J Intell Robot Syst (2012) 66:125–149
DOI 10.1007/s10846-011-9601-5

Visual SLAM Based on Rigid-Body 3D Landmarks

Patricio Loncomilla · Javier Ruiz del Solar

Received: 17 December 2010 / Accepted: 11 May 2011 / Published online: 17 August 2011

Abstract In current visual SLAM methods, point-like landmarks (As in Filliat and Meyer (Cogn Syst Res 4(4):243–282, 2003), we use this expression to denote a landmark generated by a point or an object considered as punctual.) are used for representation on maps. As the observation of each point-like landmark gives only angular information about a bearing camera, a covariance matrix between point-like landmarks must be estimated in order to converge with a global scale estimation. However, as the computational complexity of covariance matrices scales in a quadratic way with the number of landmarks, the maximum number of landmarks that is possible to use is normally limited to a few hundred. In this paper, a visual SLAM system based on the use of what are called *rigid-body 3D landmarks* is proposed. A rigid-body 3D landmark represents the 6D pose of a rigid body in space (position and orientation), and its observation gives full-pose information about a bearing camera. Each rigid-body 3D landmark is created from a set of N point-like landmarks by collapsing 3*N* state components into seven state components plus a set of parameters that describe the shape of the landmark. Rigid-body 3D landmarks are represented and estimated using so-called *point-quaternions*, which are introduced here. By using rigid-body 3D landmarks, the computational time of an EKF-SLAM system can be reduced up to 5.5%, as the number of landmarks increases. The proposed visual SLAM system is validated in simulated and real video sequences (outdoor). The proposed methodology can be extended to any SLAM system based on the use of point-like landmarks, including those generated by laser measurement.

Keywords Robotics · Localization · SLAM · 6D SLAM · Visual SLAM · MonoSLAM · 3D Mapping · Model reduction

1 Introduction

Simultaneous Localization and Mapping (SLAM) has been one of the most highly investigated topics in mobile robotics in the last 20 years. Several workshops, special sessions in conferences and special issues in journals have been devoted to this research topic. Vision-based or visual SLAM, i.e. the attempt to solve SLAM using standard

P. Loncomilla · J. R. del Solar
Department of Electrical Engineering,
Universidad de Chile, Santiago, Chile

P. Loncomilla (✉) · J. R. del Solar
Advanced Mining Technology Center,
Universidad de Chile, Santiago, Chile
e-mail: ploncomi@ing.uchile

cameras as the main sensory input [2], has attracted the attention of the SLAM community in recent years. Main challenges in vision-based SLAM are robust feature detection, efficient and robust data association and loop-closure, and computationally efficient large-scale state estimation [2]. Visual-landmark definition, representation, and estimation are some of the key issues to tackle in order to address these challenges.

In the current vision-based SLAM literature, points are selected as landmarks because of their direct geometrical interpretation, which enables the straightforward formulation of the SLAM problem [3–5]. However, when a fully calibrated camera observes a point, only weak angular information that relates the observation to the poses of the landmarks and the camera is obtained. As only angular information is available, several possible maps can explain the observations, since any rotation, translation or scale transformations applied to them preserve the coherence between the model and the measurements [3].

As each point-like observation fulfills only 2 degrees of freedom, and the map has 7 degrees of freedom, sets of several simultaneously observed points must be used in order to estimate the map, which necessitates the use of a full-covariance matrix [6]. When the size of the map increases, the number of landmarks becomes very relevant, as the number of computations required to update the covariance matrix is proportional to the square of the full state size. As a result of this fact, the map size is limited by the number of landmarks, which can increase only up to a few hundred for real-time applications. Since a map created by using only angular information is weakly constrained, robustness and precision of local maps become very limited [3]. Landmark recognition is based on point-projection prediction and matching of local patches around each point, which gives weak association information, forcing the use of RANSAC-like strategies for discarding sets of false associations [4]. Then, alternative landmark-modeling methodologies are required in order to overcome the inherent limitations of point-like landmarks.

Landmarks, in their widest sense, are geometrical features that enable the description of a map in a fashion understandable to humans, and that make possible self-localization. By following this wide definition, it can be noted that humans localize themselves using landmarks that do not correspond to points, but instead correspond to wide regions in space that are recognized by visual inspection, by means of a hierarchy of increasingly sophisticated representations. Visual observations from these wide-region landmarks are not limited to angular information, since they include both relative distance and orientation between the observer and each landmark. In addition, humans are able to give descriptions of places or paths between different places by using references to semantic information that is more related to full objects than to points. Thus, the ability of a robot to use landmarks related to wide areas, instead of to points, is desirable in order to generate more robust observations, to facilitate semantic labeling, and to reduce the amount of data needed to maintain the map.

In order to address the previously mentioned aspects, a methodology for generating, representing and estimating *rigid-body 3D landmarks* is proposed. A rigid-body 3D landmark represents the 6D pose of a rigid body in space (position and orientation), and its observation gives full-pose information about the camera. Each rigid-body 3D landmark is created from a set of N point-like landmarks by collapsing $3N$ state components into seven state components plus a set of parameters that describe the shape of the landmark (so-called body points and their covariance matrices). Rigid-body 3D landmarks are represented and estimated using *point-quaternions*, which are here introduced and named.

A visual SLAM system that uses point-like and rigid-body 3D landmarks, based on the EKF-SLAM formulation, is also proposed. The use of rigid-body 3D landmarks permits reducing the computational time of the EKF-SLAM system up to 5.5%, as the number of landmarks increases. The proposed visual SLAM system is validated in simulated and real video sequences.

This paper is organized as follows. Important related work is presented in Section 2. In Section 3, the proposed methodology used to represent and estimate rigid-body 3D landmarks is described. In Section 4, the proposed visual SLAM system is explained. An experimental

evaluation of the system is presented in Section 5. Finally, some conclusions of this work are drawn in Section 6.

2 Related Work

Vision-based SLAM is an important research topic that has attracted increasing attention in the mobile robotics community. Interestingly, as smart-phones and digital cameras are gaining popularity, vision-based SLAM has acquired many potential applications beyond robotics, "due to the capability it can give a camera to serve as a general-purpose 3D position sensor" [2].

Most of the current work related to monocular-based visual localization stands on two approaches: structure-from-motion recovery, and monocular SLAM. Structure-from-motion recovery is based on algorithms that estimate corresponding points between consecutive images without using a dynamic model. Methodologies based on Nistér's visual odometry [7] using optimal preemptive RANSAC [8] applied over sets of three- and five-points, which are extracted by Harris filtering and processed using local bundle adjustment optimization, can achieve impressive results over large paths, but they accumulate an ever-increasing error over time, as they are based only on relative motion.

Monocular visual SLAM approaches, based on the seminal works of Davison [6, 9], can achieve impressive results in small to middle-size maps, but the management of large maps is a hard topic to face because of scale drifts, covariance matrix expansion, and loop-closure limitations. Live dense reconstruction [10] can be achieved by updating an active mesh by means of constrained optical-flow based minimization. Scalable active matching [5] has been proposed to manage large maps that involve a large amount of cross-correlation by using a graph-pruning approach in order to reduce covariance data, and to limit uncertainty propagation between distant points. A drift-aware monocular SLAM [3] has been proposed to model scale-drift by using a Lie group approach over the rotation-translation-scale transformation group, to achieve differential-constrained bundle-adjustment optimization for loop closing. As the time needed for RANSAC to solve a problem increases dramatically with the number of points needed for conforming a minimal subset, 1-point RANSAC [4] has been proposed to achieve fast data association. This approach is based on using 1 point to update the pose of the robot, and then using the new robot's pose for evaluating consensus on the other points using a chi-square test. Finally, appearance and 3D geometry [11] have been used to cluster a map into sets of points that are close in space, and that have similar image areas around them. This approach looks promising for building semantic models.

As has been already mentioned, some of the current problems of visual SLAM systems are derived from the fact that the perception of a point-like landmark does not allow the camera's pose to be inferred, and several points must be perceived and analyzed. To overcome this drawback, high-level landmarks based on sets of points scattered over the object surfaces can be used [12, 13]. In [12] high-level structures, such as planes and lines, are built online using a bottom-up process that first maps point-like and line-like landmarks, and then searches for sets of them that agree with the high level landmarks' hypothesis. In [13] locally planar landmarks represented using the inverse depth parametrization [14] are defined. The camera's state, landmark's normal-plane, and measurement errors are represented as Lie groups. Local reference frames defined by a central point and Euler-like angles have been used in 3D laser SLAM for representing local planar patches, which generate more compact and meaningfully maps [15–18]. However, the use of plane-based features limits the ability of the methods to handle general outdoor environments, and observations related to planes lose two degrees of freedom respect to full pose information, which limits the amount of information gathered from each observation. The approach proposed in this work is also based on the collapsing of point-like landmarks into high-level landmarks, but the main differences are the use of non-planar 3D landmarks, which adds flexibility to the system, and the definition of a methodology for landmark representation and estimation that is based on the use of point-quaternions, which form a

rotationally-symmetric algebraic group representation for poses in space.

3 Rigid-Body 3D Landmark Representation and Estimation

A rigid-body 3D landmark, from now on referred to as a 3D landmark, represents the 6D pose of a rigid body in space. A rigid body is composed by a set of observable points called *body points*, which are used to create a 3D landmark. The pose of the 3D landmark is determined by the location of the rigid body points when referred to a global reference frame. The pose of the 3D landmark is encoded by using a point and a quaternion [19] chained into a unique object named a *point-quaternion*. The covariance of a 3D landmark's pose is determined by the covariance of its associated point-quaternion.

In a SLAM system, every time a subset of the body points is observed, a compatible pose for the 3D landmark is computed and used as a virtual observation. Uncertainty related to the observation of the body points can be propagated into uncertainty in the pose of the 3D landmark. The virtual observation and its covariance enable the correction of the 3D landmark pose estimation.

3.1 6D pose representation using point-quaternions

A point-quaternion η is introduced in this paper as a 7D mathematical object that is composed by a point t and a quaternion q. The point is used to denote a position, and the quaternion is used to denote an orientation. In this way, a point-quaternion can represent a 6D pose in space. A quaternion can be formed by specifying a unitary rotation axis ω, and a rotation angle θ. Then, η is defined as:

$$\eta_{7\times1} = \begin{pmatrix} t_{3\times1} \\ q_{4\times1} \end{pmatrix}; q = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ \omega_X \sin(\theta/2) \\ \omega_Y \sin(\theta/2) \\ \omega_Z \sin(\theta/2) \end{pmatrix}; \quad t = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

Similarly as in the case of using transformation matrices, using point-quaternions allows defining a transformation operation, *transop*, over a vector p, consisting of a rotation followed by a translation:

$$transop\,(\eta, p) = q \cdot p \cdot q^{-1} + t \tag{2}$$

The inverse transformation, *inv_transop*, is defined as:

$$inv\_transop\,(\eta, p) = q^{-1} \cdot (p - t) \cdot q \tag{3}$$

Point-quaternions can be composed by using a multiplication operation, which is defined as:

$$\eta_1 \cdot \eta_2 = \begin{pmatrix} t_1 \\ q_1 \end{pmatrix} \cdot \begin{pmatrix} t_2 \\ q_2 \end{pmatrix} = \begin{pmatrix} q_1 \cdot t_2 \cdot q_1^{-1} + t_1 \\ q_1 \cdot q_2 \end{pmatrix} \tag{4}$$

Point-quaternions containing a zero quaternion are ill-posed as they do not represent any rotation. Valid point-quaternions and their multiplication form a group as they have closure, associativity, an identity element (η_I) and an inverse element (see proof in [20]):

$$\eta_I = \begin{pmatrix} 0_{3\times1} \\ 1 \end{pmatrix} = \begin{pmatrix} 0, 0, 0^T \\ 1, 0, 0, 0^T \end{pmatrix}, \quad \eta^{-1} = \begin{pmatrix} -q^{-1} \cdot t \cdot q \\ q^{-1} \end{pmatrix} \tag{5}$$

A special sum for point-quaternions is not defined because of the lack of distributive properties, but vector summation can be applied for Jacobian-calculation purposes [20].

