1588	2.7273	9.8419	6.9925
Total	1.1668	5.6600	7.4822	3.0683	14.9810	11.0678

With regard to 3D positioning, Figure 10 shows the projection, onto the image plane, of the 3D trajectory of the mobile robot estimated by the algorithm (using 1, 2 and 3 cameras) and measured by the odometry sensors on board the robots. The represented trajectory has been calculated using 250 images belonging to each sequence.

The trajectories shown in Figure 10 are obtained by projecting the estimated trajectory in Γ_w, obtained using the proposed algorithm, onto the image plane of the camera 1.

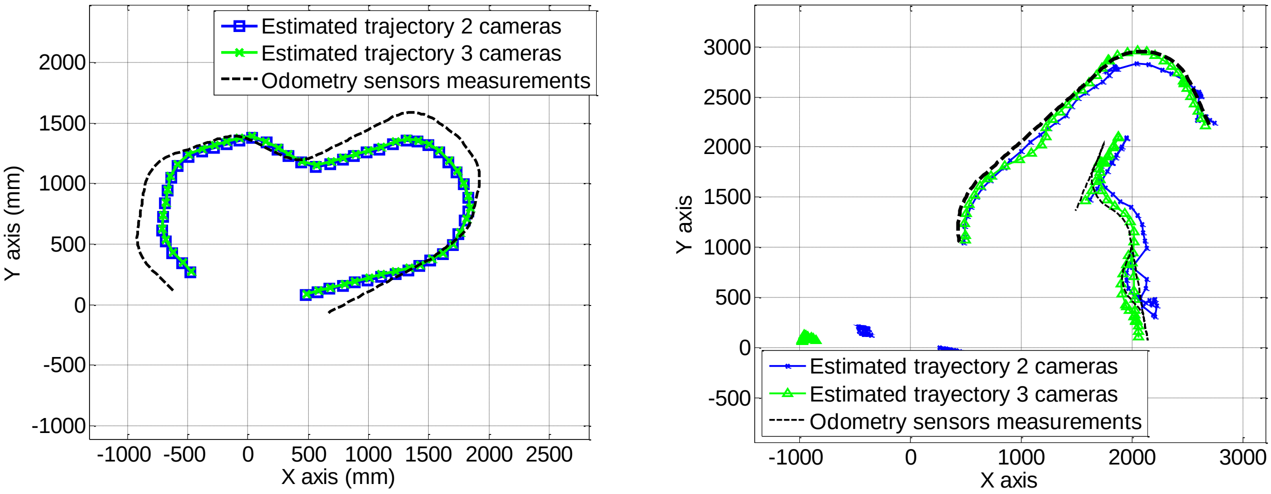
These trajectories obtained using 250 images belonging to each sequence can also be represented in the world coordinate system. The coordinates of the centroid of the points belonging to each robot are projected onto the plane (X_w, Y_w) in Γ_w to obtain the 3D position. The result of this projection for a 250 images belonging to each sequence is shown in Figure 11. In this figure, we have represented the estimated trajectory obtained using 2 and 3 cameras.

Figure 10. 3D trajectory estimated by the algorithm and measured by the odometry sensors on board the robots projected onto the image plane (a) Image belonging to the sequence 1 (b) Image belonging to the sequence 2.



(a) (b)

Figure 11. 3D trajectory estimated by the algorithm and measured by the odometry sensors on board the robots on the X_w, Y_w plane (a) Sequence 1 (b) Sequence 2.



