Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 671376, 11 pages
http://dx.doi.org/10.1155/2013/671376

Research Article

Validation of Data Association for Monocular SLAM

Edmundo Guerra,[1] Rodrigo Munguia,[2] Yolanda Bolea,[1] and Antoni Grau[1]

[1] *Automatic Control Department, Technical University of Catalonia UPC, 08034 Barcelona, Spain*
[2] *Computer Science Department, CUCEI, University of Guadalajara, 44430 Guadalajara, Mexico*

Correspondence should be addressed to Antoni Grau; antoni.grau@upc.edu

Received 25 November 2012; Revised 27 February 2013; Accepted 15 March 2013

Academic Editor: Quang Phuc Ha

Simultaneous Mapping and Localization (SLAM) is a multidisciplinary problem with ramifications within several fields. One of the key aspects for its popularity and success is the data fusion produced by SLAM techniques, providing strong and robust sensory systems even with simple devices, such as webcams in Monocular SLAM. This work studies a novel batch validation algorithm, the highest order hypothesis compatibility test (HOHCT), against one of the most popular approaches, the JCCB. The HOHCT approach has been developed as a way to improve performance of the delayed inverse-depth initialization monocular SLAM, a previously developed monocular SLAM algorithm based on parallax estimation. Both HOHCT and JCCB are extensively tested and compared within a delayed inverse-depth initialization monocular SLAM framework, showing the strengths and costs of this proposal.

1. Introduction

The problem (Simultaneous Localization and Mapping) SLAM, also referred as (Concurrent Mapping and Localization) CML, has been known and studied for long in the field of robotics. The main objective of the techniques addressing this problem is solving how to explore and build a map of an unknown environment with a robotic device while localizing the same device within the map and the environment, leading to navigation capabilities in some cases. Thus, SLAM developments are of capital importance in many fields of robotics, as it is one of the key requisites to achieve autonomous robotic navigation unknown environments. There are many techniques and algorithms developed to address this SLAM problem, with some of the implementations aiming at obtaining a good and enough performance to run online within a robotic device, generally of autonomous nature. These solutions usually revolve around the estimation of self-mapped features located through sensors, traditionally highly specialized and costly devices. As such, the most frequently used sensors within the context of SLAM applied to robotic systems navigation include odometers, radars, GPS, and several kinds of range finders, such as laser, sonar, and infrared-based devices [1, 2].

While all these sensors have advantages and produce reliable readings from the environment which enable obtaining precise implementations of SLAM algorithms, they have also several drawbacks. Many of them require complex hardware with powerful computational capabilities, thus making them unsuitable for deployment into small robotic devices with limited performance, while others are limited to 2-D mapping of the environment. Several sensors provide environment data hard to process by data association algorithms, requiring a lot of computational effort or having problems with map representations, producing complex models. Besides, these kinds of sensors are normally of rather expensive nature to be considered suitable to wide deployment and utilization. Meanwhile, consumer demand and mobile communications gadget development have pushed industry agents to produce relatively cheap and reliable camera devices. These camera sensors have resulted plenty accessible and easy to use, thus contributing to the emergence camera-based SLAM works. Another factor helping to popularize the use of a camera as a main sensor device is the diversity of information that can be obtained from processing adequately the data provided. For example, taking advantage of years of developments produced in the computer vision field, the data association

problem is easily treated when dealing with data obtained from camera sensor, while enabling the introduction of vision-based segmentation, tracking, and other capabilities [3–5].

One of the most promising types of camera-based SLAM problem is the monocular SLAM. In the monocular SLAM problem, only a single camera is used as sensory input, normally without the help of any other device. This makes it a completely different approach from other popular camera-based SLAM and navigation techniques, such as those based on stereo vision, time-of-flight (ToF) cameras, or those which combine cameras with other sensors like odometers, range finders, and so forth. The main difference with ToF cameras and stereo vision approaches is that monocular SLAM is unable to reliably know the depth to any given point captured in an image, at least considering the image alone and isolated, and thus this problem must be addressed. Still, stereo vision is only able to obtain depth estimation only through processing of interest points or features in the two images it receives simultaneously, unlike ToF cameras, which deliver the depth estimation in real time with the images. Some of the other variants which include additional sensors employ these sensors to deal with that, treating besides the problem of knowing how the robotic device is moving along its trajectory, thus getting odometry. Then, it is worth noting how in this context monocular SLAM is one of the most difficult variants of vision-based SLAM, specially the 6 degrees of freedom (DOFs) case without any other input to know range and odometry.

So, the first issues to address in 6-DOF monocular SLAM is the inability of camera sensor to provide depth information in an image, as only bearing data are provided by the sensor. This problem has several solutions in the structure-from-motion (SFM) field [6, 7], being some of them closely related to monocular SLAM. Nevertheless, many of them rely on global nonlinear optimization and several batch techniques unsuitable for SLAM, especially if the aim is reaching a good performance and on-line functionality. Thus, this issue is generally addressed through the inverse-depth (I-D) initialization technique [8], which initially assigns a heuristic value to the depth of image elements. Another important issue to address is the robotic camera motion odometry. The previously mentioned SFM techniques can compute precisely the trajectory of the camera, but these data are generally obtained as part of batch processing of the whole data, thus being unusable. This problem, considered already solved for SLAM, is of great importance as being described as the visual odometry problem [9] and therefore it constitutes a separate problem because SLAM tracks features in the order of tens per frame while visual odometry deals normally with hundreds of features, and several works address to integrate them [10]. By contrast, some other relevant works on Monocular SLAM rely in the utilization of additional sensors to address the problem, as Strelow and Sanjiv in [11], who propose mixing inertial sensors into a camera-based iterated Extended kalman filter (IEKF). Several important works use different estimation techniques, as particle filters (PFs) in Kwok and Dissanayake [12, 13]. Still, many of the most notable works are based on the well-known EKF; both Davison et al. [14] probed the feasibility of real-time monocular SLAM within EKF, and Montiel et al. [8] developed the I-D Initialization within an EKF framework, as the classic original solution to SLAM [15].