Point-quaternion multiplication can be used to relate different reference systems as transformation matrices do. Coordinates from points on a reference system A can be transformed into coordinates on a reference system B by using a point-quaternion η_{AB}. Coordinate transformations between reference systems A, B and C can be composed by using point-quaternion multiplication (the multiplication direction is the same as that used in homogeneous matrix composition):

$$\eta_{AC} = \eta_{BC} \cdot \eta_{AB} \tag{6}$$

Each point-quaternion η has an associated homogeneous matrix $\mathbf{H}(\eta)$ that represents the same transformation:

$$\mathbf{H}(\eta) = \mathbf{H}\left(\begin{pmatrix} t \\ q \end{pmatrix}\right) = \begin{pmatrix} \frac{a^2+b^2-c^2-d^2}{a^2+b^2+c^2+d^2} & \frac{2bc-ad}{a^2+b^2+c^2+d^2} & \frac{2bd+2ac}{a^2+b^2+c^2+d^2} & t_X \\ \frac{2bc+2ad}{a^2+b^2+c^2+d^2} & \frac{a^2-b^2+c^2-d^2}{a^2+b^2+c^2+d^2} & \frac{2cd-2ab}{a^2+b^2+c^2+d^2} & t_Y \\ \frac{2bd-2ac}{a^2+b^2+c^2+d^2} & \frac{2cd+2ab}{a^2+b^2+c^2+d^2} & \frac{a^2-b^2-c^2+d^2}{a^2+b^2+c^2+d^2} & t_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Thus, point-quaternions can be used for the same purposes as homogeneous matrices, being more compact and well-posed as they do not include distorting effects associated with homogeneous matrices. As each homogeneous matrix contains 12 variable components, the error covariance representation associated with a homogeneous matrix uses 12×12 components, and it is ill-posed when representing pose uncertainty as it can encode uncertainty about axis orthogonality and scaling. Conversely, the covariance matrix of a point-quaternion, a 7×7 symmetric semipositive-definite matrix, encodes the uncertainty about a pose in space, and it is well posed as it always represents pure pose uncertainties.

3.2 Rigid-body 3D Landmark Generation Procedure

The procedure used for creating a rigid-body 3D landmark from N individual point-like landmarks (points) involves transforming $3N$ position state components into seven-pose state components. The covariance representation must be transformed at the same time.

First, the SLAM state vector x (see Section 4) is divided into the set of points p_{SET} to be fused, and the other state components o:

$$x = \begin{pmatrix} p_{SET} \\ o \end{pmatrix}; p_{SET} = \begin{pmatrix} p_1 \\ \dots \\ p_N \end{pmatrix}; p_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \tag{8}$$

Body points Π_i are computed by subtracting the mean position value from each point p_i to be fused:

$$\Pi_i = p_i - m, i = 1, \dots, N; \qquad m\frac{1}{N}\sum_{i=1}^{N} p_i \tag{9}$$

In general terms, a point-quaternion defining a coordinate transformation T that relates a set of points in a reference frame A to a set of points in a reference frame B, can be computed with minimal error as:

$$T\left(p_1^{(B)}, \dots, p_N^{(B)}, p_1^{(A)}, \dots, p_N^{(A)}\right) = \arg\min_{\eta_{LN}} \left(\sum_{i=1}^{N} \left\| q_{LN} \cdot p_i^{(A)} \cdot q_{LN}^{-1} + t_{LN} - p_i^{(B)} \right\|^2\right) \tag{10}$$

Given that the transformation that relates the body points to the original points corresponds to a translation m and an identity rotation:

$$\eta_{LAND-MAP} = \begin{pmatrix} t_{LAND-MAP} \\ q_{LAND-MAP} \end{pmatrix} = T(p_1, \dots, p_N, \Pi_1, \dots, \Pi_N) = \begin{pmatrix} m \\ 1 \end{pmatrix} \tag{11}$$

The new state representation of the 3D landmark is put into the full state vector:

$$x_{NEW} = \begin{pmatrix} \eta_{LAND-MAP} \\ o \end{pmatrix} \tag{12}$$

The covariance matrix of the state P is divided into four sub matrices, where P_{pp} contains the covariances from the points to be fused:

$$P = \begin{pmatrix} P_{pp} & P_{po} \\ P_{op} & P_{oo} \end{pmatrix}, \quad P_{pp} = \begin{pmatrix} P_{3\times3}^{(1,1)} & P_{3\times3}^{(1,2)} & \dots & P_{3\times3}^{(1,N)} \\ P_{3\times3}^{(2,1)} & P_{3\times3}^{(2,2)} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ P_{3\times3}^{(N,1)} & \dots & \dots & P_{3\times3}^{(N,N)} \end{pmatrix} \tag{13}$$

Considering the size reduction of the vector state, P will adopt the following form:

$$P = \begin{pmatrix} P_{77} & P_{7o} \\ P_{o7} & P_{oo} \end{pmatrix} \tag{14}$$

By considering that each body point Π_i will have an associated covariance $P_{\Pi}^{(i)}$, covariance propagation from p_i and Π_i into $\eta_{LAND-MAP}$ can be estimated using a first order Taylor expansion:

$$P_{77} = J_p P_{pp} J_p^T - J_\Pi \begin{pmatrix} P_{\Pi}^{(1)} & & \\ & \cdots & \\ & & P_{\Pi}^{(N)} \end{pmatrix} J_\Pi^T \tag{15}$$

with

$$J_p = \frac{\partial T\,(p_1, \ldots, p_N, \Pi_1, \ldots, \Pi_N)}{\partial\,(p_1, \ldots, p_N)};$$

$$J_\Pi = \frac{\partial T\,(p_1, \ldots, p_N, \Pi_1, \ldots, \Pi_N)}{\partial\,(\Pi_1, \ldots, \Pi_N)} = -J_p \tag{16}$$

P_{7o} and P_{o7} are updated as:

$$P_{7o} = J_p P_{po};\ P_{o7} = P_{op} J_p^T \tag{17}$$

The error associated to P_{pp} must be divided into the P_{77} pose covariance and the $P_{\Pi}^{(i)}$ body-points covariance. The decomposition is not unique as any choice for the set of covariances $P_{\Pi}^{(i)}$ and P_{77} is valid, when all of the involved covariance matrices are positive semidefinite. Then, several criteria for selecting the $P_{\Pi}^{(i)}$ can be generated. In this work, two criteria will be considered, maximal pose covariance and maximal body-points covariance.

1. *Maximal pose covariance*. The procedure considers the following steps. First, transfer all the covariance error associated to P_{pp} into P_{77}:

   $$P_{77} = J_p P_{pp} J_p^T \tag{18}$$

   Then, compute the covariance matrix P_{REC}, which corresponds to an approximation of P_{77}:

   $$P_{REC} = G P_{77} G^T \tag{19}$$

   with

   $$G = \frac{\partial U\,(\Pi_1, \ldots, \Pi_N, \eta_{LAND-MAP})}{\partial\,(\Pi_1, \ldots, \Pi_N)} \tag{20}$$

   and

$$U\,(\Pi_1, \ldots, \Pi_N, \eta_{LAND-MAP}) = \begin{pmatrix} q_{LAND-MAP} \cdot \Pi_1 \cdot q_{LAND-MAP}^{-1} + t_{LAND-MAP} \\ \cdots \\ q_{LAND-MAP} \cdot \Pi_N \cdot q_{LAND-MAP}^{-1} + t_{LAND-MAP} \end{pmatrix} \approx \begin{pmatrix} p_1 \\ \ldots \\ p_N \end{pmatrix} \tag{21}$$

   Afterwards, calculate the covariance matrix of each body point $P_{\Pi}^{(i)}$ by subtracting the original P_{pp} and its approximation P_{REC} as:

   $$P_{\Pi}^{(i)} = D_{3\times3}^{(i,i)} \tag{22}$$

   $$P_{DIFF} = P_{pp} - P_{REC} = \begin{pmatrix} D_{3\times3}^{(1,1)} & D_{3\times3}^{(1,2)} & \ldots & D_{3\times3}^{(1,N)} \\ D_{3\times3}^{(2,1)} & D_{3\times3}^{(2,2)} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ D_{3\times3}^{(N,1)} & \ldots & \ldots & D_{3\times3}^{(N,N)} \end{pmatrix} \tag{23}$$

   Finally, each $P_{\Pi}^{(i)}$ must be checked for positive semidefiniteness by making zero its negative eigenvalues and reconstructing the matrix.

2. *Maximal body-points covariance*. Transfer the maximal amount of covariance to the set of body points $P_{\Pi}^{(i)}$ by minimizing the amount of covariance that is transferred to P_{77} (the Levenberg-Marquardt optimization procedure is used):

   $$\min_{\alpha_1, \alpha_2} \left\{ \lambda_{LOWER} \left(P_{pp} - \alpha_1 \begin{pmatrix} P_{3x3}^{(1,1)} & & \\ & \cdots & \\ & & P_{3x3}^{(N,N)} \end{pmatrix} - \alpha_2 D \right) \right\}^2 \tag{24}$$

   with

   $$D = diag\,(P_{pp}) \tag{25}$$

   and $\lambda_{LOWER}(M)$ the lower eigenvalue of a certain matrix M.

As any covariance matrix must be positive semidefinite, it is minimal when its lower eigenvalue is near to zero. After α_1 and α_2 are determined, the $P_{\Pi}^{(i)}$ are updated as:

$$\begin{pmatrix} P_{\Pi}^{(i)} & & \\ & \cdots & \\ & & P_{\Pi}^{(N)} \end{pmatrix} = \alpha_1 \begin{pmatrix} P_{3\times3}^{(1,1)} & & \\ & \cdots & \\ & & P_{3\times3}^{(N,N)} \end{pmatrix} + \alpha_2 D \tag{26}$$

Then, P_{77} is computed using (15).

The body points and their covariance matrices are stored in a special data structure, which does not need to be updated by the SLAM update procedure.

3.3 Rigid-body 3D Landmark Generation Criterion

The decision for generating a new rigid-body 3D landmark depends on the covariance of the points to be fused. Small covariances indicate smaller errors in the observations. Positive and similar cross-covariances between the points assure that the correction of one point generates a similar correction in all the other points, and then they behave as a rigid body.

The proposed fusion criterion is fast to compute and enables the creation of sets of landmarks that are candidates for fusing. It is based on the analysis of the covariance matrix of the points to be fused. A variability index (*varIndex*) is computed. It indicates the degree of variation of the components from a subset of the covariance matrix. Subsets with low variability indicate that cross covariances are similar.

Before computing the variability indices of the covariance matrix P_{pp}, their diagonal components $P_i = P_{3\times3}^{(i,i)}$ (see Eq. 13) are ordered by decreasing trace-value of the covariance sub matrices for each point:

$$\begin{aligned} P_i > P_j \Rightarrow\ & P_{iXX} + P_{iYY} + P_{iZZ} \\ & > P_{jXX} + P_{jYY} + P_{jZZ} \end{aligned} \tag{27}$$

with

$$P_i = \begin{pmatrix} P_{iXX} & P_{iXY} & P_{iXZ} \\ P_{iYX} & P_{iYY} & P_{iYZ} \\ P_{iZX} & P_{iZY} & P_{iZZ} \end{pmatrix} \tag{28}$$

The ordering indicated in (27) can be altered for eliminating terms that have several negative cross-covariances over the X, Y or Z components (see details in [20]). The variability index is computed on several windows in the covariance matrix using summed area tables for computing fast average values into the window:

$$\begin{aligned} varIndex = \min_{q,r}\ & C_X(q, r, q+M, r+M) \\ & + C_Y(q, r, q+M, r+M) \\ & + C_Z(q, r, q+M, r+M) \end{aligned} \tag{29}$$

with

$$C_X(q_0, r_0, q_1, r_1) = \frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrXX}^2}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \times \left(\frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrXX}}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \right)^2 \tag{30}$$

$$C_Y(q_0, r_0, q_1, r_1) = \frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrYY}^2}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \times \left(\frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrYY}}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \right)^2 \tag{31}$$

$$C_Z(q_0, r_0, q_1, r_1) = \frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrZZ}^2}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \times \left(\frac{\sum_{q=q_0}^{q_1}\sum_{r=r_0}^{r_1} P_{qrZZ}}{(q_1 - q_0 + 1)(r_1 - r_0 + 1)} \right)^2 \tag{32}$$

and M the number of points to be grouped (e.g. $M = 10$).

Finally, when a window has a *varIndex* below a threshold *th*, the selected points are collapsed into a 3D landmark using the procedure described in Section 3.2.

