INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

A Highest Order Hypothesis Compatibility Test for Monocular SLAM

Regular Paper

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

1 Automatic Control Dept, Technical University of Catalonia, Barcelona, Spain
2 Department of Computer Science, CUCEI, Universidad de Guadalajara, México
* Corresponding author E-mail: antoni.grau@upc.edu

Received 15 Jun 2012; Accepted 7 Jun 2013

DOI: 10.5772/56737

Abstract Simultaneous Location and Mapping (SLAM) is a key problem to solve in order to build truly autonomous mobile robots. SLAM with a unique camera, or monocular SLAM, is probably one of the most complex SLAM variants, based entirely on a bearing-only sensor working over six DOF. The monocular SLAM method developed in this work is based on the Delayed Inverse-Depth (DI-D) Feature Initialization, with the contribution of a new data association batch validation technique, the Highest Order Hypothesis Compatibility Test, HOHCT. The Delayed Inverse-Depth technique is used to initialize new features in the system and defines a single hypothesis for the initial depth of features with the use of a stochastic technique of triangulation. The introduced HOHCT method is based on the evaluation of statistically compatible hypotheses and a search algorithm designed to exploit the strengths of the Delayed Inverse-Depth technique to achieve good performance results. This work presents the HOHCT with a detailed formulation of the monocular DI-D SLAM problem. The performance of the proposed HOHCT is validated with experimental results, in both indoor and outdoor environments, while its costs are compared with other popular approaches.

Keywords Monocular SLAM, Robotics, Only-bearing Sensor

1. Introduction

Simultaneous Localization and Mapping (SLAM), or Concurrent Mapping and Localization (CML), is a well-known and well-studied problem among the members of the robotics community, being one of the most active fields of research over the last years. The SLAM problem concerns how a mobile robot can operate in an a priori unknown environment by means of only on-board sensors to simultaneously build a map of its surroundings using this to track its position. Thus, SLAM is one of the most important problems to solve in robotics in order to build truly autonomous mobile robots.

There are many techniques and algorithms that have been developed in order to address this problem, many of them aiming to be run on-line on a robotic device. Most of these solutions focus on the estimation of self-mapped features located through assorted types of sensors. The most frequently used sensors for SLAM techniques include odometers, radar, GPS and several kinds of range finders such as laser, sonar and infrared-based ones [1][2]. All these sensors have their own advantages, but also several drawbacks have to be considered, such as: the increasing difficultly regarding correspondence or data

association, the limitation to 2D mapping, excessive computational requirements, or being too expensive to be deployed on certain commercial platforms. At the same time, consumer demand has pushed the industry to the mass production of cheap and reliable camera devices at relatively low prices. All these factors have contributed to the appearance of recent works concerning the use of cameras as main sensors. The requirements regarding the accessibility and easiness of the use of the cameras, and the amount and diversity of the data provided by them as sensors, make computer vision an obvious choice for streamlining autonomous robotics. This will lead to robotic platforms with less high-end sensors with complex requirements and being hard to integrate, substituted by off-the-shelf cameras, maintaining their levels of performance and accuracy. The use of a camera as a main sensor for SLAM is of interest given the information that it provides, can serve as the information required for solving the data association problem and do so easily. Research into computer vision is constantly developing techniques to obtain information from images that can be used in visual SLAM. In any case, Monocular SLAM with six degrees of freedom (as presented in this work) remains one of the hardest SLAM variants, as only bearing data is provided by the camera, thus special techniques are needed to obtain information about the depth of a given point in an image.

This problem has several solutions stemming from the structure-from-motion field of research [3][4], being closely related to monocular SLAM. But a great deal of these solutions is based on methods of global nonlinear optimization, best performed offline, making them unsuitable for SLAM necessities.

Some relevant works on monocular SLAM rely on additional sensors, such as Strelow [5], who proposes mixing inertial sensors with a camera into an Iterated Extended Kalman Filter (IEKF). Other works try to employ different estimation techniques, such as Particle Filters (PF), in Kwok and Dissanayake [6][7]. Still, some of the most notable works are based on the well-known EKF: Davison [8] proved the feasibility of real-time monocular SLAM using an EKF; and Montiel [9] developed the Inverse-Depth parameterization, which allows initializing features to be seen by the robot with a heuristically chosen value for depth.

With regards to the monocular SLAM problem, the batch validation problem has been considered only in recent years. A good survey of the techniques for dealing with this can be found in the works of Durrant-Whyte and Bailey [10][11]. The early implementation of SLAM techniques dealt with each data association individually, but this validation strategy frequently leads to unreliable results, as it is easy to have a wrong match due to the texture and geometry of the environment. Eventually batch validation was introduced, considering multiple data associations to be validated at the same time. Currently, several tests and methodologies exist for dealing with the batch validation of data association. One of the most usual validation methods is the Joint Compatibility Branch and Bound technique, JCBB [12]. The JCBB method is considered as a strong batch validation technique, but the algorithm has a worst-case exponential cost, partly mitigated by several optimizations. This method gives great results within the context of undelayed depth feature initialization, as it allows for those matches deemed incompatible to be ignored with the rest of the data association pairs. Another widely known batch validation technique is the Combined Constraint Data Association (CCDA) developed by Bailey [13], based upon graphs instead of trees. The strengths of these techniques reside in the ability to test batch validation without knowing the device pose robustly in a cluttered environment.