The data association problem is relatively simple to solve within a monocular SLAM context: addressed explicitly by matching detected features on different frames with computer vision approaches, or implicitly resolved by the utilization of an active search technique to produce matches to known features on a new image. Still, this data association requires a mechanism to validate the produced data. Traditionally, the validation was performed based on single data association statistical matching, inherited from the computer vision background. But eventually, the validation problem acquired relevance, thus prompting the introduction of batch validation. One of the most usual validation methods is the Joint Compatibility Branch and Bound technique, JCBB [16]. The JCBB methodology is considered a strong batch validation technique, but the algorithm has a worst-case exponential cost. This method shows great results within the context of undelayed depth feature initialization, as it allows ignoring those matches deemed incompatible with the rest of data association pairs. Another widely known batch validation technique is the Combined Constraint Data Association (CCDA) by Bailey [17], based upon graphs instead of trees. This technique's strengths reside in the ability to test batch validation without knowing the device pose robustly in cluttered environments. In a more recent work [18], an approach based on cost functions was presented, introducing decision theory principles. Latest trends include the utilization of random-sample-consensus (RANSAC-) based techniques [19]. These, while being able to exploit many of the strengths of RANSAC, have a potentially unbound computational cost in the form of a high number of RANSAC hypotheses tested to generate models trying to find a good fit. This characteristic has been dealt with the introduction of restrictions to movements, like [20, 21], but as mentioned on [22], the less restricted models offer better estimations of movement. In [22], a fully integrated combination of EKF and RANSAC, like the one proposed on [23], is used, with hypothesis size limited to 1 point.

This paper presents a new batch validation technique for active search-based monocular SLAM, the highest order hypothesis compatibility test, HOHCT, within the context of the delayed I-D initialization approach [24, 25]. As such, a description of the delayed I-D initialization technique is introduced. The relevance of batch validation is studied, with emphasis in the JCBB, the default methodology on I-D initialization SLAM, to provide a ground of reference to properly evaluate HOHTC. The proposed HOHTC technique is detailed, and experimental results are provided, comparing it against JCBB in terms of search complexity and theoretical computational costs.

2. Monocular SLAM with Delayed I-D Initialization

This following section describes the general procedure for monocular SLAM, with an implementation based upon

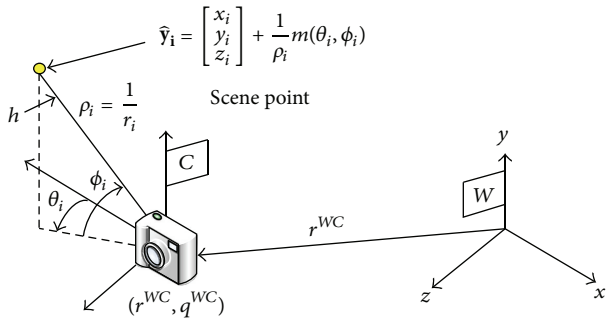

Figure 1: Camera and features in augmented state description.

the delayed inverse-depth initialization. For the sake of simplicity, at each step subscript k represents the initially given estimation or covariance, while $k+1$ represents those same variables from the current step prediction. In terms of coordinate frames, superscripts W and C denote magnitudes expressed in the world reference, and the camera reference respectively, while WC denotes a transformation or vector direction from *World* to *Camera*.

The implemented filtering procedure is based upon the extended kalman filter. while keyframe methods produce more accurate results [26], the same authors conclude that filtering may probe a better option if the processing power is limited and probably deals in a more accurate fashion with the high uncertainty present during initialization.

The monocular SLAM method uses an augmented state data model where data about localization and mapping are maintained within a so-called augmented state vector;

$$\hat{\mathbf{x}} = \left[\hat{\mathbf{x}}_v, \hat{\mathbf{y}}_\mathbf{1}, \ldots, \hat{\mathbf{y}}_\mathbf{n}\right]^T. \tag{1}$$

The first part of this column vector contains a vector $\hat{\mathbf{x}}_v$ that represents a robotic camera device, describing both its pose and movement speeds;

$$\hat{\mathbf{x}}_v = \left[\mathbf{r}^{WC} \quad \mathbf{q}^{WC} \quad \mathbf{v}^W \quad \boldsymbol{\omega}^W\right]^T. \tag{2}$$

The decomposition of vector $\hat{\mathbf{x}}_v$ yields position of the camera optical centre represented by $\mathbf{r}^{WC}$; orientation with respect the navigation frame represented by a unit quaternion $\mathbf{q}^{WC}$; linear and angular velocities are described by $\mathbf{v}^W$ and $\boldsymbol{\omega}^W$. The rest of the augmented state vector describes the map to be estimated, composed by a set of features

$$\hat{\mathbf{y}}_\mathbf{i} = \left[x_i \quad y_i \quad z_i \quad \theta_i \quad \phi_i \quad \rho_i\right]^T, \tag{3}$$

each of them represented by a vector which models the localization of the point where the feature is expected to be according to the inverse-depth model [8] for feature localization, where

$$\hat{\mathbf{y}}_\mathbf{i} = \left[x_i \quad y_i \quad z_i\right] + \frac{\mathbf{m}\left(\phi_i, \theta_i\right)}{\rho_i}, \tag{4}$$

as seen in Figure 1.

The initialization of the system is used to obtain a metric scale with previous knowledge, being analogous to a well-known and solved problem in computer vision, the PnP (perspective of n-points) problem [27]. This problem tries to find the orientation of a camera with respect an object from a set of points. If the points are coplanar, with 4 points with spatial coordinates $(x_i,\ y_i,\ 0)$, the PnP problem can be solved through a linear system to find a unique solution [19].

At the start of each EKF iteration, predictions for the augmented state and its covariance are computed. In order to predict the state, the unknown velocities and accelerations of the robotic camera need to be modelled. An unconstrained constant-acceleration camera motion prediction model can be described by (5) [28]. Here, $\mathbf{q}((\boldsymbol{\omega}_k^W + \boldsymbol{\zeta}^W)\Delta t)$ is the quaternion defined by the rotation vector $(\boldsymbol{\omega}_k^W + \boldsymbol{\zeta}^W)\Delta t$. An unknown linear and angular velocity, $\mathbf{a}^W$ and $\boldsymbol{\alpha}^W$, is assumed described as Gaussian processes with zero-mean acceleration and known covariance, σ_V and σ_Ω. These assumptions produce impulses of linear and angular velocity, $\mathbf{V}^W = \mathbf{a}^W\Delta t$ and $\Omega^W = \boldsymbol{\alpha}^W\Delta t$. The features in the EKF-SLAM are assumed to be static, and the propagation of uncertainty is performed through the usual Jacobian-based formulation as follows

$$f_v = \begin{bmatrix} \mathbf{r}_{k+1}^{WC} \\ \mathbf{q}_{k+1}^{WC} \\ \mathbf{v}_{k+1}^{W} \\ \boldsymbol{\omega}_{k+1}^{W} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k^{WC} + \left(\mathbf{v}_k^W + \boldsymbol{v}_k^W\right)\Delta t \\ \mathbf{q}_{k+1}^{WC} \times \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \boldsymbol{\zeta}_k^W\right)\Delta t\right) \\ \mathbf{v}_k^W + \boldsymbol{v}_k^W \\ \boldsymbol{\omega}_k^W + \boldsymbol{\zeta}_k^W \end{bmatrix}, \tag{5}$$

$$P_{k+1} = \nabla F_x P_k \nabla {F_x}^T + \nabla F_u Q \nabla {F_u}^T. \tag{6}$$