3.4 Virtual and Estimated 3D Observations

Every time the rigid body represented by the rigid-body 3D landmark is observed, measurements involving the body points are obtained. In this work, body points are detected as points of interest using the SURF methodology [21]. The position of each interest point $(pos_x,\ pos_y)$ is translated into normalized pixel coordinates, and defines a basic observation z_{uv} to be used by the SLAM system:

$$z_{uv} = \begin{pmatrix} u_x \\ v_y \end{pmatrix} = \begin{pmatrix} pos_x/distFoc_x \\ pos_y/distFoc_y \end{pmatrix} \tag{33}$$

with $distFoc_X/distFoc_Y$ the focal distances in x and y, respectively.

After data association (see description in Section 4.2), the set of measurements in normalized coordinates $\{z_{uv}\}$ can be transformed into a virtual observation of a rigid-body 3D landmark z_{rb3D}. This requires minimizing a measurement error that relates the coordinates of the body points, the pose of the corresponding rigid-body 3D landmark (whose identity is determined in the data association process), and the real measured observations. As the virtual observation computation involves minimizing an error, an initial pose must be provided for the minimization algorithm. The initial pose is estimated by using the three-point algorithm [22] in several random-selected triplets of measured interest points. The three-point algorithm (*alg3p* function) enables the calculation of the positions of three points in space when the projected points and the distances between the points in space are known. As up to four solutions can be obtained, a fourth point is needed for disambiguation. Twelve sets of four points are used to generate a set of candidate poses. The last detected pose is also added to this set. The candidate pose with the lowest error is selected. By using this procedure, an initial pose η_0 that projects a triplet of body points into three measured interest points on the image with low error is obtained:

$$\eta_0 = \arg\min_{\eta_{abcd}\in \mathrm{I}} \left(E_P\left(\eta_{abcd}\right)\right);\ \mathrm{I} = \{\eta_1, , , \eta_{13}\} \tag{34}$$

with

$$\eta_{abcd} = \mathrm{alg3p}\left(\Pi_a, \Pi_b, \Pi_c, \Pi_d, \begin{pmatrix} u_a \\ v_a \end{pmatrix}, \begin{pmatrix} u_b \\ v_b \end{pmatrix}, \begin{pmatrix} u_c \\ v_c \end{pmatrix}, \begin{pmatrix} u_d \\ v_d \end{pmatrix}\right) a \neq b \neq c \neq d \tag{35}$$

and

$$E_P\left(\eta_{LC}\right) = \sum_{i=1}^{N} \left\| projection\left(q_{LC}\cdot \Pi_i \cdot q_{LC}^{-1} + t_{LC}\right) - \begin{pmatrix} u_i \\ v_i \end{pmatrix} \right\| \tag{36}$$

The projection operation maps points in space into the image space:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x/z \\ y/z \end{pmatrix} = projection \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{37}$$

Then, the virtual observation z_{rb3D} is computed by iterative optimization using Levenberg-Marquardt, using as the initial solution η_0:

$$\begin{aligned} z_{rb3D} &= V\left(u_1, v_1, \ldots, u_N, v_N, \Pi_1, \ldots, \Pi_N\right) \\ &= \eta_{LAND-CAM-MEASURED} \\ &= \arg\min_{\eta_i} E_P\left(\eta_i\right) \end{aligned} \tag{38}$$

The error covariance matrix associated with the virtual observational process, R_{rb3D} is computed by propagating the errors associated with the observations R_{UV} and the error associated with the body points P_Π:

$$R_{rb3D} = \sum_{i=1}^{N} J_{UV}^{(i)} \cdot R_{UV} \cdot \left(J_{UV}^{(i)}\right)^T + J_\Pi^{(i)} \cdot P_\Pi^{(i)} \cdot \left(J_\Pi^{(i)}\right)^T \tag{39}$$

with

$$R_{UV}\begin{pmatrix}\alpha^2_{pixelX}/distFoc^2_X & 0\\ 0 & \alpha^2_{pixelY}/distFoc^2_Y\end{pmatrix} \quad (40)$$

$$J^{(i)}_{UV} = \frac{\partial V(u_1, v_1, \ldots, u_N, v_N, \Pi_1, \ldots, \Pi_N)}{\partial (u_i, v_i)} \quad (41)$$

$$J^{(i)}_{\Pi} = \frac{\partial V(u_1, v_1, \ldots, u_N, v_N, \Pi_1, \ldots, \Pi_N)}{\partial \Pi_i} \quad (42)$$

The procedure used to compute these Jacobians is detailed in the Appendix.

An observation function h_{rb3D} allows computing an estimated pose for the rigid-body 3D landmark. Since the observation function depends on the camera pose and the 3D landmark pose, it depends on the representation used for the camera state. In this work, a point-quaternion $\eta_{CAM-MAP}$ encodes the pose of the camera in respect to the global reference frame (see Section 4.1). Considering that the pose of the 3D landmark is encoded by $\eta_{LAND-MAP}$, h_{rb3D} is given by:

$$\begin{aligned} h_{rb3D}(x) &= \eta_{LAND-CAM-EXPECTED} \\ &= \eta^{-1}_{CAM-MAP} \cdot \eta_{LAND-MAP} \end{aligned} \quad (43)$$

When few body points are observed, a virtual observation z_{rb3D} cannot be computed, but the body points observations can be used in the SLAM procedure (see Section 4.3). The observation function associated with each body point $h^{(i)}_{bp}$ is given by:

$$\begin{aligned} h^{(i)}_{bp}(x) = projection(&\eta_{LAND-CAM-EXPECTED} \cdot \Pi_i \\ &\times \eta^{-1}_{LAND-CAM-EXPECTED} \\ &+ t_{LAND-CAM-EXPECTED}) \end{aligned} \quad (44)$$

And the associated covariance computed as:

$$R^{(i)}_{bp} = R_{UV} + \frac{dh^{(i)}_{bp}(x)}{d\Pi_i} P^{(i)}_{\Pi} \left(\frac{dh^{(i)}_{bp}(x)}{d\Pi_i}\right)^T \quad (45)$$

3.5 Quaternion Sign Compatibility

For each possible pose, infinity point-quaternions can be selected as a possible representation. When a unitary quaternion constraint is imposed, there are two possible options: (t, q) and $(t, -q)$. Because the virtual observation z_{rb3D} and the estimated observation h_{rb3D} are computed in an independent way, they may lack compatible signs. For this reason a procedure for correcting this problem is required. The procedure is based on computing a cosine distance between the point-quaternions $\eta_{LAND-CAM-MEASURED}$ and $\eta_{LAND-CAM-EXPECTED}$. If they have different signs, the distance becomes negative and both the observation and its covariance are corrected:

$$\begin{aligned} &\langle q_{LAND-CAM-MEASURED}, q_{LAND-CAM-EXPECTED}\rangle < 0 \\ &\Rightarrow \begin{cases} q_{LAND-CAM-MEASURED} = -q_{LAND-CAM-MEASURED} \\ R_{rb3Dtq} = -R_{rb3Dtq} \\ R_{rb3Dqt} = -R_{rb3Dqt} \end{cases} \end{aligned} \quad (46)$$

with

$$\langle q_1, q_2\rangle = a_1a_2 + b_1b_2 + c_1c_2 + d_1d_2 \quad (47)$$

and R_{rb3Dtq} and R_{rb3Dqt} the covariance elements that are associated with the point-quaternion's components t and q.

3.6 Computational Complexity

Rigid-body 3D landmarks are generated by transforming N point-like landmarks into a 7D pose plus shape parameters. If the original state has $n_O + 3N$ components before fusion, it will remain with only $n_O + 7$ components after the transformation. For illustrating the speed gain caused by state reduction, two opposing cases will be analyzed.

In the first case the state of the system contains a camera state and D rigid-body 3D landmarks. Given that the camera state, composed by a point-quaternion, a linear velocity vector and an angular velocity vector (see Section 4.1), has 13 dimensions, the covariance matrix size is:

$$size_1(D) = (13 + 7D)^2 \quad (48)$$

In the second case the state of the system contains a camera state and D^*n_P point-like landmarks, with n_P being the number of points that are

required to form a 3D landmark. Then, the covariance matrix size is:

$$size_2\left(D, n_p\right) = \left(13 + 3Dn_p\right)^2 \tag{49}$$

As the number of landmarks increases, size differences become more significant:

$$\frac{size_1\left(D\right)}{size_2\left(D, n_P\right)} = \frac{\left(13 + 7D\right)^2}{\left(13 + 3D\,n_p\right)^2} = \frac{49D^2 + O\left(D\right)}{9^2 D^2 n_p^2 + O\left(D\right)} \approx \frac{5,\bar{4}}{n_p^2} \tag{50}$$

Then, in case all point-like landmarks are grouped into 3D landmarks, using 10 point-like landmarks to form each 3D landmark ($n_p = 10$), the state covariance matrix size can be reduced up to 5.5% as the number of landmarks increases. It is well known that the computing time needed in each iteration of the EKF-SLAM is limited by the computations required in the correction step, when the number of landmarks is large. As this time is proportional to the size of the state covariance matrix, computational time can be reduced up to 5.5%. Thus, the use of 3D landmarks is especially well-suited for large maps.

4 Visual SLAM System Using Rigid-Body 3D Landmarks

The proposed visual SLAM system is based on MonoSLAM [6], but it incorporates the simultaneous use of point-like and rigid-body 3D landmarks. EKF-SLAM is used as the basis algorithm for implementing the SLAM system. In a first stage point-like landmarks are stored using the inverse depth parametrization [14], then as standard 3D points.

4.1 State Representation

The state of the system x incorporates information about the camera state, and the poses of point-like, inverse-depth, and rigid-body 3D landmarks. The camera state x_{CAMERA} includes the camera pose, represented by using a point-quaternion $\eta_{CAM-MAP}$, and linear and angular velocity vectors, $v_{CAM-MAP}$ and $\omega_{CAM-MAP}$, respectively:

$$x_{CAMERA} = \begin{pmatrix} \eta_{CAM-MAP} \\ v_{CAM-MAP} \\ \omega_{CAM-MAP} \end{pmatrix} \tag{51}$$

The state update equation for the camera is given by (assuming a zero-mean Gaussian noise added to both velocities):

$$\mathrm{f}_{camera} = \begin{pmatrix} t_{cam-map(k+1)} \\ q_{cam-map(k+1)} \\ v_{cam-map(k+1)} \\ \omega_{cam-map(k+1)} \end{pmatrix} = \begin{pmatrix} t_{cam-map(k)} + \left(v_{cam-map(k)} + n_{V(k)}\right)\Delta t \\ quat\left(\left(\omega_{cam-map(k)} + n_{W(k)}\right)\Delta t\right) q_{cam-map(k)} \\ v_{cam-map(k)} + n_{V(k)} \\ \omega_{cam-map(k)} + n_{W(k)} \end{pmatrix} \tag{52}$$

with

$$quat\left(\omega\right) = \begin{pmatrix} \cos\left\|\omega/2\right\| \\ \omega_X/\left\|\omega\right\| \cdot \sin\left\|\omega/2\right\| \\ \omega_Y/\left\|\omega\right\| \cdot \sin\left\|\omega/2\right\| \\ \omega_Z/\left\|\omega\right\| \cdot \sin\left\|\omega/2\right\| \end{pmatrix} \tag{53}$$

and

$$n_V \sim N\left(0,\, P_V\right)$$
$$n_W \sim N\left(0,\, P_W\right) \tag{54}$$

Given the fact that the inverse depth parametrization [14] permits an efficient and accurate representation of uncertainty during undelayed initialization of point-like landmarks, the position of these landmarks is represented in a first stage using 6D inverse depth points q_i:

$$q_i = \left(x_i\ y_i\ z_i\ \theta_i\ \phi_i\ \rho_i\right)^T \tag{55}$$

with $\left(x_i\, y_i\, z_i\right)^T$ the first camera position from which the feature was observed [14], θ_i and ϕ_i azimuth and elevation angles of the first feature observation, and ρ_i the inverse of the distance to the first observation.

The error covariance associated with q_i is given by (40). Every time the uncertainty associated with landmark represented using the inverse-depth parametrization drops below a given

threshold (see details in [14]), the landmark is represented as a 3D Cartesian point p_i.

The pose of rigid-body 3D landmarks is represented using point-quaternions η_i, as explained in Section 3.