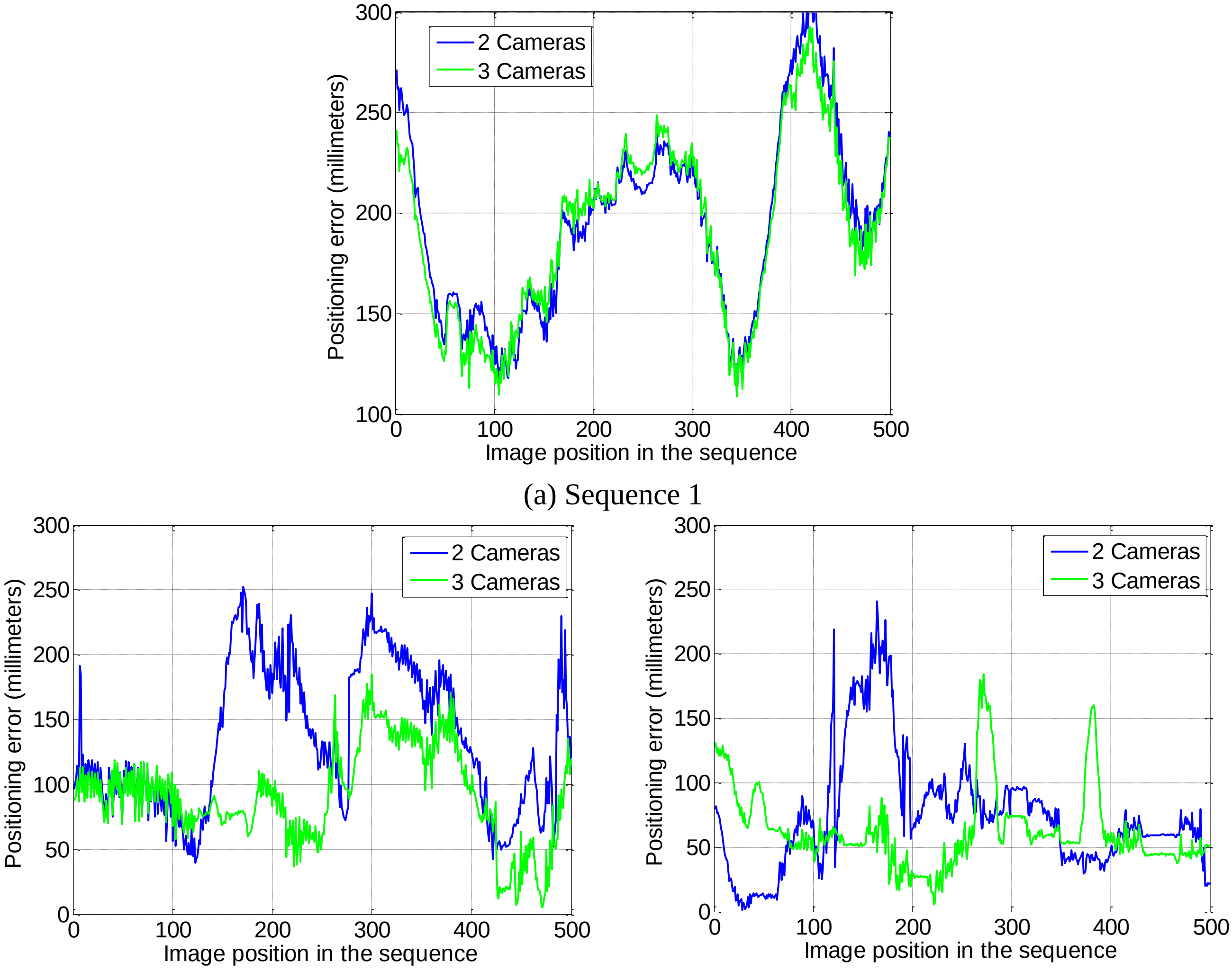
(a) (b)

As can be observed in Figure 10 and Figure 11, the estimated trajectories are closer to the measurements of the odometry sensors as the number of cameras increases. This fact can also be observed in the positioning error calculated as the difference between the estimated and the measured positions along X_w and Y_w axis, using Equation (28):

$$\varepsilon_p = \sqrt{\varepsilon_{px}^2 + \varepsilon_{py}^2} \qquad (28)$$

The positioning error, calculated for 500 images belonging to each sequence, has been represented in Figure 12. It is worth highlighting that the wheels of the robot in sequence 1 tend to skid on the floor. This is the reason why, in some positions, the difference between the estimated position and the measured one is high.

Figure 12. Positioning error (in millimetres) of the mobile robots, calculated using Equation (28) for 2 and 3 cameras. (a) Robot in the image sequence 1 (b) Robot 1 in the image sequence 2 (c) Robot 2 in the image sequence 2.



(a) Sequence 1

(b) Robot 1 in sequence 2

(c) Robot 2 in sequence 2

As can be observed in Figure 12, using the multi-camera sensor system with 2 or 3 cameras the positioning error is lower than 300 millimeters. It can also be observed in Table 2, where the average value of the positioning error for 500 images represented in

Figure 12 is shown. Moreover, the positioning error reduces as the number of cameras increases. This reduction is more important in the sequence 2. It is because sequence 2 is more complex than sequence 1 and the addition of more cameras allows removing the points that do not belong to mobile robots and dealing with robot occlusions.

Finally, it is noteworthy that although we have obtained better results using the images acquired by three cameras, two cameras are enough to obtain suitable 3D positions. For this reason, we can conclude that the proposal in this paper can work properly even if one of the three cameras looses track of one robot. Even in the worst case, if all the cameras lose some of the robots, they can still be controlled by the intelligent space. In this case, the positions of the unseen robots are estimated through the measurements of the odometry sensors they have onboard.

Table 2. Average value of the value of the positioning error (mm) obtained using 500 images belonging to each test sequence.