Once the predicted camera location is known, the image pixels (u_i, v_i) where the known features should appear are predicted for each of them, obtaining the feature prediction $\mathbf{h}_i = (u_i, v_i)$. These coordinates are obtained through an observation model that defines a tracing ray expressed in camera frame coordinates as

$$\mathbf{I}^\mathrm{C} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = R^{CW}\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right) - r^{WC}\right), \tag{7}$$

where $\mathbf{I}^\mathrm{C}$ is observed by the camera through its projection in the image. R^{CW} is the transformation matrix from the global reference frame to the camera reference frame.

At each iteration of the EKF, a new image from a sequence is processed, searching matches for the predicted features through and active search algorithm. This algorithm defines a search area around each predicted feature in the image and uses a cross-correlation technique [29] to determine the point where the feature predicted is best matched. After obtaining the feature predictions, these are double checked, through batch validation of the data associations produced. This process will be further described in detail in the next section. Once this process is done, the remaining features are used to perform a standard EKF-SLAM update, computing

innovation $\mathbf{g}$ (10), Kalman gain W (11), covariance S (12), and the updates for state (8) and covariance (9) as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k+1} + W\mathbf{g}, \tag{8}$$

$$P_k = P_{k+1} - WS_iW^T, \tag{9}$$

$$\mathbf{g} = \mathbf{obs}_i - \mathbf{h}_i, \tag{10}$$

$$W = P_{k+1}\nabla H_i^T S_i^{-1}, \tag{11}$$

$$S_i = \nabla H_i P_{k+1} \nabla H_i^T + R_{uv}. \tag{12}$$

The initialization of new features into the filter is performed through parallax-based delayed I-D, presented first in [24], replacing the widely known approach of undelayed I-D [8]. The detection and tracking of features is based on searching points of interest through Harris detector, which are tracked by an active search technique that matches features using cross correlation thresholds. This approach is described in detail in [24, 25].

3. Batch Validation

The matching methodology described in the previous section uses an active search technique to address the problem of data association. So, for each of the predicted landmark features obtained during the prediction phase, an "image observed feature" is found. This yields a set of pairs, each one composed of a predicted landmark and its matching feature in image. Finding a correct pairs list is usually a critical problem in any EKF-based SLAM system, as there are many factors that may introduce errors. These data association errors may even not be incorrectly matched; a moving object can be correctly matched, but can give landmark information which disrupts the map, as this "fake" landmark is not static. Other errors may arise when dealing with ambiguous textures and features on the mapped environment. Thus, the objective of a batch validation test is to reject those data association pairs found that can be considered erroneous.

3.1. Joint Compatibility Branch and Bound. In the context of classical approach to I-D initialization monocular SLAM, the undelayed I-D technique Joint Compatibility Branch and Bound (JCBB) [8] has probably been the most used batch validation methodology. This test is based on the notion of Joint Compatibility [16] and its evaluation for different data association hypotheses. A data association hypothesis is a subset of the pairs set produced by the active search technique; so, the validation test will consider all the pairs on the set "jointly compatible" or consistent, thus valid, or inconsistent as a whole. Evaluation of the compatibility of a hypothesis is based upon the computation of a quality metric and comparing its scored value against a statistical threshold.

The quality metric used is the Mahalanobis distance, of the hypothesis innovation. This value is estimated through (13) and tested against a value from the Chi-squared distribution:

$$D_H^2 = \mathbf{g}_H^T S_{iH}^{-1} \mathbf{g}_H \leq \chi_{d,\tau}^2, \tag{13}$$

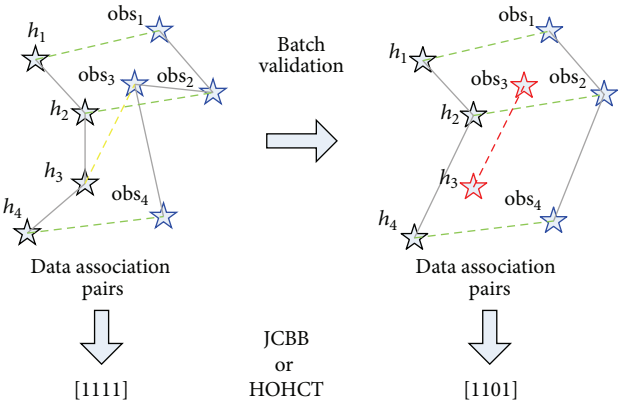

Figure 2: Batch validation: joint compatibility illustrated.

where $\chi_{d,\tau}^2$ is the Chi-squared distribution with a default confidence level of τ, normally set at 0.95, and d is the number of data association pairs accepted into the hypothesis. The distance itself is estimated from $\mathbf{g}_H$ and S_{iH}, which are the innovation and innovation covariance for the hypothesis, respectively, computed as in the update and matching steps of EKF, in (11) and (12). As not all the data association pairs are taken into account in each hypothesis, $\mathbf{g}_H$ and S_{iH} will not be taken completely to obtain the Mahalanobis distance, only those rows related to the considered pair, without necessity of fully computing $\mathbf{g}_H$ and S_{iH} again.

The mentioned hypothesis can be represented as an array of Boolean values, as shown in Figure 2, where each found pair is accepted (true) or rejected (false). Thus, the JCBB uses a purely recursive algorithm to make sure that it finds the best hypothesis [30], requiring it to be compatible with maximal number of pairs and lowest Mahalanobis distance. The algorithm makes a branch and bound search on a binary tree to build the Boolean vector representing the hypothesis, being order independent; it will try all the hypotheses even after a jointly compatible one has been found to guarantee the optimality of the given result. Because of this uninformed, unordered exhaustive search, the algorithm has a potentially exponential cost, with no mechanism to control the growth or keeping it low, and estimates the Mahalanobis distance within each node. This is partly mitigated by several optimizations that reduce the worst costs of the algorithms [31] such as matrix inversions, and exploiting the nature of the Mahalanobis distance to cut as early as possible bad branches of the tree.