The state update equation for point-like and rigid-body 3D landmarks is the identity.

4.2 Visual Observations and Data Association

Observations are generated by computing SURF's interest points and descriptors [21]. Since SURF's interest points computation is based on the use of non-smooth square kernels, they can be computed quickly, but some interest points appear over lines. These interest points have non-repeatable positions because they move on the line from frame to frame. Unrepeatable points are deleted by applying the Harris *cornerness* test [23] on each individual interest point (point with a *cornerness* less than 1E-30 are eliminated). The parameters for the Harris filter are $sd = 1.3$, $si = 2.0$, a $= 0.04$.

Observations, i.e. measured interest points, are compared with estimated observations that are produced by projecting 3D points l_i belonging to point-like, inverse-depth, and rigid-body landmarks onto pixels coordinates. First, point positions are estimated using h_{uv}:

$$h_{uv}^{(i)}\begin{pmatrix} u \\ v \end{pmatrix} = projection \times \left(q_{CAM-MAP}^{-1} \cdot (l_i - t_{CAM-MAP}) \cdot q_{CAM-MAP}\right) \tag{56}$$

Then, pixel coordinates are obtained by using the focal distance in x and y:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \cdot distFoc_x \\ v \cdot distFoc_y \end{pmatrix} \tag{57}$$

In the case of point-like landmarks the points to be projected are the ones defining the landmarks (p_i). In the case of rigid-body landmarks the points to be projected are the rigid-body points associated with the landmark, which position is given by $p_i = q_{LN} \cdot \Pi_i \cdot q_{LN}^{-1} + t_{LN}$, with q_{LN} and t_{LN} the quaternion and point defining the landmark.

Finally, in the case of inverse-depth landmarks, the coordinates of the points to be projected are given by [14]:

$$l_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\begin{pmatrix} \cos\phi_i \sin\theta_i \\ -\sin\phi_i \\ \cos\phi_i \cos\theta_i \end{pmatrix} \tag{58}$$

A planar model is detected on the set of measured descriptors by searching a similarity transformation that relates both sets of associated descriptors. The similarity transformation is computed by using the L&R matching procedure [24, 25], which uses an approximate *nearest neighbor* procedure based on a kd-tree representation for generating matches between descriptors, a Hough transform for filtering outliers, and several tests to reject incorrect transformations. The system works by generating correspondences between keypoints from both sets of descriptors, then it uses differences in position, orientation and scale associated to each correspondence for computing similarity transformations. Hypothesis with high consensus are used to generate an affine transformation that relate both images, and several consistence tests are done for rejecting transformations with low score, transformations suffering from excessive distortion and for deleting wrong matches in correct transformations. When camera rotates, all keypoints are displaced in a coherent way and the system is able to find the transformation that relates all of the displacements, which give it the ability to cope with significant camera rotations. Transformations that have an excessive associated translation or scaling are rejected as possible detections. A chi-square test is used to reject some false landmark detections that can survive the tests. For rigid body 3D landmarks, 7×7-dimensional covariance S matrices are used for the chi-square test, while for the point-like landmarks, 2×2-dimensional matrices are used. The similarity transformation stage can be relaxed when the camera is lost.

This system does not use pixel tracking, but uses landmark detection in each frame. Then, loop closure occurs naturally as old descriptors are found.

4.3 SLAM Algorithm Formulation

The SLAM algorithm includes the following stages: SURF features detection, Matching of SURF features with point-like, inverse-point and rigid-body landmarks, EKF State Prediction, EKF State Update, Inverse-depth landmarks collapsing, Point-like landmarks collapsing, Inverse-depth landmarks generation, and Inverse-depth landmarks deletion.

1. *SURF features detection*. SURF features are detected in the current image, and translated into normalized pixel coordinates as described in Section 4.2.
2. *Matching of SURF features with point-like, inverse-point and rigid-body landmarks.* As outlined in Section 4.2, the L&R matching system is used, which includes several rejection tests.
3. *EKF State Prediction*. The state x_k and the covariance matrix of the state P_k are updated using the standard EKF prediction step [26]. The camera state is updated using (52) and (53). The state update equation of the landmarks is the identity. P_k is updated using the usual EKF procedure.
4. *EKF State Update*. The observation model is used to update the system state and covariance by using the difference between the expected and real values of the observations for correcting the model.

As usual, the innovation y_k and the innovation covariance S_k are computed as:

$$\begin{aligned} y_k &= z_k - \mathbf{H}_k \cdot x_k^- \\ S_k &= \mathbf{H}_k \cdot P_k^- \cdot \mathbf{H}_k^T + R_k \end{aligned} \tag{59}$$

Four different cases for the innovation computation need to be considered:

- In the case of point-like landmarks, z_k and R_k are given by (33) and (40), respectively, and $\mathbf{H}_k$ is the Jacobian matrix of partial derivatives of $h_{uv}^{(i)}$ (given by (56)), with respect to x.
- In the case of inverse-depth landmarks, z_k and R_k are given by (33) and (40), respectively, and $\mathbf{H}_k$ is the Jacobian of $h_{uv}^{(i)}$, given by (56) and (58).
- In the case of rigid-body landmarks, z_k and R_k are given by (38) and (39), respectively, and $\mathbf{H}_k$ is the Jacobian of h_{rb3D}, given by (43).
- In case a virtual observation can not be obtained for an existing landmark because not enough body points are observed, body points can also be used in the correction process. In this case, for each observed body point, z_k and R_k are given by (33) and (45), respectively, and $\mathbf{H}_k$ is the Jacobian of $h_{bp}^{(i)}$, given by (44).

Fast covariance correction can be achieved by decomposing the state covariance matrix P into observed (o) and non-observed (n) components before applying Kalman correction step, as follows:

$$\begin{aligned} K_k &= \left(H_k P_k^-\right)^T S_k^{-1} \\ P_k &= P_k^- - K_k \left(H_k P_K^-\right) \end{aligned} \tag{60}$$

with

$$\begin{aligned} H_k &= \begin{pmatrix} H_o & 0 \end{pmatrix}, \quad P_k^- = \begin{pmatrix} P_{oo} & P_{on} \\ P_{on}^T & P_{nm} \end{pmatrix} \\ H_k P_k^- &= \begin{pmatrix} H_o P_{oo} & H_o P_{on} \end{pmatrix} \end{aligned} \tag{61}$$

In a very small percentage of the frames, numerically unstable state covariance matrices are obtained by using the fast covariance update formula because of floating-point rounding errors. In this work, a covariance matrix is considered unstable if $P_{ii} P_{jj} < P_{ij}^2$ for any combination of (i, j). In that case, covariance correction step is done by using Cholesky downdating, which is a method involving Cholesky decomposition that gives a positive semidefinite matrix as result:

$$\begin{aligned} P_k^- &= L_P L_P^T, \;\; S_k^{-1} = U_S^T U_S \\ K\left(H_k P_K\right) &= \left(H_k P_k\right)^T S_k^{-1} \left(H_k P_k\right)^T \\ &= \left(H_k P_k U_S\right)^T \left(H_k P_k U_S\right) \\ &= \begin{pmatrix} v_1 & v_2 & \ldots & v_{n_o} \end{pmatrix} \begin{pmatrix} v_1 & v_2 & \ldots & v_{n_o} \end{pmatrix}^T \\ P_k &= P_k^- - K H_k P_k^- \Leftrightarrow \left(L_P L_P^T\right)_k \\ &= \left(L_P L_P^T\right)_k^- - \sum_{i=1}^{n_o} v_i v_i^T \end{aligned} \tag{62}$$

A good implementation of Cholesky decomposition is faster than normal matrix multiplication. As a C or C++ efficient code for Cholesky down-

dating is not available, a C version of the zchdd subroutine from LINPACK library [27], originally written in Fortran, was obtained using fable [28].

After each correction step, the quaternion components in state are normalized. To ensure coherence in the SLAM, a Jacobian from the normalizing function is used to propagate normalization effects into the state covariance matrix.

5. *Inverse-depth landmarks collapsing*. Inverse-depth landmarks whose uncertainty drops below a threshold are converted into normal point-like landmarks.
6. *Point-like landmarks collapsing*. The covariance of points P_{pp} is analyzed in order to verify if a set of point-like landmarks exists that can generate a rigid-body landmark. As explained in Section 3.3, the procedure requires verifying whether a variability index associated with a set of point-like landmarks is below a threshold *th* (Eqs. 29, 30, 31). In case a rigid-body landmark can be generated, the procedure described in Section 3.2 is used (Eqs. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26).
7. *Inverse-depth landmarks generation*. Image SURF features that were not matched, i.e. they are distant from landmarks in the image domain, are added as new inverse-depth landmarks, using the procedure described in Section 4.1. By selecting an appropriate distance threshold, the density of observed descriptors in the image can be kept within a desired range.
8. *Inverse-depth landmarks deletion*. New inverse-depth landmarks need to be observed for a certain number of frames in order to be confirmed. If the number of frames in which an inverse-depth landmark was not observed, but was expected to be, is over a certain threshold, the landmark is deleted.

5 Experimental Evaluation

5.1 Simulated Experiments

The system is evaluated by simulating the movement of a camera using four types of trajectories, and applying different visual SLAM approaches for recovering the camera's path. In all cases the simulated camera moves looking all the time at a fixed point. The following six trajectories are used, which have closed form for repeatability purposes.

1. *U-shaped* path
   - Trajectory: $x = -60 \sin\left(\frac{\pi}{2} \sin\left(2\pi \frac{t}{8}\right)\right), y = -90 \cos\left(\frac{\pi}{2} \sin\left(2\pi \frac{t}{8}\right)\right), z = 0$
   - Camera looking at position (0,0,0)
2. *S-shaped* path
   - Trajectory: $x = -40 \frac{1+t}{54} \cos\left(\cos\left(\frac{t}{3t}\right)\right), y = 40 \frac{1+t}{54} \sin\left(\cos\left(\frac{t}{3t}\right)\right), z = 0$
   - Camera looking at position (30,0,0)
3. *Continual Lost* path
   - Trajectory: S-shaped path (80 s) followed by a square path with four very distant points (−90, −40), (−90, 40), (−30, 40), (−30, −40). Each periodic sequence takes 4 s, and the transition between the four points is done without any delay.
   - Camera looking at position (30,0,0)
4. *S-shaped+Random-Walk* path
   - Trajectory: S-shaped path (80 s) followed by a random walk: $x_t = x_{t+1} + n_X$, $y_t = y_{t+1} + n_Y$, $z_t = z_{t+1} + n_z$, $n_X, n_Y, n_Z \rightarrow N(0, 100)$.
   - Camera looking at position (30,0,0)

The trajectories were sampled into control points by using a 1 s step. Intermediate points were calculated by using *spline* interpolation in the point-quaternion space. A set of 900 SURF descriptors are generated randomly in a 400×400 square area centered at the origin of the global coordinate system by using a uniform distribution; 64D values for the SURF descriptors are initialized in a random way by using a uniform distribution followed by a normalization. Some of the descriptors will give rise to landmarks when observed by first time, and then each landmark corresponds to a unique known feature in space, which enables comparing landmarks and original features. The frame rate of the simulated camera is 15 fps. Gaussian noise with a standard devia-

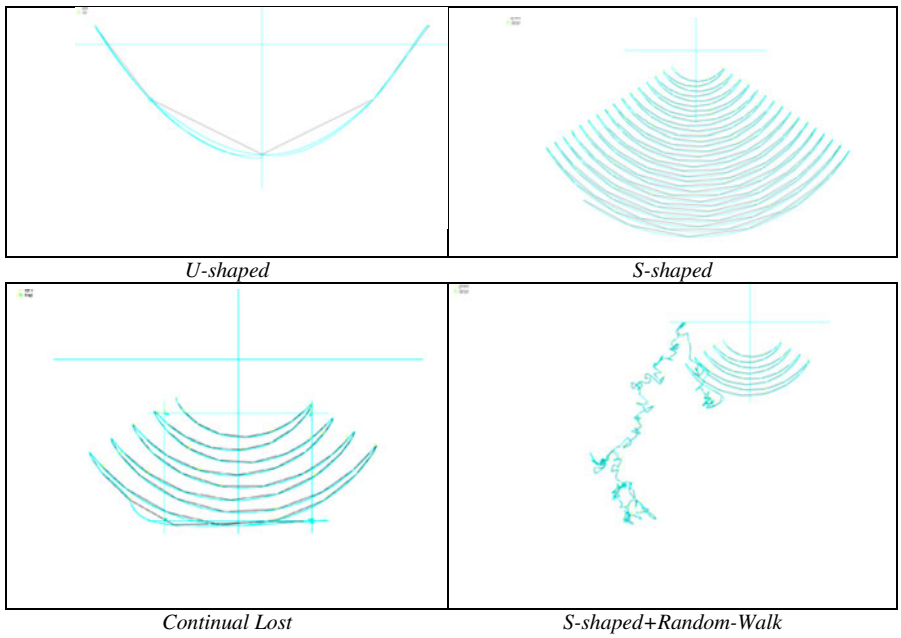

Fig. 1 Simulated camera trajectories used in experiments

tion of six pixels was added to the observations in order to simulate noise which is intrinsic to SURF detection process. The simulated camera has a resolution of 320 × 200. A visualization example of each path, showing both the control points and the *spline* interpolation is shown in Fig. 1.