Developments within the field of monocular SLAM are having a growing impact on applied robotics, as seen over the last years. Examples of visual SLAM-navigated applications are to be found deployed on land, at sea and in the air, creating new ways of and techniques for autonomous navigation. Though based on stereo vision, in [14] a high density map is reconstructed from a stereo video, leading to a dense cloud map allowing for accurate navigation. For sea depths, a monocular SLAM-based mosaicing technique [15] allows ROVs to reconstruct visual maps of the seabed. Recent developments in Micro Aerial Vehicles (MAV), as presented in [16], introduce monocular SLAM as a way to stabilize and navigate MAVs without external tracking or prior knowledge of the environment. Although this last work is based on a new parallel tracking batch optimization technique [17], it shows the potential of monocular SLAM for autonomous robotics applications. Further developments in filtering SLAM techniques will expand the possibilities of the limited autonomous robotics systems, where filtering monocular SLAM techniques still have an edge over more complex but also computationally expensive approaches [18].

This paper significantly expands on the authors' previous works proposed in [19] and [20]. Those works presented and dealt with several aspects of DI-D initialization, such as parameter adjustment and a comparison with similar approaches, without dealing with the data association validation problem. The aim of this work is to explicitly address the batch validation problem with the proposal of a novel algorithm, the HOHCT, which evaluates the compatibility of a given set of matches, and tries to obtain the largest subset of compatible matches efficiently. These improvements allow developers to start testing the technique deployed in a robotic system in order to

determine its feasibility as an enabling technology with which to decrease the sensory requirements of robotics systems. The following sections deal with the accurate description of the system in terms of its formulation, with emphasis on the novel HOHCT proposed algorithm; after the description of the method, several experimental cases are presented which test the proposal with real data, discussing the results and their implications on the performance of this novel contribution.

2. Monocular SLAM with DI-D initialization

The procedure and formulation of DI-D Monocular SLAM is described in this section. For the sake of simplicity, at each step subscript *k* represents the initially given estimation or covariance, while *k+1* designates those same magnitudes from the current step prediction. In terms of coordinates frames the superscripts *W* and *WC* will denote magnitudes expressed in the world reference and the camera reference respectively. This notation is the same as that in the authors' previous works, where further details of the DI-D initialization are provided [19].

2.1 State and system specification

The EKF SLAM methodology requires that data regarding localization and mapping are maintained within the so-called augmented state vector, $\hat{x}$ (Eq. 1). The first part of this column vector contains a vector $\hat{x}_v$ which represents a robotic camera device, describing both its pose and movement speeds (Eq. 2):

$$\hat{x} = [\hat{x}_v, \hat{y}_1, \hat{y}_2, \ldots \hat{y}_n]^{\mathrm{T}} \tag{1}$$

$$\hat{x}_v = \begin{bmatrix} r^{WC} & q^{WC} & v^{W} & \omega^{W} \end{bmatrix}^{\mathrm{T}} \tag{2}$$

The vector $\hat{x}_v$ can be broken down in the description of the pose and the movements. The position of the camera's optical centre is represented by r^{WC}, while its orientation with respect to the navigation frame is represented by a unit quaternion q^{WC}. Linear and angular velocities are described by v^W and ω^W respectively:

$$r^{WC} = \begin{bmatrix} x_v & y_v & z_v \end{bmatrix}^{\mathrm{T}} \tag{3}$$

$$q^{WC} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^{\mathrm{T}} \tag{4}$$

$$v^{W} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^{\mathrm{T}} \tag{5}$$

$$\omega^{W} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathrm{T}} \tag{6}$$

The map to be found and estimated is composed of a set of features, $\hat{y}_i$, each of which are represented by a vector which models the localization of the point where the feature is expected to be:

$$\hat{y}_i = \begin{bmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{bmatrix}^{\mathrm{T}} \tag{7}$$

The obtained values form the following model for feature localization:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \phi_i) \tag{8}$$

In the model represented in (Eq. 7), x_i, y_i, z_i are the coordinates of the optical centre of the camera where the feature was seen for the first time; and θ_i,ϕ_i, represent azimuth and elevation (in relation to the world reference *W*) for the directional unitary vector $m(\theta_i,\phi_i)$. The point depth r_i is coded by its inverse: $\rho_i=1/r_i$ as quoted in reference [9]. Figure 1 illustrates the camera and features parameterization.

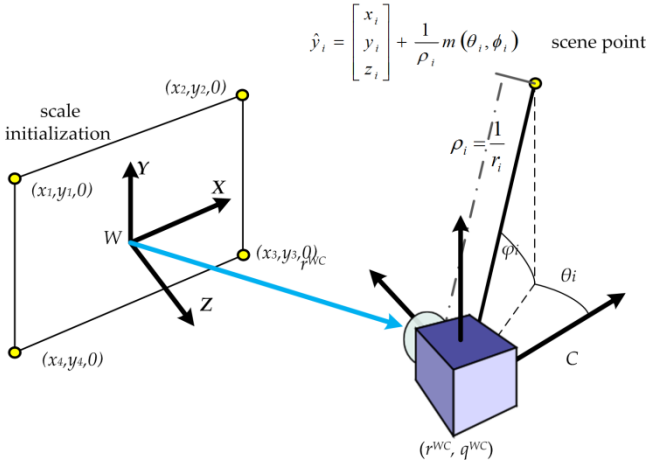

Figure 1. Camera, scene point feature and scale initialization reference in DI-D model.