3.2. Highest Order Hypothesis Compatibility Test. The JCBB has shown good results within the context of undelayed I-D initialization monocular SLAM but became rather inefficient within the context of the delayed I-D SLAM approach. It is worth noting that there are some key differences between the undelayed I-D and the delayed I-D SLAM techniques; while the undelayed approach tries to initialize a good number of features as landmarks as soon as possible with a heuristic value for depth representation, the delayed approach probably will contain less features, but with greater accuracy

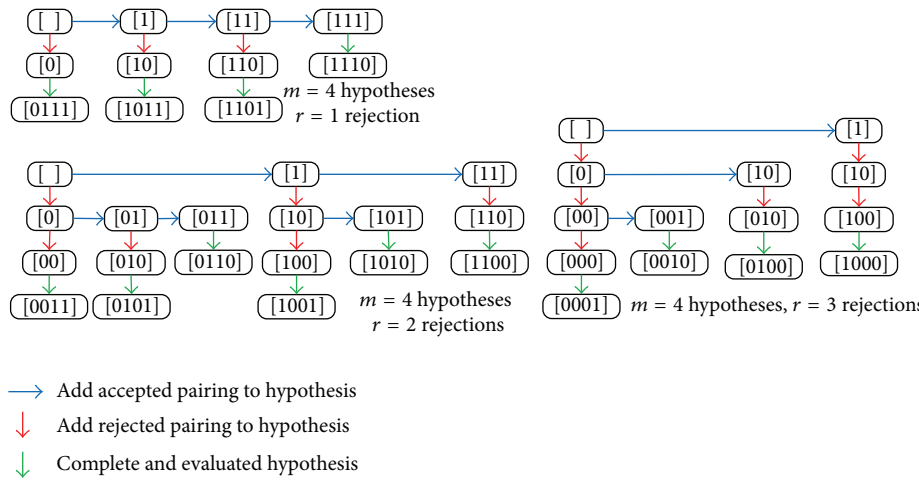

Figure 3: Example with $m = 4$ with increasing number of pairs to be rejected generating different pseudobinary.

and with a good estimation of the depth value before the initialization; the depth is obtained by parallax (as detailed earlier). This makes the delayed approach more expensive in terms of computational cost per feature, but as it holds more accurate information, it requires less mapped features initialized to work, thus achieving better performance. Besides, the appearance of instances is relatively rare where a test is failed by more than one data association pair because of having most of the time good estimations; so, the binary search tree is built almost completely frequently.

So, accounting for DI-D initialization, a new batch validation technique is derived, the highest order hypothesis compatibility test (HOHCT). This new technique uses the same joint compatibility notion, while implementing an ordered informed search through a hybrid recursive and iterative algorithm, which exploits the fact that delayed initialization implementations for monocular SLAM are generally robust to data association errors. This robustness is provided by a series of tests and conditions to be passed by candidate features to be considered landmarks, hence, the initialization delay. In the context of the implemented delayed initialization [24, 25], a feature is required to be tracked correctly within a minimal number of frames achieving a parallax value greater than a minimum α_{min} to guarantee an accurate enough depth estimation, as detailed earlier. Thus the weakest features, generally the most prone to data association errors, are rejected from the start. Still, data association errors can be incurred sparsely; so, the technique works on the optimistic hypothesis that most of the time the number of incorrect data association pairs will be low.

So, initially, the test checks the optimistic hypothesis taking all the pairs m, and failing it, it starts an iterative process to exclude some of them, as shown in Algorithm 1. At each iteration this process will perform the test searching for all the hypotheses which have an exact number of pairs, ignoring as much association pairs as the number of times the test has failed, i. So, after the initial fail $i = 1$, given m data pairs, the hypotheses tested would include only those containing exactly $m - 1$ data pairs. As the number of test fails i increases, the test will be repeated, searching only hypotheses which include $m-i$ data pairs, thus avoiding repetition of previously tested hypotheses.

The algorithm to perform each of the searches mixes both recursive and iterative steps to build an n-ary tree which essentially works as a binary tree but allows skipping exploration of nodes (see Figure 3). The iterative steps will add accepted pairs into the hypothesis (noted as "1"), and the recursive steps will introduce rejection of pairs (noted as "0"). Thus, in the end, the search is performed on a subtree of the hypothetical binary search tree, and the compatibility test, and so the Mahalanobis distance estimation, is only performed on the leaf nodes. By contrast, JCBB evaluates each node of the tree to know if it should cut the branch; so, the Mahalanobis distance estimation is computed an exponentially growing number of times. Note also how the sparse error conditions found in the DI-D initialization SLAM, where the ordered search will have normally linear cost with the number of landmarks matched, with exceptional cases achieving cubic cost over some frames, still very far from the exponential cost that binary tree recursion could suppose over the whole number of landmarks matched, as JCBB.

4. Experimental Results

A series of tests were performed to test the effectiveness and efficiency of the batch validation technique introduced in the context of SLAM delayed I-D initialization. These experiments are focused deliberately into the initialization of the estimation of a trajectory and the initial steps, where the delayed I-D differs more from other techniques. As discussed in [26], a SLAM framework reliable locally can be scaled up with the introduction of submapping or several other techniques, being the loop-closing problem simpler to deal as the proposed technique provides a relatively accurate a metric scale estimation.

Function (h_i, obs_i, S_i, ∇H_i) := **HOHCT-test** (h_i, obs_i, ∇H_i, S_i)
Input:
obs_i matching observations found
h_i features observation prediction
S_i innovation covariance matrix
∇H_i observation Jacobian
Output:
obs_i matching observations found
h_i features observation prediction
S_i innovation covariance matrix
∇H_i observation Jacobian

begin
m := Number of Matches in obs_i
hyp := $[1]^m$ // *Grab all matches*
  if ~JointCompatible (hyp, h_i, obs_i, ∇H_i, S_i) **then**
    $i := 1$
      while $i < m$ **do** // *Hypothesis reducer loop*
       (hyp, $d2$) := **HOHCT-Rec** (m, θ, [ ], i, h_i, obs_i, ∇H_i, S_i)
        if JointCompatible (hyp, h_i, obs_i,∇H_i, S_i) **then**
          $i := m$
        else
          $i := i + 1$
          end if
        end while
    remove incompatible pairings from h_i and obs_i
    update jacobian ∇H_i and matrix S_i
  end if
return (h_i, obs_i, S_i, ∇H_i)