Several simulation tests were carried out in each of the paths. Each test takes 2,800 frames, and it is started with the restriction of having a maximum of 60 landmarks:

Test 1: Only point-like landmarks are used.
Test 2: All kind of landmarks are used. The maximal pose covariance criterion is used in case of body-point 3D landmarks.
Test 3: All kind of landmarks are used. The maximal body-points covariance criterion is used in case of body-point 3D landmarks.
Test 4: Only point-like landmarks are used. The maximum number of landmarks is constrained to 4 at frame 1,800.
Test 5: All kind of landmarks are used. The maximal pose covariance criterion is applied. The maximum number of rigid body landmarks is constrained to 4 at frame 1,800, and no point-like landmarks are used.
Test 6: Same as test 5, but the maximal body-points covariance criterion is applied.

In all cases, point-like landmarks are first created as inverse-depth landmarks.

Given that map building by using a single camera can produce differences in position, orientation and scale respect to the ground-truth, an optimal transformation that consider all three characteristics is found and applied to experimental path for making comparison possible (least squares procedure). The average Euclidean distance between correspondent pairs of points in the ground truth and the computed paths, i.e. pairs of points that correspond to the same time, is used as error measurement.

In Fig. 2 are shown some examples of recovered paths together with the corresponding ground truth. As obtained results need to be analyzed very carefully, histograms of the errors will be presented in addition to mean errors and standard deviation values. In the histograms visualization, errors with values over 60 will be cut to that valor for maintaining an adequate scale, and they will be considered failures. Table 1 presents experimental results in terms of mean error, standard deviation, and failure percentage for all experiments, while Figs. 3, 4, 5, 6 shows the histograms

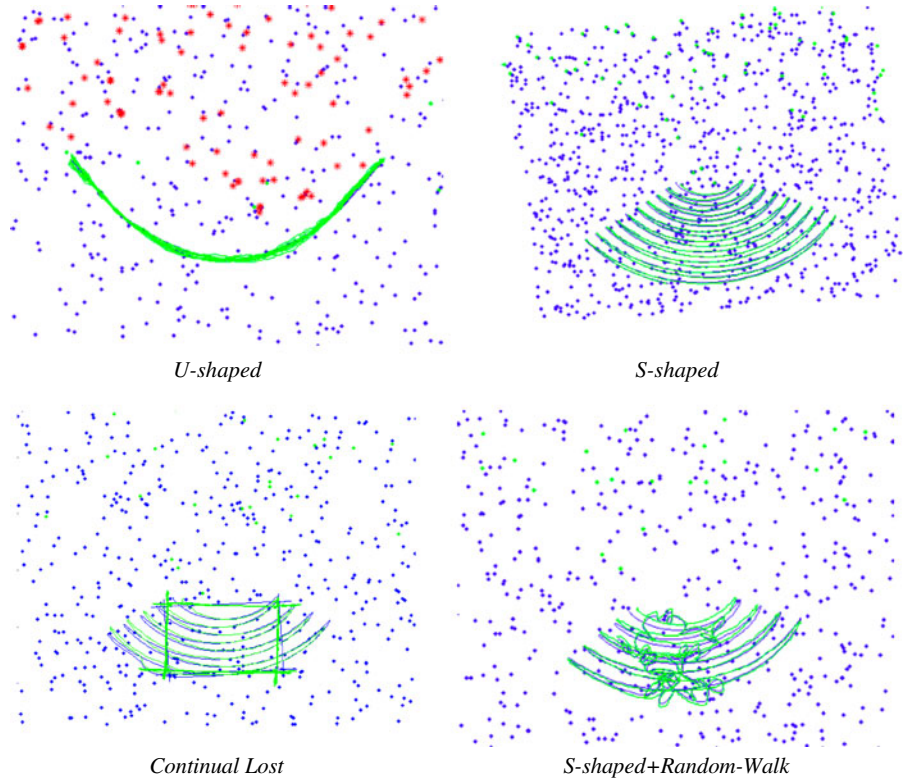

Fig. 2 Simulation example of the recovered paths drawn over the ground truth one for each kind of path is shown. The set of all features is shown in *blue*, the set of features that were selected as landmarks is shown in *green* and current landmarks are shown in *red*

Table 1 Experimental results of visual SLAM for the different trajectories and simulated tests

Path	Test	Max num. Landmarks	Mean error	Standard deviation	Failure percentage
U-shaped	1	60	1.79	0.69	0%
U-shaped	2	60	8.02	5.80	0%
U-shaped	3	60	3.67	0.84	0%
U-shaped	4	4	–	–	100%
U-shaped	5	4	41.03	16.18	76.7%
U-shaped	6	4	4.19	0.93	0%
S-shaped	1	60	6.81	3.53	0%
S-shaped	2	60	26.12	20.78	86.67%
S-shaped	3	60	10.95	6.64	0%
S-shaped	4	4	18.34	5.67	90%
S-shaped	5	4	–	–	100%
S-shaped	6	4	13.22	6.52	0%
Continual lost	1	60	11.69	3.18	0%
Continual lost	2	60	30.07	14.49	70%
Continual lost	3	60	25.34	10.9	6.67%
Continual lost	4	4	–	–	100%
Continual lost	5	4	–	–	100%
Continual lost	6	4	25.75	6.25	0%
S-shaped+R	1	60	2.54	1.24	0%
S-shaped+R	2	60	20.89	13.66	0%
S-shaped+R	3	60	3.80	1.10	0%
S-shaped+R	4	4	32.97	7.10	0%
S-shaped+R	5	4	29.75	7.87	0%
S-shaped+R	6	4	3.57	0.86	0%