2.1 Scale and System Initialization

The initialization of a metric scale with previous knowledge is analogous to a well-known and previously solved problem in computer vision, the PnP (perspective of n-points) problem, [21]. This problem tries to find the orientation of a camera with respect to an object from a set of points. If the points are coplanar, with four points with spatial coordinates (x_i, y_i, 0) as seen in Fig. 1, the PnP problem can be solved with a linear system to find a unique solution [22].

At the beginning of the iterative process, the system state $\hat{x}_{ini}$ (Eq. 9) is formed by the camera-state $\hat{x}_v$ and the four initial features used for estimating the extrinsic camera parameters, forming the first augmented state vector $\hat{x}$ (Eq. 1):

$$\hat{x}_{ini} = \begin{bmatrix} r^{WC}{}_{ini} & q^{WC}{}_{ini} & v^{W}{}_{ini} & \omega^{W}{}_{ini} & \hat{y}_1 & \hat{y}_2 & \hat{y}_3 & \hat{y}_4 \end{bmatrix}^{\mathrm{T}} \tag{9}$$

where

$$\begin{matrix} r^{WC}{}_{ini} = t & q^{WC}{}_{ini} = q((R^{CW})^{\mathrm{T}}) \\ v^{W}{}_{ini} = [0_{3\times1}] & \omega^{W}{}_{ini} = [0_{3\times1}] \end{matrix} \tag{10}$$

where t is a translation vector for the position of the camera centre and R^{CW} is the world to camera rotation matrix for the camera orientation as described in [23]. Each initial feature $\hat{y}_i$, for i = (1,..,4), corresponds to each reference point (x_i, y_i, 0), but parameterized as Eq. 7.

2.2 State prediction

An unconstrained constant-acceleration camera motion prediction model can be defined by the following equation (Eq. 3), from [24]:

$$f_v = \begin{bmatrix} r^{WC}_{k+1} \\ q^{WC}_{k+1} \\ v^{W}_{k+1} \\ \omega^{W}_{k+1} \end{bmatrix} = \begin{bmatrix} r^{WC}_{k} + \left(v^{W}_{k} + V^{W}_{k}\right)\Delta t \\ q^{WC}_{k} \times q\left(\left(\omega^{W}_{k} + \Omega^{W}\right)\Delta t\right) \\ v^{W}_{k} + V^{W} \\ \omega^{W}_{k} + \Omega^{W} \end{bmatrix} \tag{11}$$

being $q((\omega^{W}{}_k+\Omega^{W})\Delta t)$ the quaternion defined by the rotation vector $(\omega^{W}{}_k+\Omega^{W})\Delta t$. At every step it is assumed that there is an unknown linear and angular velocity with acceleration zero-mean and known covariance Gaussian processes, a^{W} and α^{W}, producing an impulse of linear and angular velocity: $V^{W}= a^{W}\Delta t$ and $\Omega^{W}= \alpha^{W}\Delta t$.

This model, (Eq. 11), updates the robotic camera part of the state, while the features are assumed to remain static, thus the complete state prediction model is defined as:

$$\hat{x}_{k+1} = \begin{bmatrix} f_v\left(\hat{x}_v\right) \\ \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} \tag{12}$$

The prediction step is completed by propagating the estimation uncertainty through the covariance matrix, Eq. 13, where ∇F_x is the Jacobian of the prediction model and ∇F_u the Jacobian of the process noise.

$$P_{k+1} = \nabla F_x P_k \nabla F_x{}^{\mathrm{T}} + \nabla F_u Q \nabla F_u{}^{\mathrm{T}} \tag{13}$$

2.3 Measurement prediction

The different locations of the camera, along with the location of the already mapped features, are used to predict the feature position h_i. The model observation of a point $\hat{y}_i$ from a camera location defines a ray expressed in the camera frame as:

$$h^c = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = R^{CW}\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} m\left(\theta_i, \phi_i\right) - r^{WC}\right) \tag{14}$$

h^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. The projection is modelled by using a full perspective wide angle camera.

A description of this process is found in Algorithm 1, where ∇H_i denotes the Jacobian derived from the measurement model h_i for each feature predicted (Eq. 14), to be used in the active search technique used for the matching process.

2.4 Features Matching Process

When a feature $\hat{y}_i$ is added to the map, a unique image patch of n-by-n pixel is stored and linked to the feature. To match a feature in the current image frame a patch cross-correlation technique [25] is applied over all the image locations (u_d, v_d) within an elliptical search region (Eq. 15) derived from the innovation matrix Eq. 16:

$$\begin{bmatrix} s_x \\ s_y \end{bmatrix} = \begin{bmatrix} 2n\sqrt{S_{i(1,1)}} \\ 2n\sqrt{S_{i(2,2)}} \end{bmatrix} \tag{15}$$

$$S_i = \nabla H_i P_{k+1} \nabla H_i{}^{\mathrm{T}} + R_{uv} \tag{16}$$

If the score of a pixel location (u_d,v_d), determined by the best cross-correlation between the feature patch and the patches defined by the region of the search, is higher than a given threshold then this pixel location (u_d,v_d) is considered as the current feature measurement z_i.