Function (hyp_b, $d2_b$) := **HOHCT-Rec** (m, m_{hyp}, hyp_s,r_m, h_i, obs_i, ∇H_i, S_i)
Input:
m size of full hypothesis
m_{hyp} size previously formed hypothesis
hyp_s hypothesis built through recursion
r_m matches yet to remove
Output:
hyp_b best Hypothesis found from hyp_s
$d2_b$ best Mahalanobis distance

begin
if ($r_m = \theta$) or ($m = m_{\text{hyp}}$) **then**
  $\text{hyp}_b := [m_{\text{hyp}}[1]^{m - m_{\text{hyp}}}]$
  $d2_b$:= **Mahalanobis** (h_i, obs_i,∇H_i, S_i)
else
  $\text{hyp}_b := [\text{hyp}_s[1]^{m - m_{\text{hyp}}}]$
  $d2_b$:= **Mahalanobis** (h_i, obs_i, ∇H_i, S_i)
    for $r := (m_{\text{hyp}+1}) : (m - r_m + 1)$ **do**
      (h, d) := **HOHCT-Rec** (m, $m_{\text{hyp}} + 1$, [$\text{hyp}_s\theta$], $r_m - 1$, h_i, obs_i, ∇H_i, S_i)
        If ($d < d2_b$) **then**
          $d2_b := d$
          $\text{hyp}_b := h$
        end if
        $\text{hyp}_s := [\text{hyp}_s\ 1]$
        $m_{\text{hyp}} := m_{\text{hyp}} + 1$
    end for
end if
return (hyp_b, $d2_b$)

Algorithm 1: Pseudocode for HOHCT test and HOHCT hybrid recursive/iterative search.

Figure 4: Environment used to capture sequences with known ground truth for trajectory.

For the experiments, a set of twenty short video sequences were acquired with a low cost camera. The SLAM technique studied has been implemented in C++, developing it with both batch validation techniques: the JCBB as reference, and the presented HOHCT. These implementations were run offline with the acquired sequences as input, with algorithm parameters set to the following values: variances for linear and angular velocity, respectively, $\sigma_V = 4\,(\mathrm{m/s})^2$, $\sigma_\Omega = 4\,(\mathrm{m/s})^2$, noise variances $\sigma_u = \sigma_v = 1$ pixel, minimum baseline $b_{\min} = 15$ cm, and minimum parallax angle $\alpha_{\min} = 5^\circ$. The default confidence level for the Chi-squared distribution in the HOHCT was set to $\tau = 0.95$. The policy employed with found jointly incompatible data association pairs was to eliminate the related feature as soon as possible, both from the map and from the candidate features database. As in order to be mapped again, the feature would require being correctly initialized after being tracked once more and achieving enough parallax, and this allowed rescuing good landmarks with strong features that just failed due to a nonpermanent disruption.

The videos were captured with a Logitech C920 HD camera. This low cost camera device has a USB interface and wide angle lens. Although it is capable of acquiring HD colour video, the experiments were run capturing grey level video sequences in a reduced resolution of 424 × 240 pixels. The frame rate of the camera was 15 frames per second (fps), as most low cost USB capturing devices. The environment used to acquire the video sequence was the Vision and Intelligent Systems laboratory and its surrounding corridors. Inside the laboratory, most of the experimental videos were captured slowly sliding the camera over a rail guide and tables providing and approximated ground truth reference, as seen in Figure 4. The duration of the different sequences ran from 30 seconds to 1 minute 10 seconds (450 to 1050 frames), varying duration accordingly to the length of the different trajectories and the speed at which they were traversed, thus proving the local reliability [26] of the SLAM framework proposed.

4.1. Mapping and Localization with Batch Validation. Figure 5 shows the different maps produced by the monocular SLAM technique for two sample video sequences moving along different trajectories. Trajectory 1 starts with a U-turn around a table with several objects (the cluttered zone on the centre of the map) and continues along a straight line of three meters. Note how map (a1) (Figure 5 upper left plot) displays a blue trajectory which follows approximately the described path, while on map (b1) (Figure 5 upper right plot) there is a clear drift in orientation, making a more open turn. Besides, once the turn is complete, the straight part of the trajectory is clearly too long on map (b1), probably exceeding the really travelled distance by one-third in this segment (from 3 m to about 4 m).

Maps for trajectory 2 (Figure 5 lower left plot and Figure 5 lower right plot) show similar results. Here the travelled path made an almost full turn around a cluttered table. The map with data association applied shows an almost closing map. As the current implementation of the SLAM approach used does not incorporate any loop closing technique, the results must be considered very solid. At the same time, the same procedure without batch validation techniques applied can yield clearly drifted results.

Therefore, results shown in maps in Figure 5 reveal the importance and impact of incorporating a data association validation technique in the context of monocular SLAM. As the data validation rejects erroneous and weak matching features, it helps to reduce the drift, and in many cases, it keeps the EKF from losing convergence capabilities.

The target of a batch validation technique can be clearly seen in Figures 6 and 7, which illustrate some examples of incompatible data associations. The star-shaped marks envelope the area where the point matched to the landmark in an incompatible data association lies. In Figure 6, an incorrect match is produced due the pattern repetition, while in Figure7, an artificial landmark emerges where the wire and the box edge intersect. Although the incompatibilities look small, the accumulated effect of the error induced over several frames could produce intense dampening effect on map estimation.

4.2. Computational Costs of Batch Validation. Although it has been shown that introducing a batch validation technique greatly improves the results of the proposed monocular SLAM implementation, it comes with a heavy computational cost. The main reason to develop the HOHCT technique was the heavy burden that supposed the introduction of JCBB as batch validation technique. It is worth noting that given the differences in the approach to exploring the data association hypothesis space, the computational cost in terms of expanded leaf nodes can be described by different equations for each technique.