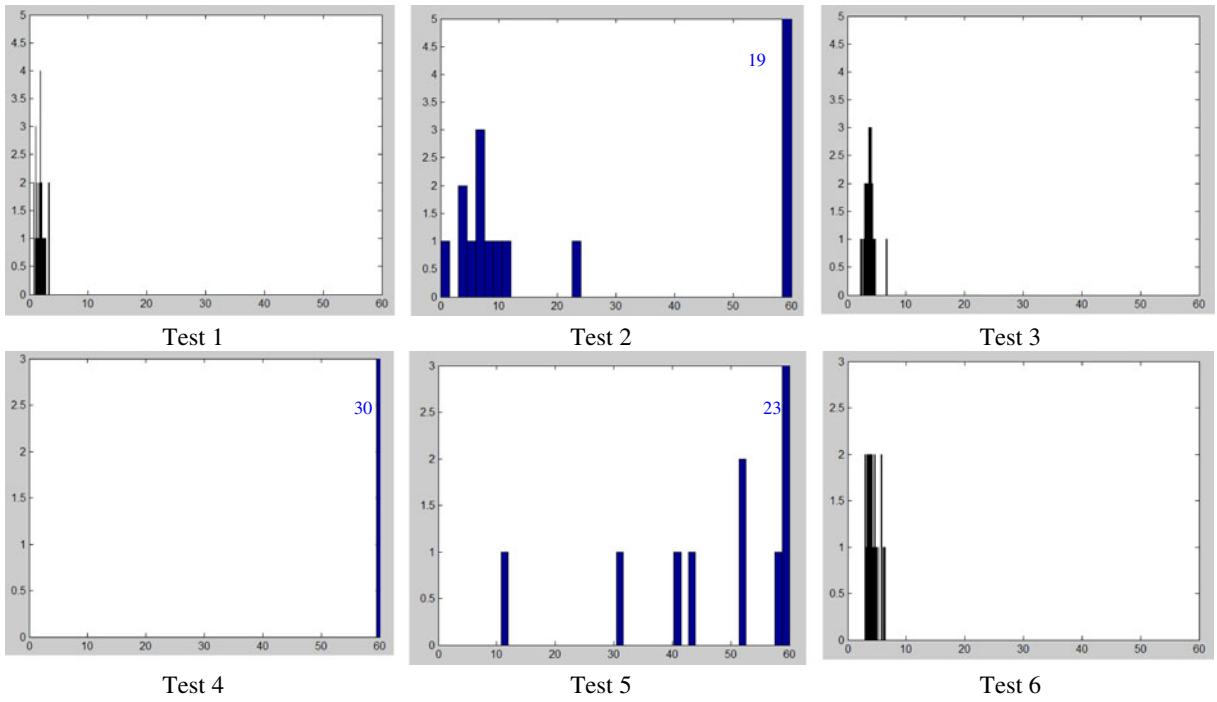

Fig. 3 Histograms of tests applied over the *U-shaped* path

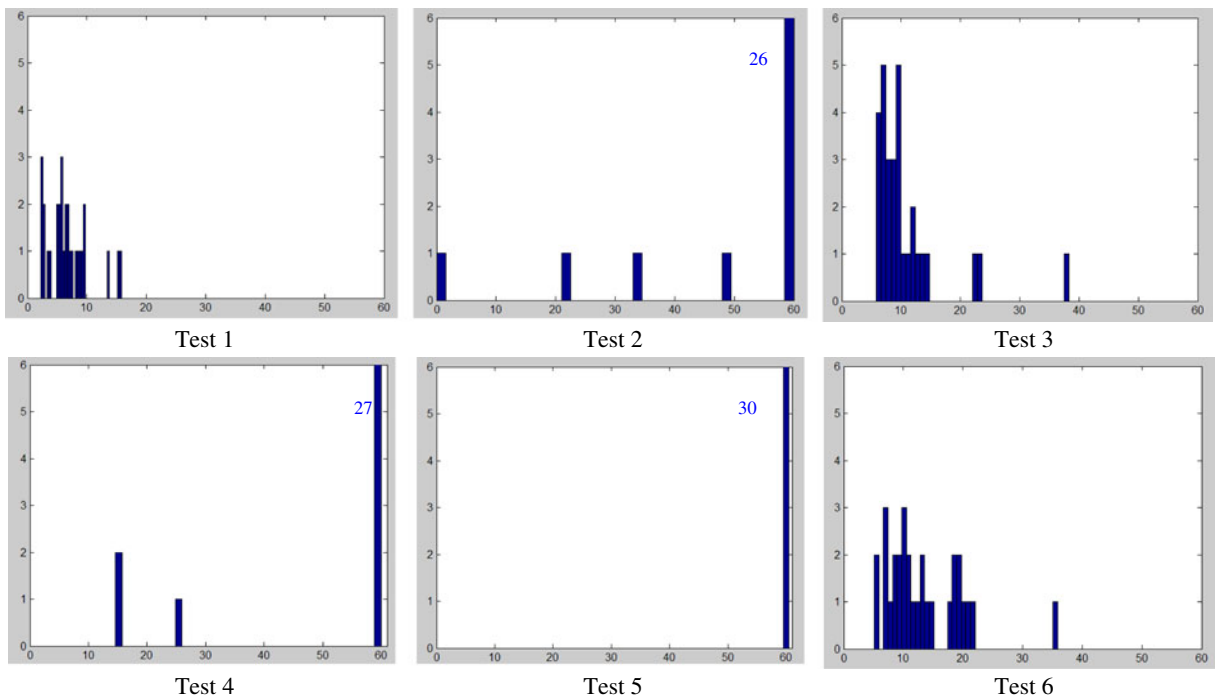

Fig. 4 Histograms of tests applied over the *S-shaped* path

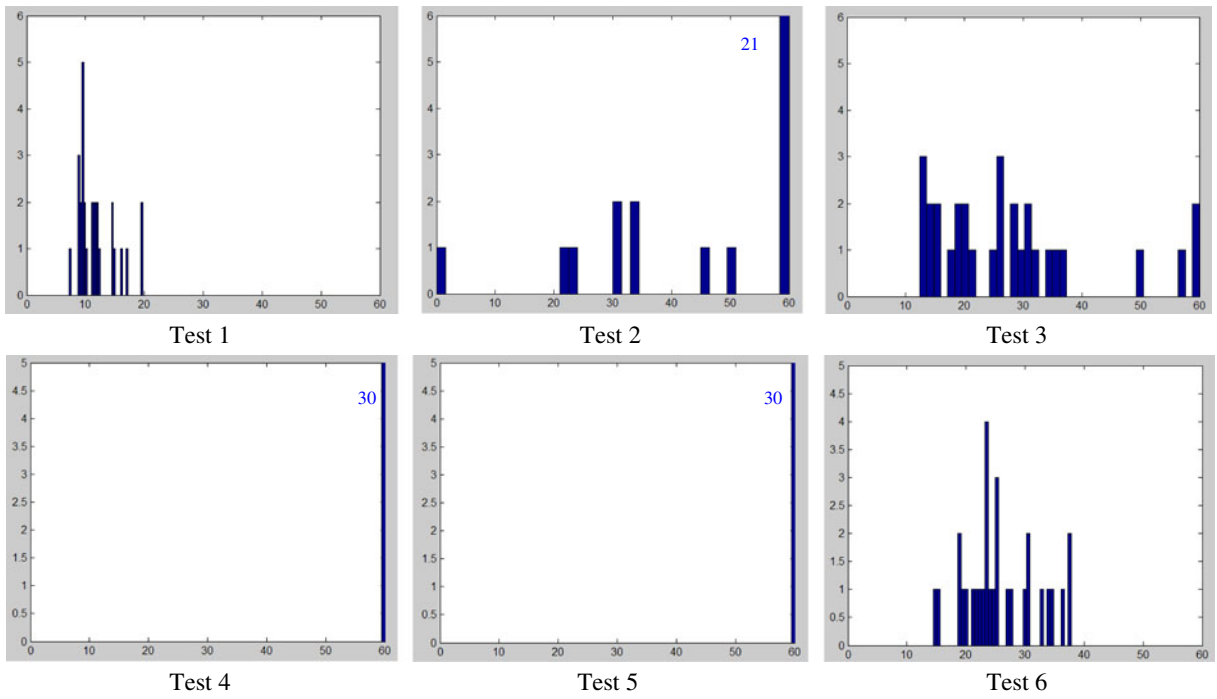

Fig. 5 Histograms of tests applied over the *Continual Lost* path

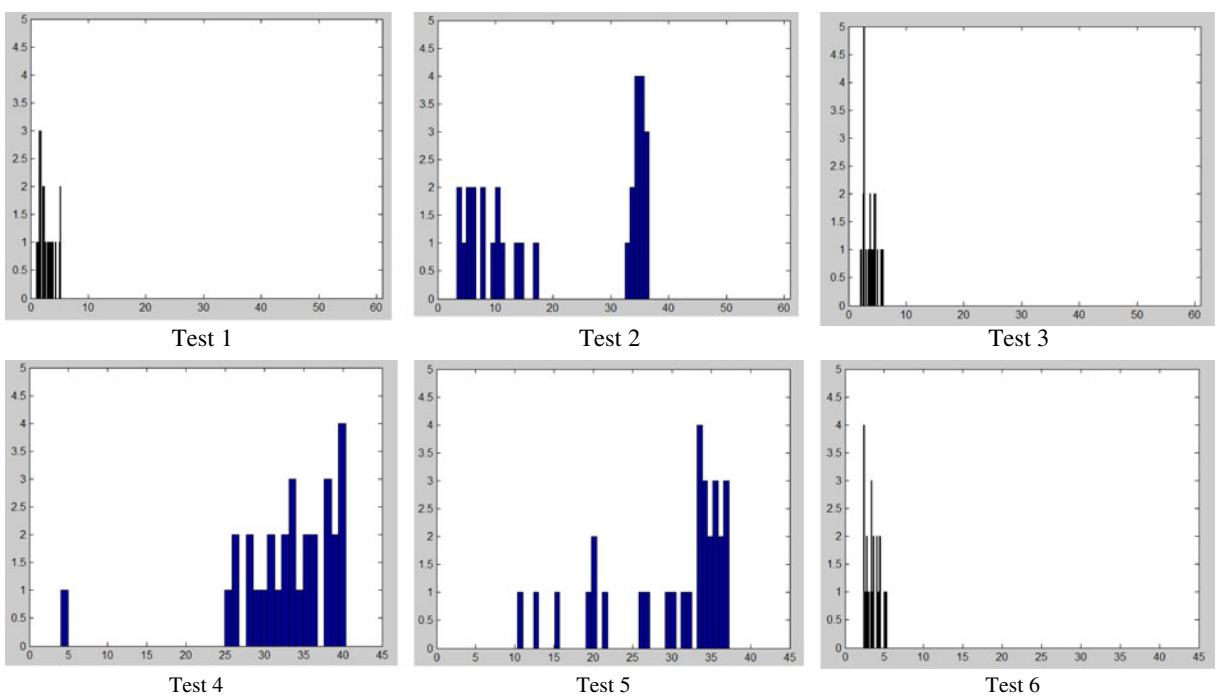

Fig. 6 Histograms of tests applied over the *S-shaped + Random walk* path

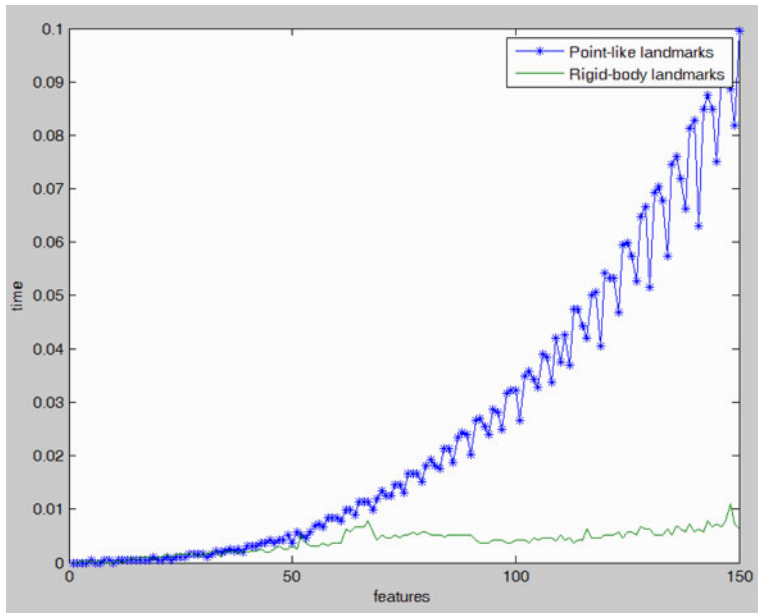

Fig. 7 Execution time for SLAM prediction-update steps versus number of features in the map

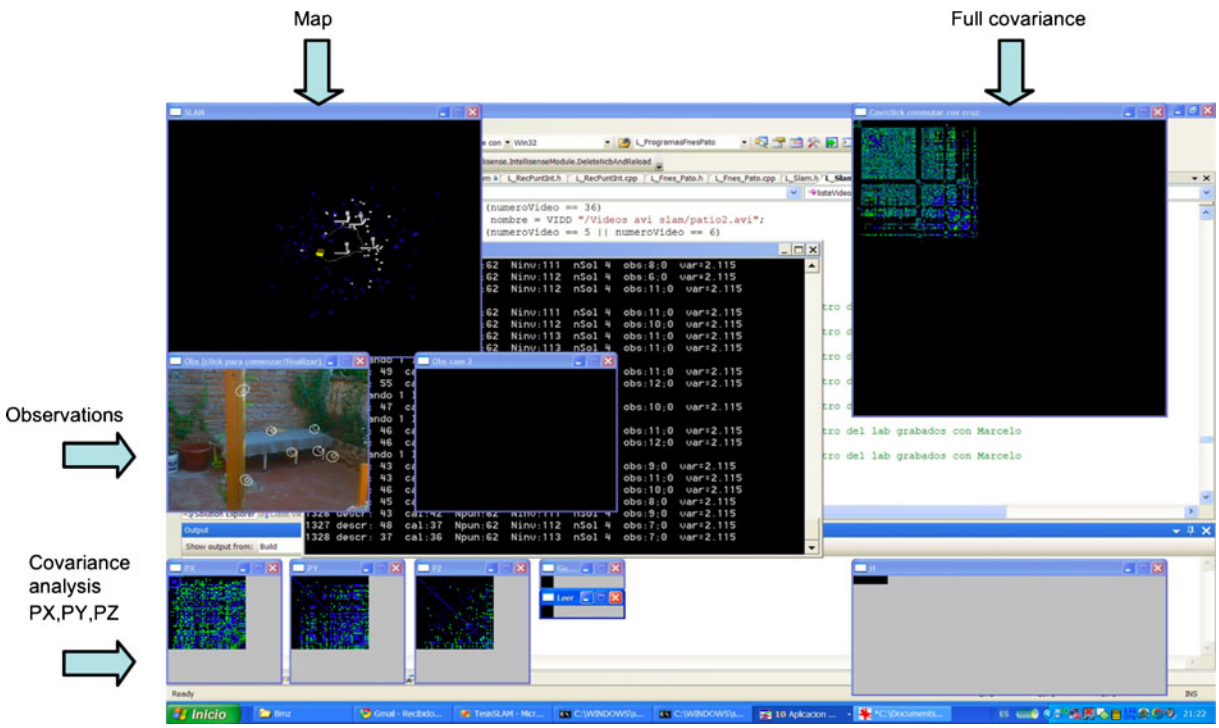

Fig. 8 Visualization example of the SLAM system working in a video sequence. Observations are shown as *rhombs* over imposed on captured image and innovation covariance is drawn as a set of *ellipses*. Point-like landmarks are drawn as *blue dots*. Rigid body landmarks are drawn as *white reference systems with white body points*. Matrices C_X, C_Y and C_Z, used to evaluate variability index, are shown *bottom left*. Full covariance matrix is shown *top right*

Fig. 9 Selected images from the garden video database. In each image *rhombs* correspond to real observations (SURF interest points), and *ellipses* represent the innovation covariance

for the errors. In all cases each test was run 30 times in every path in order to generate robust statistics.

In the case of the *U-shaped*, *S-shaped*, and *S-shaped+Random-Walk* paths (see Table 1 and Figs. 3, 4, 5, 6), it can be observed that when the number of landmarks is limited to 60, the best option is to use point-like landmarks, and rigid-body landmarks using the maximal body-points covariance criterion produce a slightly larger error.