2.5 Batch Validation of Data Association

In SLAM, the injurious effect of incorrect or incompatible matches is well known. In monocular SLAM systems delayed initialization techniques implicitly prune some weak image features prior to their addition to the stochastic map, e.g., image features produced by fast lighting changes, shining on highly reflective surfaces, or even caused by some dynamic elements in the scene. Nevertheless, the risk of incorrect or incompatible matches could remain due to several factors:

- Incompatibility due to a repeated design.
- Fake composite landmark.
- Incompatibilities produced by reflections on curved surfaces and materials.
- Detection of landmarks running along edges.

The main contribution of this article is to present a novel validation technique that the authors call the *Highest Order Hypothesis Compatibility Test*, (HOHCT). The HOHCT is intended to detect incorrect or incompatible matches, and is explained in detail in section 3.

2.6 Filter Update

With the information obtained from the data association pairs found by the matching process and validated by the HOHCT the filter state and covariance are updated according to Eqs. 17 and 18 respectively:

$$\hat{x}_k = \hat{x}_{k-1} + Wg \tag{17}$$

$$P_k = P_{k-1} - WS_iW^{\mathrm{T}} \tag{18}$$

where the innovation g is:

$$g = z_i - h_i \tag{19}$$

And the Kalman gain W is (Eq. 20):

$$W = P_{k-}\nabla H_i^{\mathrm{T}} S_i^{-1} \tag{20}$$

2.7 Delayed Inverse Depth Initialization of Features

Depth information cannot be obtained in a single measurement when bearing sensors (e.g., a single camera) are used. In this case, in order to incorporate new features into the map, special techniques for feature initialization are needed to enable the use of bearing sensors in SLAM systems.

In this work the Delayed Inverse-Depth (DI-D) Feature Initialization is used to incorporate new features $\hat{y}_{new}$ into the stochastic map. This method implements a stochastic triangulation technique in order to define a hypothesis of an initial depth for the features using a delay (Fig. 2). When cameras are used in real cluttered environments, the delay can be used to efficiently reject weak features, thus initializing only the best features as new landmarks to the map.

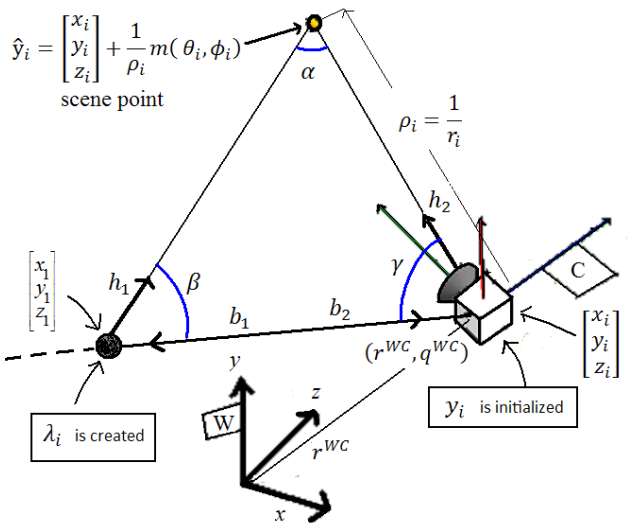

Figure 2. Diagram of parallax α estimation.

The initialization process can be summarized in a few steps, with further details in [19]:

- When a feature is detected for the first time k, feature information, current state $\hat{x}_k$ and covariance P_k are stored. This data is called *candidate point* λ.
- In subsequent instants $k+dt$, the parallax α is estimated using: *i)* the base-line, *ii)* the associated data λ, *iii)* the current state $\hat{x}$.
- If a minimum parallax threshold α_{min} is reached, the candidate point is initialized as a new feature $\hat{y}_{new}$. The new covariance matrix P is fully estimated by the initialization process.

3. Highest Order Hypothesis Compatibility Test, HOHCT

The matching methodology described in the previous section uses an active search technique to address the problem of data association. This problem is usually a critical part of any EKF-based SLAM system, as errors could dampen the convergence of the filter. These data association errors may even not be incorrect matching: 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.

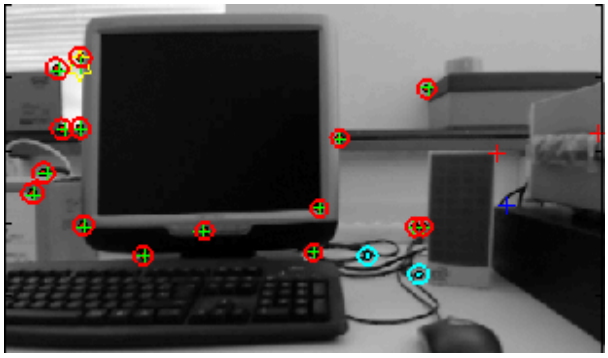

Figure 3. Fake composite landmark (marked by a star) at the intersection of the display border and the blinds.

It is worth noting that previous works based on DI-D did not feature data association validation, but could achieve good results [19][20] due to the resilience to both 'fake' (formed by the superposition of different elements seen from a given perspective) and 'weak' (due to texture, lightning or any other factors that are hard to track in consecutive frames) features. Figure 3 and 4 illustrate examples of both a 'fake' composite feature and 'weak' features, respectively. The fake landmark appears at the intersection of the display edge and the blinds, and as the camera moves, it would slide along the blinds and the display border so as to always be found at the intersection point. The example of weak features (Fig. 4) shows two features (on the desktop PC case) that could be correctly initialized, but later on they proved to be weak due to reflections on the surface.