The JCBB essentially explores the majority of a binary tree, ignoring the number of the nodes correspondent to the subtree containing the hypotheses that would have derived from jointly incompatible hypotheses. The computational cost h_n^{JCBB} of such a search in terms of evaluated leaf nodes can be approximated as

$$h_n^{\mathrm{JCBB}} = 2^n - \sum_{i=1}^{r} 2^{n-i}, \tag{14}$$

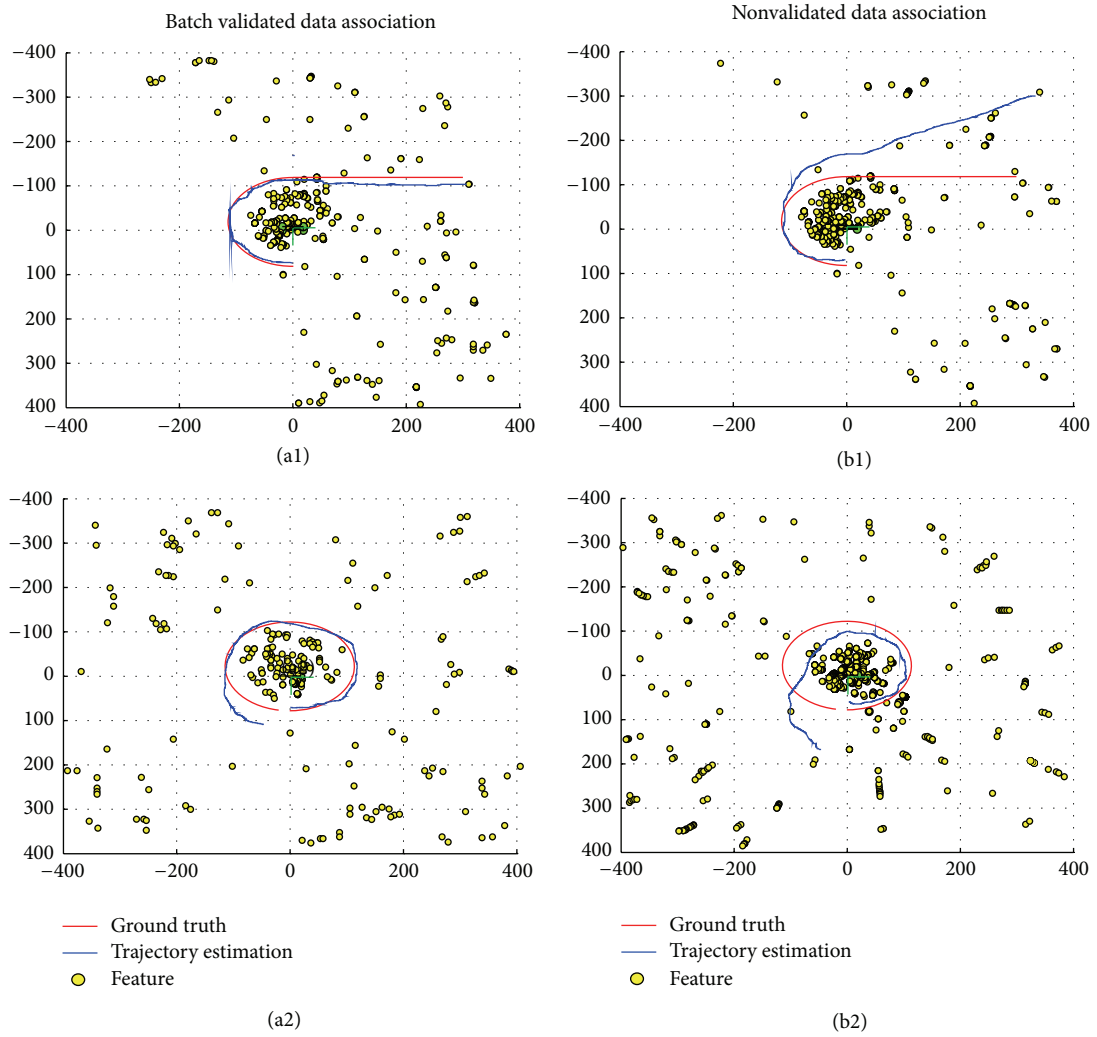

Figure 5: Results of two trajectories, 1 and 2, with HOHCT applied: (a1) and (a2) and without it: (b1) and (b2).

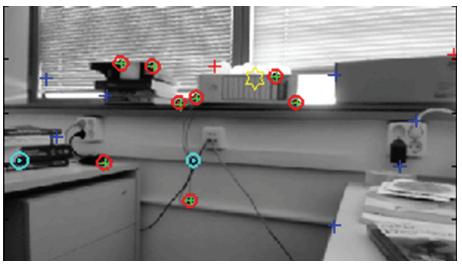

Figure 6: Incompatibility due to repeated design.

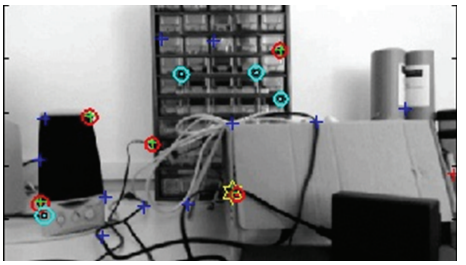

Figure 7: Incompatibility found at composite landmark.

where n and r are the number of data association pairs and the number of pairs to be rejected at the optimal hypothesis, respectively. As the number of rejected data pairs r is usually low, h_n^{JCBB} grows towards exponential cost rather easily, because the main term keeps being of exponential nature. Besides, the Mahalanobis distance is evaluated at each node of the tree (including nonleaf nodes), thus making the process of traversing the tree even more cumbersome.

Meanwhile, the application of the HOHCT validation has a computational cost h_n^{HOHCT}, described in terms of number of leaf nodes evaluated in the pseudobinary tree as

$$h_n^{\text{HOHCT}} = \sum_{i=1}^{r} n^i, \tag{15}$$

where n is the number of pairs observed and r the quantity of these pairs deemed incompatible. See how r limits both

Table 1: Average and total experimentations lengths.

Metric	Average	Total
Frames	857.53	17150
Seconds	57.17	1143.5
JCBB/HOHCT*	125.54	2051

*Searches due to joint incompatibilities rejecting data association pairs.

Table 2: Average nodes explored at each experiment *(full sequence); for each frame and for JCBB/HOHCT search.

	JCBB	HOHCT
Total nodes*	3845478.1	22451.2
Nodes/frame	4484.36	26.18
Nodes/search	37575.01	219.37

the iterative nature of sum and exponential term growth; so a low r value limits greatly the cost of this search. To better understand the implications of these cost equations, several statistics were computed with the data collected from experimentation with the set of video sequences.

The total durations, frames, and executions of the JCBB/HOHCT searches are shown in Table 1. On average, the relevant batch validation algorithm was performed in less than 1/7th of the frames. It must be noted how most of the incompatibilities emerged towards the end of the trajectories when drift is already noticeable. Thus, the number of batch validation searches performed could still be probably reduced further with some drift reducing technique. The number of nodes explored on average per video sequence by the different techniques is shown in Table 2, accounting also for the rate against the number of frames and searches performed. The computed nodes include all nodes built during tree exploration.