	Sequence 1	**Sequence 2**	
	Robot1	Robot 1	Robot 2
1 Camera	1001.5683	371.8227	769.7783
2 Cameras	194.8882	136.1451	75.3317
3 Cameras	191.7257	91.5264	63.2390

5. Conclusions

A method for obtaining the motion segmentation and 3D localization of multiple mobile robots in an intelligent space using a multi-camera sensor system has been presented. The set of calibrated and synchronized cameras are placed in fixed positions within the environment (in our case, the ISPACE-UAH). Motion segmentation and 3D position of the mobile robots are obtained through the minimization of an objective function that incorporates information from the multi-camera sensor. The proposed objective function has a high dependence on the initial values of the curves and depth. In this sense, the use of GPCA allows obtaining a set of curves that are close to the real contours of the mobile robots. Moreover, Visual Hull 3D allows us to relate the information from all the cameras, providing an effective method for depth initialization. The proposed initialization method guarantees that the minimization algorithm converges after a few iterations. The reduction in the number of iterations also decreases the processing time against other similar works.

Several experimental tests have been carried out in the ISPACE-UAH and the obtained results validate the proposal presented in this paper. It has been demonstrated that the use of a multi-camera sensor increases significantly the accuracy of the 3D localization of the mobile robots against the use of a single camera. It has also been proved that, the positioning error decreases as the number of cameras increases. In any case, using a multi-camera sensor, the positioning error is lower than 300 millimeters. With regard to the processing time, it depends on both, the number of cameras and the number of robots detected in the scene, having the second factor a bigger impact. In fact, the processing time can be reduced if the number of cameras is increased, because the noise measurements (that do not belong to mobile robots) are reduced when the number of cameras is increased.

Regarding to the future work, the most immediate task is the implementation of the whole system in real time. Currently the system is working in a small space (ISPACE-UAH). It will be extended, in order to cover a wider area, by adding more cameras to the environment and properly re-dimensioning the image processing hardware. This line of future work has a special interest towards its installation in buildings with multiple rooms.

Acknowledgements

The work described in this paper was supported by the Ministry of Science and Innovation (MICINN) under RESELAI project (REF-TIN2006-14896-C02-01).

References and Notes

1. Vázquez-Martín, R.; Núñez, P.; Bandera, A.; Sandoval, F. Curvature-Based Environment Description for Robot Navigation Using Laser Range Sensors. *Sensors* **2009**, *9*, 5894-5918.
2. Pizarro, D.; Mazo, M.; Santiso, E.; Marron, M.; Fernandez, I. Localization and Geometric Reconstruction of Mobile Robots Using a Camera Ring. *IEEE Trans. Instrum. Meas.* **2009**, *58*, N 8.
3. Lee, J.; Ando, N.; Yakushi, T.; Nakajima, K. Adaptative guidance for Mobile robots in intelligent infrastructure. *Proceedings of IEEE/RSJ International Conference on Robots and Systems*, Outrigger Wailea Resort, Maui, HI, USA, 2001; pp. 90-95.
4. Steinhaus, P.; Walther, M.; Giesler, B.; Dillmann, R. 3D global and Mobile sensor data fusion for Mobile platform navigation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, 2004; Volume 4, pp. 3325-3330.
5. Sogo, T.; Ishiguro, H.; Ishida, T. Acquisition of qualitative spatial representation by visual observation. *Proceedings of IJCAI*, Stockholm, Sweden, 1999; pp.1054-1060.
6. Fernandez, I.; Mazo, M; Lázaro, J.L.; Pizarro, D.; Santiso, E; Martín, P.; Losada, C. Guidance of a mobile robot using an array of static cameras located in the environment. *Auton. Robots.* **2007**, *23*, 305-324.
7. Sekkati, H.; Mitiche, A. Concurrent 3D Motion Segmentation and 3D Interpretation of Temporal Sequences of Monocular Images. *IEEE Trans. Image Proc.* **2006**, *15*, 641-653.
8. Sekkati, H.; Mitiche, A. Joint Optical Flow Estimation, Segmentation, and Interpretation with Level Sets. *Comput. Vis. Image Understand.* **2006**, *103*, 89-100.
9. Ye, J.P.; Janardan, R.; Li, Q. GPCA: an efficient dimension reduction scheme for image compression and retrieval. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, 2004; pp. 354-363.
10. Laurentini, A. The Visual Hull: a new tool for contour-based image understanding. *Proceedings of 7th Scandinavian Conference on Image Processing*, Aalborg, Denmark, 2001; pp. 993-1002.
11. Losada, C.; Mazo, M.; Palazuelos, S.; Redondo, F., Adaptive threshold for robust segmentation of mobile robots from visual information of their own movement. *Proceedings of the IEEE International Symposium on Intelligent Signal Processing*, Budapest, Hungary, 2009; pp.293-298.
12. Kanungo,T.; Mount, D.; Netanyahu, N.; Piatko, C.; Silverman, R.; Wu, A. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Patt. Anal. Mach. Int.* **2002**, *24*, 881-892.
13. Aubert, G.; Deriche, R.; Kornprobst, P. Computing optical flow via variational techniques. *SIAM J. Appl. Math.* **1999**, *60*, 156-182.