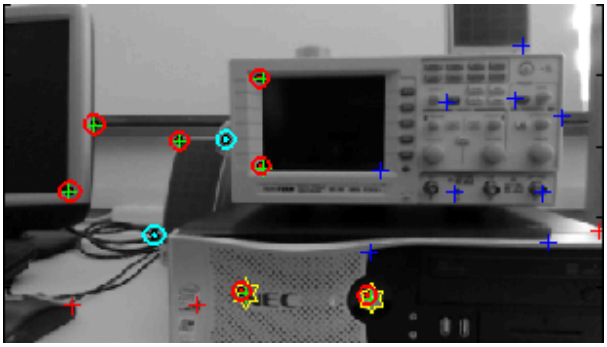

Figure 4. Incompatibilities produced by reflections on materials and curved surfaces, leading to weak features.

DI-D SLAM without data validation could reject many of these errors because candidate features must satisfy a series of tests and conditions before being considered as suitable landmarks to be initialized. In the implemented technique, a feature must be tracked correctly within a minimal number of frames (normally 10 frames) and achieve a parallax value greater than a minimum $\alpha_{\min}$ to guarantee an accurate enough depth estimation, as detailed below. However, errors could be interpreted as features producing wrong data associations, dampening the filter performance. The most common association errors to be initialized are those produced by fake (or composite) features over relatively long distances which are used most frequently, as they tend to be easy to track (thus robust) and observable over long periods.

Thus, accounting for DI-D initialization features, a new technique for batch validation of the data association has been developed, the *Highest Order Hypothesis Compatibility Test*, HOHCT. This new algorithm exploits the fact that delayed initialization implementations for monocular SLAM are generally robust to data association errors, but they still may arise sparsely. Because of this relatively low chance of data association errors, the technique works on the optimistic approach that most of the time the number of incorrect data association pairings will be low. To know if a given data pairing is valid, a batch validation test based on a comparison of a quality metric against a statistical threshold of acceptance is performed. In this test, the data association pairings obtained from the matching process are tested and deemed valid or invalid as part of a set of pairings, being the whole set jointly consistent or 'compatible'.

So, this 'compatibility test' evaluates sets of data parings, known as 'hypotheses'. Each hypothesis is a subset of the data association pairings obtained from the matching process. To determine if a hypothesis is compatible or not, its innovation Mahalanobis distance is estimated (Eq. 21). This distance is used as a quality metric tested against a threshold given by the Chi-squared distribution:

$$D_{\mathrm{H}}^2 = g_{\mathrm{H}}^T S_{i\mathrm{H}}^{-1} g_{\mathrm{H}} \leq \chi_{d,\tau}^2 \qquad (21)$$

where $\chi^2_{d,\tau}$ is the Chi-squared distribution with a default confidence level of τ, and d is the number of data association pairs accepted into the hypothesis. The distance itself is estimated from 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 Eq. 16 and Eq. 19. As not all the data association pairings are taken into account in each hypothesis, g_H and S_{iH} will not be taken completely to obtain the Mahalanobis distance, only those rows related to the considered pairing, without the necessity of fully computing g_H and S_{iH} again.

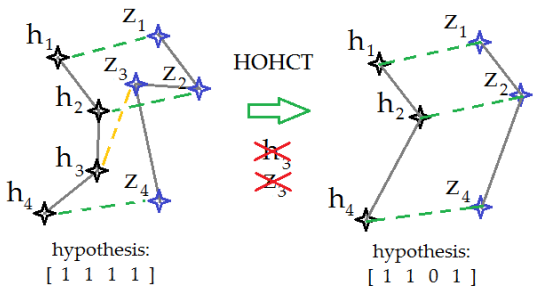

Figure 5. Example of a pair elimination and hypothesis.

As the proposed monocular SLAM performs data association based on an active search strategy, each mapped landmark is associated to an unique feature on the image, so the mentioned 'hypothesis' can be represented as an array of Boolean values the size of the total set of data pairing, N, where each of the pairings is marked as having been considered by the hypothesis (true, 1) or ignored (false, 0). Then, a complete search considering all the possible hypotheses (binary arrays of size N), would be a time consuming effort of exponential cost. Thus, the proposed HOHCT algorithm performs a search ordered in an ascending number of rejected data association pairings.

An example of pair rejection after the HOHCT application can be seen in Fig. 5. Initially an optimistic hypothesis taking all the pairings found during the matching step is tested. If this hypothesis fails to pass the compatibility test, a search for a satisfactory hypothesis is performed. In Fig. 5, the third feature matching is incompatible, thus both the prediction and the match found are rejected and ignored.

The proposed search algorithm combines iteration and recursion, as shown in Algorithm 1. The HOHCT will look for a compatible hypothesis, formed as a subset of an initial group of *'m'* pairings. If the initial hypothesis of taking all *'m'* pairing fails the test, a series of searches will be performed, each one trying to find and evaluate all hypotheses with *'m-i'* pairings. Each successive search will establish a compatible hypothesis, or increase *'i'*

before the subsequent search, so the number of pairs ignored in the hypotheses grows. Should a search looking for hypotheses with a given number of pairings find more than one compatible hypothesis, the hypothesis with the lowest Mahalanobis distance will be considered the best hypothesis and will be kept in the system.