This is done because both JCBB and HOHCT compute Mahalanobis distance at each node; thus, this is a better estimation of the real cost of the algorithms that just computing the number of leaf nodes to reach.

On average, the relevant batch validation algorithm was performed in less than 1/7th of the frames. It must be noted how most of the incompatibilities emerged towards the end of the trajectories when drift is already noticeable. Thus, the number of batch validation searches performed could still be probably reduced further with some drift reducing technique. The number of nodes explored on average per video sequence by the different techniques is shown in Table 2, accounting also for the rate against the number of frames and searches performed. The computed nodes include all nodes built during tree exploration. This is done because both JCBB and HOHCT compute Mahalanobis distance at each node; thus, this is a better estimation of the real cost of the algorithms that just computing the number of leaf nodes to reach. The difference in the order of computational cost on average is of three orders of magnitude; while an average HOHCT search explored ~22 nodes, the JCBB explored over thirty thousand. This can be comprehended observing the number of features n considered each time and the number of data pair rejections r, being the two main factors leading

Table 3: Average data association pairs present and rejected by each JCBB/HOHCT search produced, n and r, respectively, in (14) and (15).

Average data association pairs at each search (n)	12.08
Average pairings rejected at each search (r)	1.401

Table 4: Average cases of data pair tuples rejected in a single batch validation search, on the 102.34 average searches during experimental sequences and in %.

Pairs rejected per search	On sequence	%
1 pair incompatible	90.8	72.35
2 pairs incompatible	22.6	18.08
3 pairs incompatible	8.7	6.93
4 pairs incompatible	3.2	2.54
5 pairs incompatible	0.2	0.15

Table 5: Average computation time per frame for the DI-D monocular SLAM with JCBB and HOHCT validations.

	Average time per frame (ms)	Standard deviation
JCBB	52.73	19.20
HOHCT	24.31	2.62

the complexity (Table 3). Consequently, for the average case, the low number of pairs rejected on average when finding a jointly compatible hypothesis makes the cost for the HOHCT almost linear with respect the number of data association pairs, while in the case of the JCBB the cost is still dominated by an exponential value (though rather low). This low average number is obtained from a really low counting of data pairs deemed incompatible at each search; see Table 4.

So, it is worth noting that most of the cases each HOHCT/JCBB search had to reject only a pair, with linear cost, with a chance of less than a fifth to have to reject two pairs. As the cost would grow, the chances are reduced, with only one incidence of a hypothesis search requiring rejection of up 5 pairs for each video on average. Note how in fact the cases are concentrated on a subset of video sequences representing worst case scenarios, with difficult conditions. Still, with an average number of data pairs of 12.08 at each search, worst case costs for HOHCT would be marginally worst computationally than the average case using JCBB.

This difference in terms of nodes to explore was observed in experimental computational times. Table 5 shows the results obtained in terms of time per frame in milliseconds. Considering that for the experiments slow sequences at 15 fps were used, both approaches reached average real-time performance. It must be noted that the sequences were captured manually but with an artificial ground truth, easing the feature tracking and matching processes, thus improving the performance. Anyway, the average computational time per frame of the DI-D SLAM with JCBB validation doubled the total time of the DI-D SLAM with HOHCT and presented a much greater standard deviation. This was due to the fact that worst cases on JCBB, as shown in (14), are of near exponential nature, but not as time consuming as one

might expect. This is because JCBB can compute matrix inversions incrementally in optimal ways [31]. This technique reduces the matrix inversion cost from cubic to quadratic for iteratively growing matrix (such as those found at the JCBB) after the first matrix is inverted. This optimization hugely reduces the JCBB computational cost, but not enough to fully compensate the size of the search space explored.

On the other hand, the average time for the DI-D SLAM with HOHCT was well constrained in the studied cases, enough to achieve real-time operation at 15 fps and probably 25 fps, but the performance would be doubtful at 30 fps. This computational cost could be reduced further as the HOHCT, this implementation computes the Mahalanobis distance through full matrix inversion at each evaluation, and there is margin for introducing optimizations, similar to the iterative matrix inversion. Besides, the parallax feature estimation framework can probably be optimized further in its C++ implementation.

5. Conclusions

A batch validation technique to solve the problem of data association validation in Monocular SLAM [25], the HOHCT, has been detailed and evaluated against a well-known approach, the JCBB. The considered monocular SLAM technique is the delayed I-D initialization, which presents a set of features exploited to introduce a strong but efficient data association validation.

The main characteristic is that landmarks are only introduced into the extended kalman filter once the depth estimation is accurate enough, finding this estimation through parallax effect. This introduces a slight computational burden on the algorithm, vastly overcome by the fact that as the information about landmarks present at the map and filter is more accurate; so, the filter can proceed with fewer landmarks mapped than in the undelayed approach. Although mapped landmarks are highly precise, data association gating technique is still needed to treat with multiple disruptions that mainly arise from incorrect or inconsistent matching obtained through an active search. This batch validation technique, the HOHCT, is based on the notion of joint compatibility that performs an analogous search to the JCBB, employing the same statistical evaluation technique but optimized to exploit the undelayed I-D initialization, achieving similar results with lesser computational costs. Both the effectiveness and the efficiency of the HOHCT have been validated with a series of experiments with real data.

The experimental results show how the introduction of the batch validation based on joint compatibility improves the technique resilience to erroneous data association and false features or landmarks, produced by difficult illumination and feature detection errors. The HOHCT costs, while having worst case scenario of exponential cost, just like the JCBB, have been probed to tend to linear case. This tendency to linearity of cost has been probed experimentally, testing the efficiency of both the JCBB and the HOHCT within the considered monocular SLAM technique. HOHCT outperformed easily the JCBB, with a difference of computational costs clearly seen in terms of hypotheses explored and execution times.

Future works expect to produce a lightweight SLAM framework with two main lines of optimization: reducing the computational costs of the HOHCT and refining the DI-D SLAM framework implementation, especially the parallax estimation and features tracking process. In case of achieving better performance and reliability thresholds, the framework would be deployed into several small autonomous robotic devices. This will require the integration of loop-closing detection and a methodology to deal with longer sequences, both problems known to be solved. The resolution of these problems should allow more accurate evaluation of the procedure. Additionally, a deeper study of newer trends should be performed, to provide the necessary background to evaluate the technique and guarantee the fulfillment of requirements.