However, when the number of landmarks is very small (limited to 4), rigid-body landmarks using the maximal body-points covariance criterion show an impressive advantage over point-like landmarks, as the reduction of the number

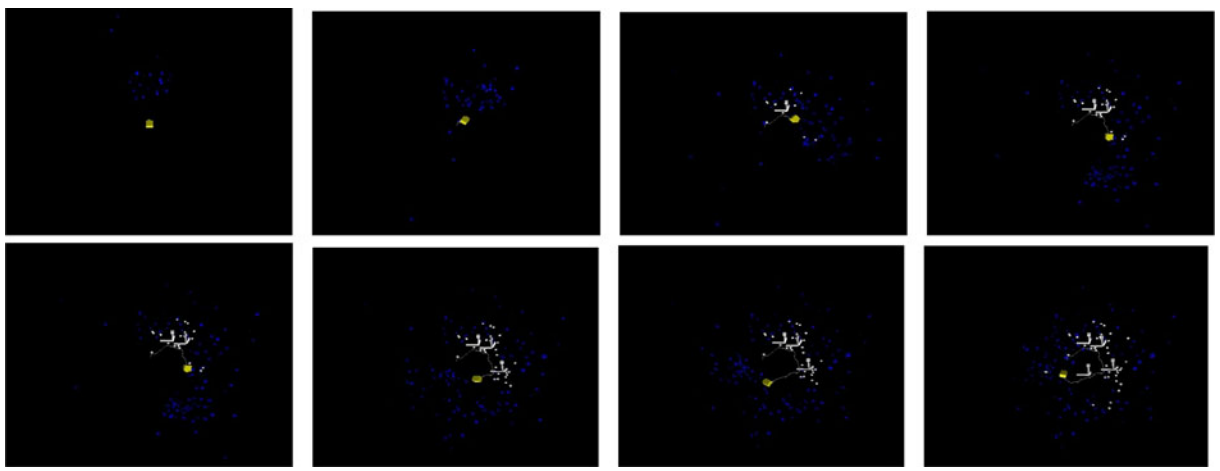

Fig. 10 Example of the visual SLAM system running in one of the real video sequence. The camera is shown in *yellow*. *Blue dots* correspond to point-like landmarks, while *white structures* consisting in three perpendicular axis and a set of white body points denote rigid-body landmarks

of landmarks produce only a very weak error increase. Even in some cases where the use of point-like landmarks fails completely (*U-shaped* path and test 4), the use of rigid-body landmarks with the maximal body-points covariance criterion behave appropriately. In all cases the use of the

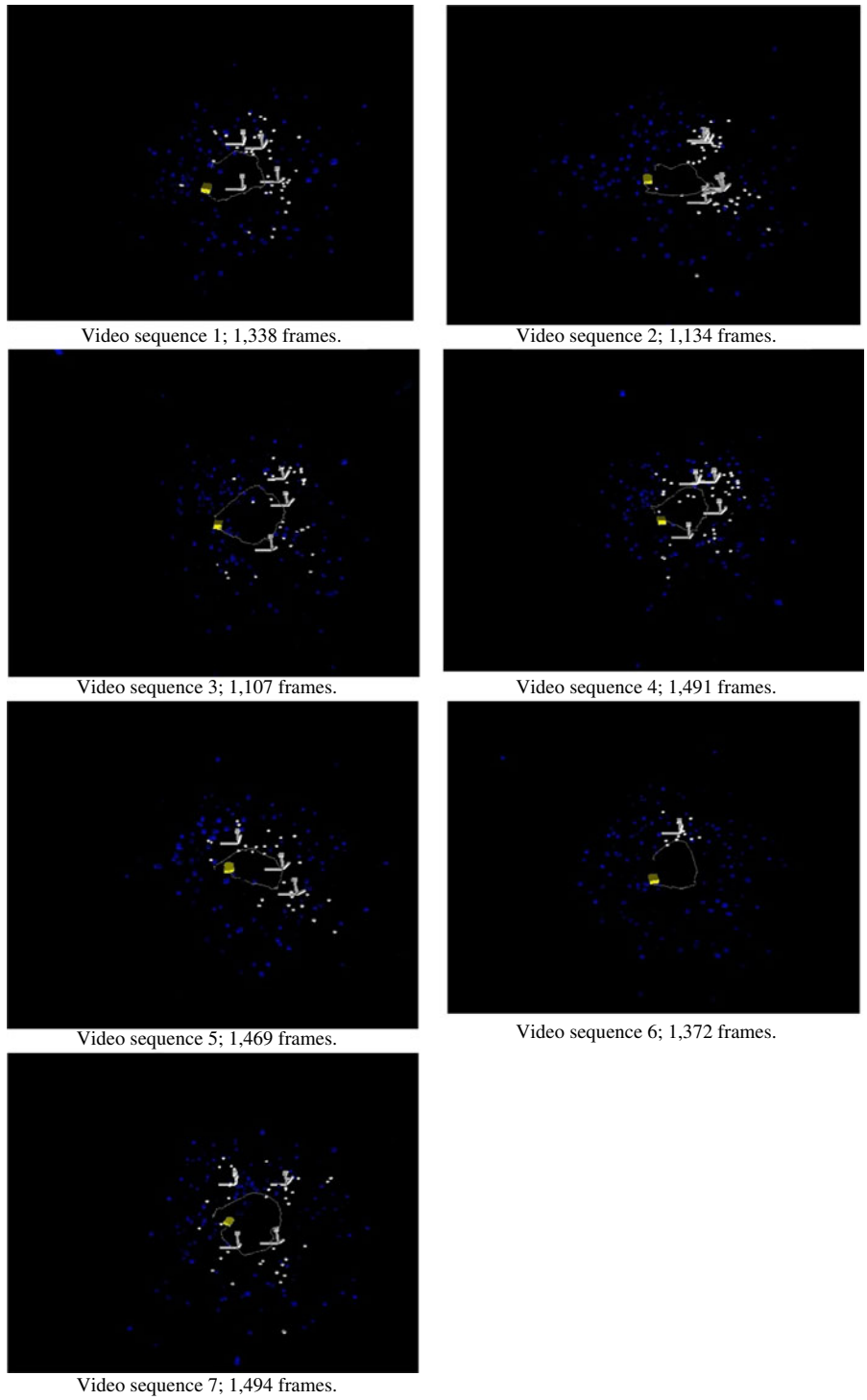

Fig. 11 Maps and reconstructed paths for the seven tested videos. The camera is shown in *yellow*. *Blue dots* correspond to point-like landmarks, while *white structures* consisting in three perpendicular axis and a set of white body points denote rigid-body landmarks

maximal body-points covariance criterion appears as the best option.

The *Continual Lost* path is a very hard test as it involves instantaneous and very large changes of the position of the camera, which can produce divergences in the SLAM system. Point-like landmarks behave better when the number of observed features is high. However, errors involved in all of the results are very high as they are of the same magnitude order than the size of the full path (around 30). In the case of using just four landmarks in the map, point-like landmarks are not able to follow the path, failing in all of the cases. Rigid-body landmarks with the maximal body-points covariance criterion are able to follow appropriately the path.

From the experimental data, it is clear that in cases where the number of landmarks needs to be limited, because of computational reasons or because of the large size of the map, the use of rigid-body landmarks is very useful. In addition, it is evident that it is convenient to propagate the most possible quantity of covariance from the original point landmarks into the body point covariances. As the positions of body points are not adapted before its creation, the error associated to them does not decrease, and then its covariances must be constant over time. If the covariance from the original points is propagated mainly into the covariance of the pose, covariance from rigid-body landmarks will be underestimated because the covariance of the pose decrease to zero when it is observed, and the covariance from body points remains very low. This fact can cause a severe covariance underestimation when observing a landmark several frames before its creation. Then, maximizing propagation of original covariance into body point covariances is the best option.

Execution times for SLAM, including both prediction and update steps, were measured as a function of the number of features used in the SLAM system. In the runtime experiment, the path and feature configuration used in the *U-shaped* path test was selected. As it can be observed in the results shown in Fig. 7, the ratio between execution times for a SLAM using only point-like points versus a SLAM using rigid-body landmarks converges to the 5.5% limit, as the number of features in the map increases. This can be explained because matrix operations in the EKF update step become the most expensive computation in SLAM when the number of features is high, because of their quadratic nature. Then the state

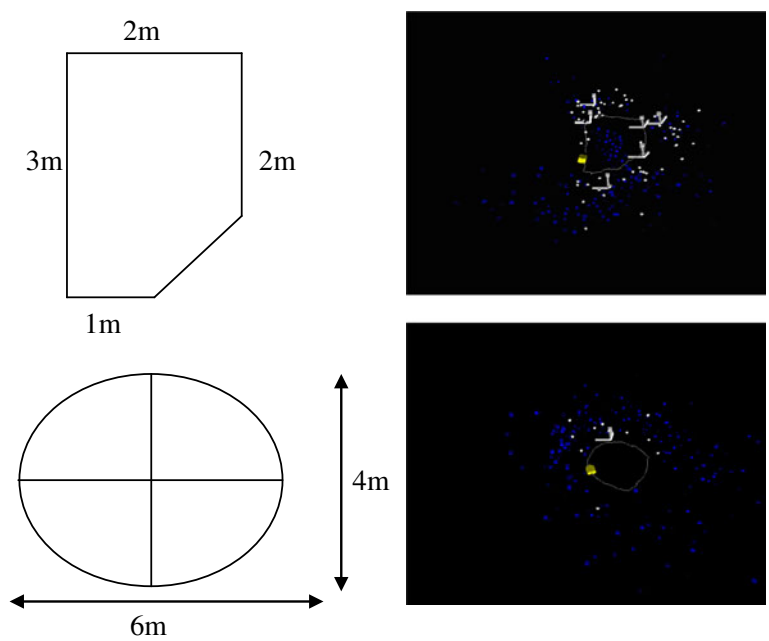

Fig. 12 Camera moving in a polygonal trajectory (*first*), and moving on an elliptical trajectory (*second row*). In both cases, the ideal camera trajectory, as well as the maps and the reconstructed paths are shown

reduction capabilities of the rigid body approach have an impressive impact in the execution time. The performance penalty related to the computation of virtual observations is very low when compared to matrix operations, as they grow in a linear fashion. This experiment was realized in an

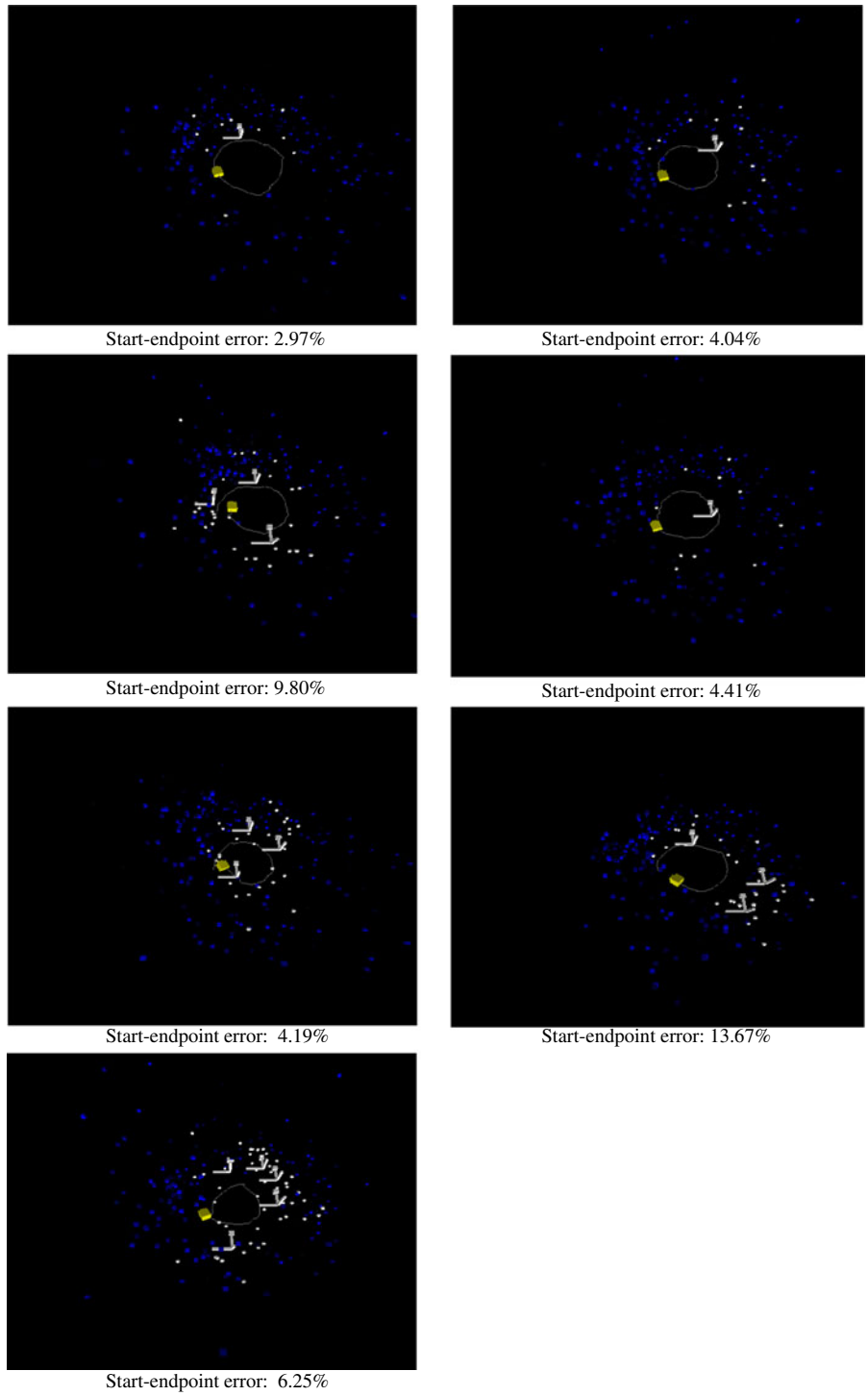

Fig. 13 Recovered elliptic paths, and error between start and end points

Intel Core Duo processor at 1.6 GHz using only one core.