So, each of the successive searches explores an *n*-ary tree: iteration steps add accepted pairs into the hypothesis (indicated as '1'); and recursive calls introduce rejected pairs (indicated as '0'). This allows for each search to be made into a pseudo-binary tree where only the interesting nodes (those having exactly '*m-i*' accepted pairs) are visited. An example of all the trees for a search with '*m=4*' is shown on Fig. 6.

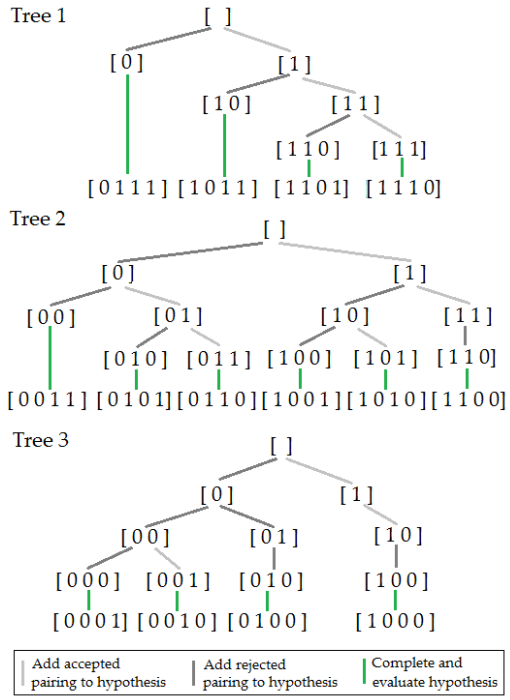

Figure 6. Example of search increasing number of pairs rejected

Given the sparse error conditions found in the DI-D initialization SLAM, this ordered search will normally have a linear cost with the number of landmarks matched, with exceptional cases achieving a cubic cost over some frames, still very far from the exponential cost that binary tree recursion could suppose over the whole number of matched landmarks. In this work, the rejected landmarks are eliminated as they are deemed incompatible.

Function (h_i, z_i, S_i, ∇H_i) := HOHCT-test (h_i, z_i,∇H_i, S_i)

Input:

z_i	matching observations found
h_i	features observation prediction
S_i	innovation covariance matrix
∇H_i	observation Jacobian

Output:

z_i	matching observations found
h_i	features observation prediction
S_i	innovation covariance matrix
∇H_i	observation Jacobian

```
begin
m: = Number of Matches in zi
hyp := [1]^m                          // Grab all matches
 if ~JointCompatible( hyp, hi, zi , ∇Hi, Si) then
   i := 1
   while i < m do                     // Hypothesis reducer loop
     (hyp,d2) := HOHCT-Rec(m,0,[],i, hi, zi,∇Hi, Si)
     if JointCompatible( hyp, hi, zi,∇Hi, Si) then
          i := m
     else
          i := i + 1
     end if
   end while
  remove incompatible pairings from hi and zi
  update jacobian ∇Hi and matrix Si
 end if
return (hi, zi, Si, ∇Hi)
```

Function (hyp_b, $d2_b$) := HOHCT-Rec (m, m_{hyp}, hyp_s, rm, h_i, z_i,∇H_i, S_i)

Input:

m	size of full hypothesis
m_{hyp}	size previously formed hypothesis
hyp_s	hypothesis built through recursion
rm	matches yet to remove

Output:

hyp_b	best hypothesis found from hyp_s
$d2_b$	best Mahalanobis distance

```
begin
if (rm = 0) or (m = mhyp) then
    hypb := [mhyp [1]^(m-mhyp)]
    d2b := Mahalanobis (hi, zi,∇Hi , Si)
else
    hypb := [hyps  [1]^(m-mhyp)]
    d2b := Mahalanobis(hi, zi, ∇Hi, Si)
    for r:= (mhyp+1) : (m-rm+1) do
          (h,d):= HOHCT-Rec (m, mhyp+1,
                [hyps 0],rm-1, hi, zi, ∇Hi, Si)
          If (d < d2b) then
                d2b := d
                hypb := h
          end if
          hyps := [hyps 1]
          mhyp := mhyp +1
    end for
end if
return (hypb, d2b)
```

Algorithm 1. HOHCT test and algorithm.