Variables

Scalars

σ_V: Linear speed variance
σ_Ω: Angular speed variance
x_i, y_i, z_i: Camera coordinates
u_i, v_i: Point coordinates on capture frame
θ_i, ϕ_i: Camera azimuth and elevation
r_i: Real depth to feature
ρ_i: Inverse depth to feature
a^W: Linear speed covariance
α^W: Angular speed covariance
$\alpha_{\min}$: Minimum parallax angle
$b_{\min}$: Minimum baseline distance
D_H^2: Mahalanobis distance for hypothesis H
$\mathrm{X}^2_{d,\tau}$: Chi-squared distribution
h_n^{HOHCT}: Terminal nodes visited by HOHCT algorithm
h_n^{JCBB}: Terminal nodes visited by JCBB algorithm.

Vector

$\hat{\mathbf{x}}$: Augmented state vector
$\hat{\mathbf{x}}_v$: Robot camera state vector
$\mathbf{r}^{WC}$: Camera optical center position quaternion
$\mathbf{q}^{WC}$: Robot camera orientation quaternion
$\mathbf{v}^W$: Robot camera linear speed vector
$\boldsymbol{\omega}^W$: Robot camera angular speed vector
$\hat{\mathbf{y}}_i$: Feature i vector
$\mathbf{f}_v$: Camera motion prediction model
$\boldsymbol{\upsilon}^W$: Linear speed pulse assumed by model
$\boldsymbol{\zeta}^W$: Angular speed pulse assumed by model
$\mathbf{l}^C$: Ray tracing for feature prediction
$\mathbf{g}$: Kalman innovation vector
obs: Features measurement vector
$\mathbf{h}_i$: Predicted features vector
$\mathbf{g}_H$: Kalman innovation for hypothesis.

Matrix

∇F_u: Process noise Jacobian matrix
∇H_i: Measurement model Jacobian matrix
S: Innovation covariance matrix
R_{uv}: Measurement noise matrix
W: Kalman gain matrix
P: Augmented state covariance matrix
∇F_x: Stated prediction model Jacobian matrix
S_{iH}: innovation covariance matrix for hypothesis H.

References

[1] F. Auat, C. De la Cruz, R. Carelli, and T. Bastos, "Navegación autónoma asistida basada en SLAM para una silla de ruedas robotizada en entornos restringidos," *Revista Iberoamericana de Automática e Informática*, vol. 8, pp. 81–92, 2011 (Spanish).

[2] R. Vázquez-Martín, P. Núñez, A. Bandera, and F. Sandoval, "Curvature-based environment description for robot navigation using laser range sensors," *Sensors*, vol. 9, no. 8, pp. 5894–5918, 2009.

[3] C. C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Proceedings of the IEEE International Conference on Robotics adn Automation*, vol. 3, pp. 2918–2924, May 2002.

[4] C. H. Hsiao and C. C. Wang, "Achieving undelayed initialization in monocular SLAM with generalized objects using velocity estimate-based classification," in *Proceedings of the International Conference on Robotics and Automation*, pp. 4060–4066, 2011.

[5] W. Benn and S. Lauria, "Robot navigation control based on monocular images: an image processing algorithm for obstacle avoidance decisions," *Mathematical Problems in Engineering*, vol. 2012, Article ID 240476, 14 pages, 2012.

[6] H. Jin, P. Favaro, and S. Soatto, "A semi-direct approach to structure from motion," *Visual Computer*, vol. 19, no. 6, pp. 377–394, 2003.

[7] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proceedings of the European Conference on Computer Vision*, June 1998.

[8] J. M. M. Montiel, J. Civera, and A. Davison, "Unified inverse depth parameterization for monocular SLAM," in *Proceedings of the Robotics: Science and Systems Conference*, August 2006.

[9] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[10] B. Williams and I. Reid, "On combining visual SLAM and visual odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 3494–3500, May 2010.

[11] S. Strelow and D. Sanjiv, "Online motion estimation from image and inertial measurements," in *Proceedings of the Workshop on Integration of Vision and Inertial Sensors (INERVIS '03)*, 2003.

[12] N. M. Kwok and G. Dissanayake, "Bearing-only SLAM in indoor environments," in *Proceedings of the Australasian Conference on Robotics and Automation*, December 2003.

[13] N. M. Kwok, G. Dissanayake, and Q. P. Ha, "Bearing-only SLAM using a SPRT based gaussian sum filter," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1109–1114, April 2005.

[14] A. Davison, Y. Gonzalez, and N. Kita, "Real-Time 3D SLAM with wide-angle vision"," in *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, July 2004.

[15] R. Smith, R. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, pp. 167–193, 1990.

[16] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transaction on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[17] T. Bailey, *Mobile robot localisation and mapping in extensive outdoor environments [Ph.D. dissertation]*, Australian Centre for Field Robotics, The University of Sydney, 2002.

[18] N. M. Kwok, Q. P. Ha, and G. Fang, "Data association in bearing-only SLAM using a cost function-based approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 4108–4113, April 2007.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[21] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4293–4299, May 2009.

[22] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for EKF-based structure from motion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 3498–3504, October 2009.

[23] A. Vedaldi, H. Tin, P. Favaro, and S. Soatto, "KALMANSAC: robust filtering by consensus," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 633–640, October 2005.

[24] R. Munguia and A. Grau, "Monocular SLAM for visual odometry: a full approach to the delayed inverse-depth feature initialization method," *Mathematical Problems in Engineering*, vol. 2012, Article ID 676385, 26 pages, 2012.

[25] R. Munguia and A. Grau, "Delayed inverse depth monocular SLAM," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, July 2008.

[26] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: why filter?" in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 2657–2664, May 2010.

[27] C. Chatterjee and V. P. Roychowdhury, "Algorithms for coplanar camera calibration," *Machine Vision and Applications*, vol. 12, no. 2, pp. 84–97, 2000.

[28] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "MFm: 3-D motion from 2-D motion causally integrated over time," in *Proceedings of the European Conference on Computer Vision*, June 2002.

[29] A. Davison and D. Murray, "Mobile robot localisation using active vision," in *Proceedings of the European Conference on Computer Vision*, June 1998.

[30] L. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardos, "Mapping large loops with a single hand-held camera," in *Proceedings of the Robotics: Science and Systems*, June 2007.

[31] A. David, *Matrix Algebra from a Statistician's Perspective*, Springer, New York, NY, USA, 1998.