5.2 Experiments with Real Video Sequences

In a first set of experiments, the system was evaluated qualitatively. A handheld camera was used to produce seven video sequences in an outdoor environment. In order to generate the sequences, the handheld camera followed a path inside a house's garden. In each of the cases, the path finished near its starting point. The camera has a normal lens, the video sequences were captures at 30 fps, and their duration in frames is 1,338, 1,134, 1,107, 1,491, 1,469, 1,372 and 1494.

The proposed visual SLAM system using point-like and rigid-body 3D landmarks was tested in these video sequences. The system runs in a standard low-end laptop. Figure 8 shows a visualization tool used to analyze the performance of the visual SLAM system in the video sequences. The poses from rigid bodies are represented by using a small reference system represented by using three orthogonal axis x, y, z, which are drawn in white. In Fig. 9 some selected images from the garden video database are shown. Figure 10 shows an example of the visual SLAM system running in one of the real video sequence.

In all of the tests the proposed SLAM system was able to build a coherent map and to recover the path. In Fig. 10 the reconstructed paths for the seven tested videos are shown: the camera is shown in yellow, point-like landmarks are shown as blue dots, and rigid-body landmarks are denote as white structures.

The system was also able to recognize the first generated landmarks (loop closing) easily because no tracking is used; instead descriptor matching between the map and the current image's observations is done by using L&R system. The robustness of the matching system is reflected in its capacity to recover the seven paths that were tested (Fig. 11).

In a second set of experiments, the system was evaluated quantitatively using ground-truth paths of specific regular shapes. In the initial experiments, about 40 runs were carried out in different environments, using different polygonal paths containing right angles, but the results were inaccurate in around half of the cases. The system was able to reconstruct the angles from the polygons, but estimations of the sides were not regular, and the estimated pose of camera sometimes moved long distances when the loop was closed. The explanation found is that straight angles cause a loss of the speed information as the camera must stop moving, and features come out of the camera when it turns in zones of the path with high curvature. Both problems limit scale preservation. This problem is aggravated by the decision of using a standard narrow-angle camera instead of a wide angle one, which could provide major parallax for the points when moving. Then, movements of the camera in visual SLAM cannot be arbitrary as they require some softness, as some parallax on the features is required to estimate the map, and the speed of the camera helps to preserve scale.

As paths with sharp angles were troublesome, elliptical paths, as the ones shown in Fig. 12, were selected in order to generate seven video sequences for testing purposes. As ellipses have some degree of rotational symmetry, the mean absolute error between the best possible ellipse and the recovered path can cause an underestimation of the error. This occurs because errors in length of the path will not produce errors as long as path remains into the ellipse. For this reason, the distance between the initial and final points from the recovered path was used as a quantitative measure of the accuracy of the proposed system. Figure 13 shows the recovered paths and the error between the start and end points, normalized respect to the length of the ellipse, for ach case. It can be observed that in most of the cases the start-end point error is smaller than 10%, and that its mean value is 6.47%.

6 Conclusions

In this work a visual SLAM system based on the use of what are called *rigid-body 3D landmarks* was proposed. A rigid-body 3D landmark represents the 6D pose of a rigid body in space, and its observation gives full-pose information about a bearing camera. The use of rigid-body 3D landmarks permits reducing the computational time of the EKF-SLAM system up to 5.5%, as the

number of landmarks increases. The proposed visual SLAM system was validated in simulated data and real video sequences using a standard, low cost camera. Remarkably the system performs very well in outdoor environments, allowing very good camera localization.

The analysis of the visual SLAM system operation in real video sequences shows that the implemented system has a good performance when tested in the real-world. Rigid-body 3D landmarks are able to reduce the state dimensionality in unstructured environments with low information loss, which enables the camera to recover the full path in a reliable way, avoiding EKF covariance overload. SURF descriptors with delayed Harris testing are both fast and repeatable enough to provide good quality information about structures in the real-world, even when systems with limited computing capabilities are used. Data association based on L&R system, which has been created for robust object recognition, shows very good performance for map association tasks and enables the EKF to work without map corruption due to wrong associations even in long video sequences, and without needing special loop-closing techniques as all the features have the same opportunity for being detected in every frame, because no features are tracked. Results show that the rigid-body landmarks paradigm is both promising and powerful, and new field application can be explored in future works.

The experimental data indicates that the visual SLAM system achieves good localization when the number of observed landmarks is very low, working very well with only four landmarks being available for observation permanently, which is possible by using feature-rich individual landmarks. This property enables them to be used in the generation of large maps as very few landmarks per area are needed. As body-points are parameters and no states in this system, their error does not decrease in time, which can explain the slight better performance of point-like landmarks when the density of landmarks is very high. Possible adaptation of body points by creating a dynamical sub system inside each rigid body landmark (EKF-like adaptation of positions and covariances from individual body points) and the use of semantical cues for improving selection of rigid bodies remains open problems that can be addressed in future work.

Acknowledgments This research work was partially funded by the doctoral grant program of CONICYT (Chile), by MECESUP Project FSM 0601, and by FONDECYT project 1090250.

Appendix

Computation of Jacobians $J_{UV}^{(i)}$ and $J_{\Pi}^{(i)}$, defined in (41) and (42).

As function $V(\cdot)$ is the result of an iterative minimization (see (38)), the computation of $J_{UV}^{(i)}$ and $J_{\Pi}^{(i)}$ by using finite differences over several minimizations is a very slow process. The partial derivatives of $E_P(\cdot)$respect to the pose must be zero when evaluated at the optimal value, this leads to a closed form for the Jacobians. For simplifying the notation, the vector $a = (u_1, v_1, ..., \Pi_1, ..., \Pi_N)^T$ collecting all the parameters will be used in the following expressions:

$$V(a) = \arg\min_{\eta} \left(E_P(\eta; a) \right) \tag{63}$$

$$\frac{\partial}{\partial \eta_i} E_P(\eta; a) \,|\eta = V(a) = 0, \ \forall i \tag{64}$$

$$\frac{d}{da_j} \left(\frac{\partial}{\partial \eta_i} E_P(V(a); a) \right) = 0, \ \forall i, j \tag{65}$$

$$\sum_k \frac{\partial^2 E_P}{\partial \eta_i \partial \eta_k} \frac{\partial V(a)_k}{\partial a_j} + \frac{\partial^2 E_P}{\partial \eta_i \partial a_j} = 0 \ \forall i, j \tag{66}$$

The last expression can be converted into matrix form by making the following definitions:

$$E_{\eta\eta}(V(a); a)_{(i,j)} = \frac{\partial^2 E_P}{\partial \eta_i \partial \eta_j} \tag{67}$$

$$E_{\eta A}(V(a); a)_{(i,j)} = \frac{\partial^2 E_P}{\partial \eta_i \partial a_j} \tag{68}$$

After the replacements, the following expressions hold.

$$E_{\eta\eta}(V(a); a) \frac{\partial V}{\partial a}(a) + E_{\eta A}(V(a); a) = 0 \tag{69}$$

$$\Rightarrow \frac{\partial V}{\partial a}(a) = -E_{\eta\eta}^{-1}(V(a); a) \, E_{\eta A}(V(a); a) \tag{70}$$

The last expression has a closed form, and enables a straightforward computation of the Jacobians of $V(\cdot)$. As the quaternion is a non-minimal representation for rotations, there is a direction in the observation vector that contains no real information, then variations of the vector in that direction leaves the value of the error unmodified. In consequence, the Hessian has a null space and cannot be inverted directly. The problem can be solved by computing the inverse using an eigenvalue decomposition and by bounding the smallest eigenvalue from the Hessian by a small value (e.g. 10^{-30}).

References

1. Filliat, D., Meyer, J.-A.: Map-based navigation in mobile robots: I. A review of localization strategies. Cogn. Syst. Res. **4**(4), 243–282 (2003)
2. Neira, J., Davison, A.J., Leonard, J.J.: Guest editorial special issue on visual SLAM. IEEE Trans. Robotics **24**(4), 929–931 (2008)
3. Hauke Strasdat, J., Montiel, M. Davison, A.J.: Scale Drift-Aware Large Scale Monocular SLAM, RSS (2010)
4. Civera, J, Grasa, O.G., Davison, A.J., Montiel, J.M.M.: 1-point RANSAC for EKF-based Structure from Motion, IROS 2009 Proceedings, pp. 3498–3504
5. Handa, A., Chli, M., Strasdat, H., Davison, A.J.: Scalable Active Matching, Proc. 2010 IEEE Conf. on Computer Vision and Pattern Recognition, June 13–18, 2010, San Francisco
6. Davison, A.J., Reid, I.D., Molton, N., Otasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
7. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. J. Field Robot. **23**(1), 3–20 (2006)
8. Nistér, D.: Preemptive RANSAC for live structure and motion estimation. Mach. Vis. Appl. **16**(5), 321–329 (2005)
9. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera, ICCV 2003 Proceedings, pp. 1403–1410 vol. 2
10. Newcombe, R, Davison, A.J.: Live Dense Reconstruction with a Single Moving Camera, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)
11. Angeli, A., Davison, A.J.: Live Feature Clustering in Video Using Appearance and 3D Geometry, BMVC (2010)
12. Gee, A.P., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering higher level structure in visual SLAM. IEEE Trans. Robotics **24**(5), 980–990 (2008)
13. Kwon, J., Lee, K.M.: Monocular SLAM with Locally Planar Landmarks via Geometric Rao-Blackwellized Particle Filtering on Lie Groups, 2010 IEEE Conf. on Computer Vision and Pattern Recognition-CVPR 2010, pp. 1522–1529, 13–18 June 2010, San Francisco, USA
14. Civera, J., Davison, A.J., Martínez-Montiel, M.: Inverse depth parametrization for monocular SLAM. IEEE Trans. Robotics **24**(5), 932–945 (2008)
15. Pathak, K., Birk, A., Vaskevicius, N., Poppinga, J.: Fast registration based on noisy planes with unknown correspondences for 3D mapping. IEEE Trans. Robotics **26**(3), 424–441 (2010)
16. Kohlhepp, P., Pozzo, P., Walther, M., Dillmann, R.: Sequential 3D-SLAM for mobile action planning, Proc. 2004 IEEE/RSJ International Conf. on Intelligent Robots and Systems, Sendai, Japan
17. Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., Poppinga, J.: Online 3D SLAM by registration of large planar surface segments and closed form pose-graph relaxation. J. Field Robot. **27**(1), 52–84 (2009)
18. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3D-NDT. J. Field Robot. **24**(10), 803–827 (2007)
19. Hamilton, W.R.: On quaternions, or on a new system of imaginaries in algebra. Philos. Mag. **25**(3), 489–495 (1844)
20. Loncomilla, P.: Generación automática de landmarks visuales naturales tridimensionales basada en descriptores locales para auto-localización de robots móviles, Ph.D. Thesis, Universidad de Chile (2010)
21. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
22. Haralick, B.M., Lee, Ch.-N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. Int. J. Comput. Vis. **13**(3), 331–356 (1994)
23. Harris, C., Stephens, M.: A combined corner and edge detector. Proc. of the 4th Alvey Vision Conference, pp. 147–151
24. Loncomilla, P., Ruiz del Solar, J.: A fast probabilistic model for hypothesis rejection in SIFT-based object recognition, Lecture Notes in Computer Science 4225 (CIARP 2006). Springer, 696–705
25. Ruiz-del-Solar, J., Loncomilla, P.: Robot head pose detection and gaze direction determination using local invariant features. Adv. Robot. **23**(3), 305–328 (2009)
26. Welch, G., Bishop, G.: An introduction to the Kalman filter. University of North Carolina, Chapel Hill (1995)
27. LINPACK library official site: http://www.netlib.org/linpack/
28. Grosse-Kunstleve, R.W., Terwilliger, T.C., Adams, P.D.: Experience converting a large Fortran-77 program to C++. IUCr Comp. Comm. **10**, 75–84 (2009)