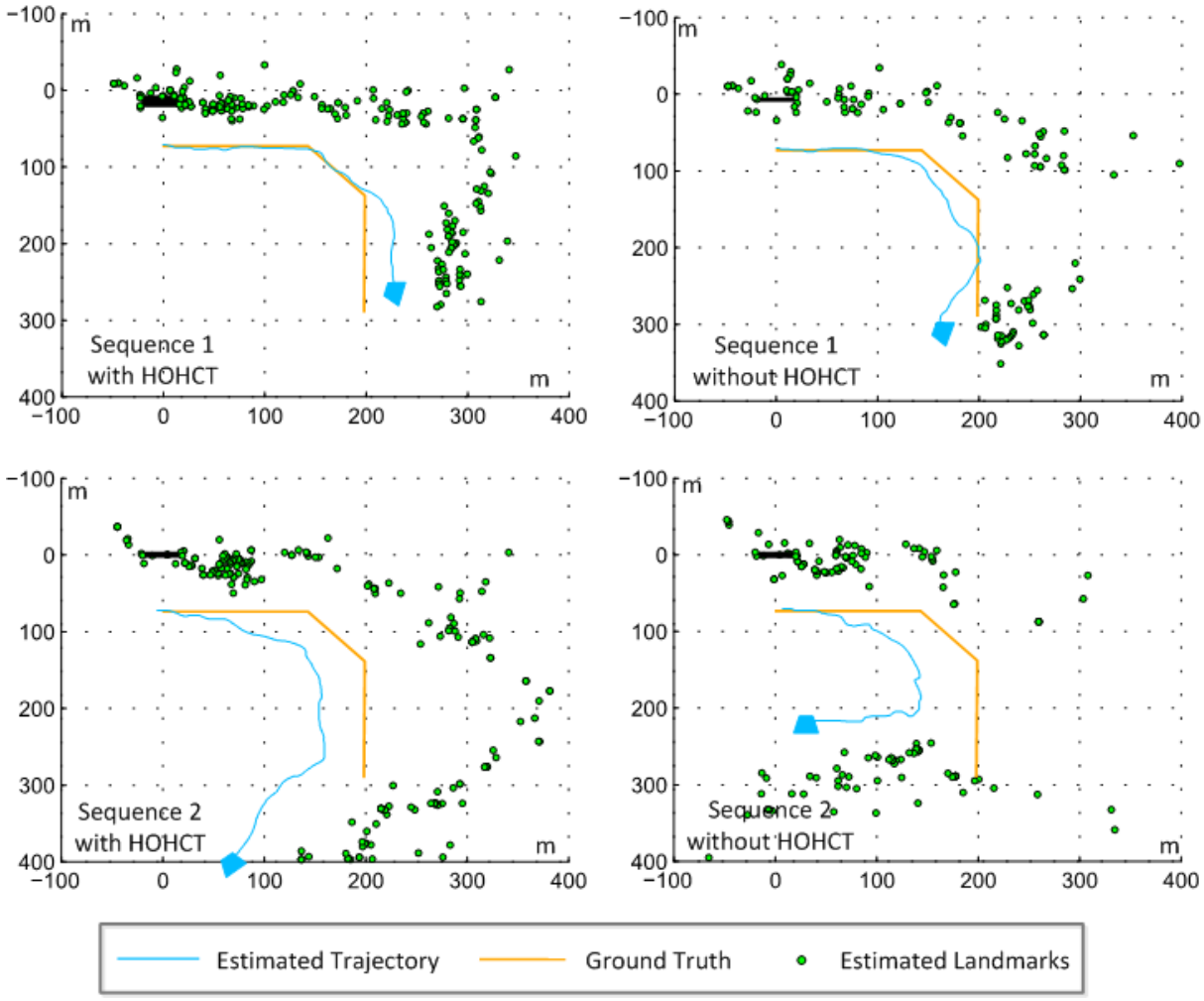

Figure 7. Map and trajectory estimation results obtained from two sequences of video: (1) 845 frames and (2) 615 frames. The left column displays results using HOHCT, while the right column displays results for the same sequences without using HOHCT batch validation.

In Fig. 5 the different trees that could be considered are represented. Tree 1 represents the first set of hypotheses that will be used to calculate the distance between the pairs. If any hypothesis is not compatible (if the statistical test fails) then Tree 2 is built and the compatibility is checked again, this process of tree creation is iteratively repeated until a compatible hypothesis is found.

4. Experimental Results and Discussion

In order to test the performance of the proposed method, several video sequences were acquired using a low cost camera, in two different environments. Later, a MATLAB© implementation of the algorithm was executed off-line using those video sequences as input signals.

4.1 Indoor and Outdoor Experiments

In experiments, the following parameter values were used: variances for linear and angular velocity respectively σ_V=4(m/s)², σ_Ω=4(m/s)², noise variances σ_u= σ_v= 1 pixel, minimum base-line b_{min}=15cm and minimum parallax angle α_{min}=5º. The default confidence level for the Chi-squared distribution was set to τ = 0.95.

A Logitech C920 HD camera was used in the experiments. This low cost camera has an USB interface and a wide angle lens. It is capable of acquiring HD colour video. However, in experiments, grey level video sequences, with a resolution of 424 × 240 pixels captured at 15 frames per second, were used. It is important to note that all the sequences of the video were captured at a relatively low frame rate of 15 frames per second (fps). While this frame rate would increase the difficulty of the SLAM process itself, and make it more prone to error, it would also give a bigger window of time in which to process each frame in an implementation aiming for real-time. So, although satisfactory results would be easier to achieve assuming 30 fps streams of image (in the literature, most of the experiments are reported to be captured at least at 25 frames per second), it has been considered a better option to evaluate SLAM results at 15 fps, to eventually allow for an easier implementation into systems with limited power, such as autonomous robots.

All the indoor video sequences were captured inside the Vision and Intelligent Systems laboratory at the authors' university. A rail guide was assembled in order to provide an approximate ground truth reference. Every video sequence in this scenario was captured by sliding the camera (manually), at different speeds, over the rail guide. The duration of the different sequences for both scenarios ran from 35 seconds to 1 minute (525 to 900 frames) for the same trajectory.

Figure 7 illustrates the estimated map and trajectory for two different video sequences for the first scenario. The left and right columns show the results, for each sequence, respectively with and without HOHCT validation. As we would expect, the estimations obtained with the HOHCT validation were consistently better. In this case the experimental results show that the HOHCT validation test significantly improves the algorithm robustness, by rejecting harmful matches, clearly noted in the improvement of orientation estimation, as seen in sequence 2. Another observed improvement was the enhanced preservation of the metric scale on estimations. The sequences taken with slower camera movements showed generally better results, although this can be easily attributed to the low frame rate deliberately employed.

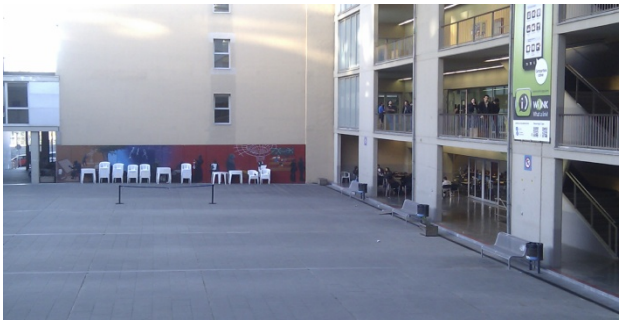

Figure 8. Outdoor environment setup used for tests.

The outdoor sequences were captured with the help of a robotic platform Pioneer 3-AT to provide accurate navigation. This robotic platform repeatedly traversed a known trajectory in a nearby courtyard with columns, benches and multiple reflective surfaces among other elements (Fig. 8). This trajectory constituted an 'L' shaped course running for 12m, with a 90° curve. While going along this trajectory, a camera installed on top of the platform captured the sequences, looking sideways. The courtyard allowed us to perform outdoor tests, with an open space and longer trajectories, while keeping disturbances from uncontrolled lightning to a minimum and other difficulties usually associated with this kind of test.

The sequences were taken with the platform moving at different speeds, ranging from 0.25 m/s to 1m/s. Thus, the duration of the sequences went approximately from 20 seconds to 90 seconds (600 to 1300 frames) for different tours in the same trajectory. Figure 9 shows the results of the off-line application of the DI-D SLAM technique, with and without the application of the HOHCT, with the robotic platform moving at 0.65 m/s. The most notable difference with the indoor handheld experiments is the capability to move at greater speeds while maintaining filter convergence in the SLAM process. This was due mainly to two facts: the robotic platform described a much cleaner trajectory, with a constant speed along the straight parts, and smoother turns; and the presence of objects at a wider range, which allowed it to keep a better estimation of the orientation. The impact of the HOHCT can be clearly seen in the different trajectory estimations: while both SLAM with and without HOHCT are able to estimate quite accurately the length of the trajectory, the estimation drifts greatly without data association validation, especially in terms of orientation.

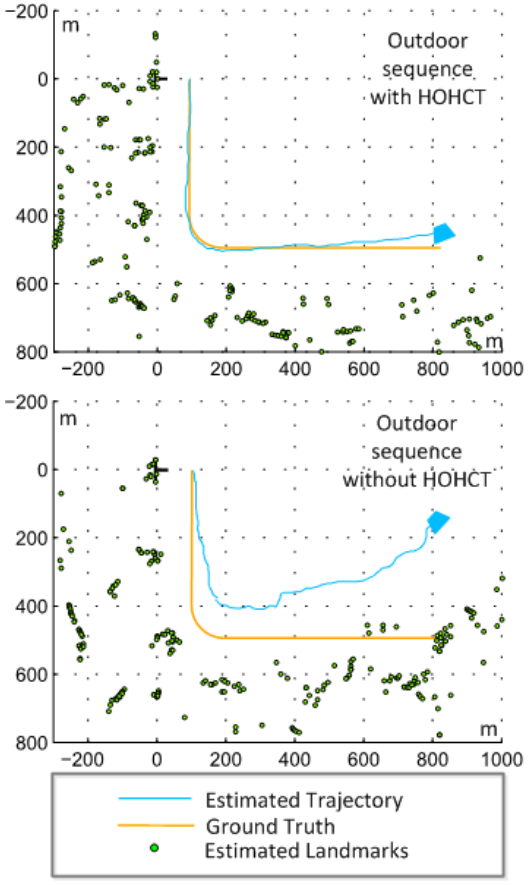

Figure 9. Outdoor experiment results navigating with a Pioneer 3-AT platform at 0.65m/s.

4.2 Discussion

The benefits of the application of the HOHCT validation come together with the addition of the computational cost h_m:

$$h_n = \sum_{i=1}^{n} n^r \qquad (22)$$

where n is the number of landmarks observed and r the quantity of these observations deemed incompatible. Thus the number of hypothesis to explore defines the order of the cost (each hypothesis' Mahalanobis distance is efficiently computed according to Eq. 21). As can be seen in Eq. 22, the theoretical computational cost for the HOHCT test could be quite high. However, the experimental data shows that the HOHCT algorithm tends to present linear cost when used along with the Delayed feature initialization technique.

Table 1 shows the results obtained in the indoor tests. For a sample of 7350 frames (accumulated over different sequences of video) in only 22 (0.3%) of the cases did the computational cost of the test become quadratic. For the indoor sample a case of r = 3 never occurred.

Pairing Rejected	Incidences	Relative Frequency
$r = 2$	22	0.3%
$r = 1$	234	3.18%
Compatible	7094	96.52%
Total	7350	100%

Table 1. Results of HOHCT failed tests and number of matching deemed incompatible for indoor test.

On the other hand, the outdoor test revealed a greater ratio of pairing rejection. Though being a relatively controlled outdoor space (inside a courtyard), a not much bigger sample (of 12750 frames) produced a greater ratio of rejection of data association pairings, and also produced cases with a cubic cost for the search. Note how the ratio of the rejection of single pairings almost doubles. This means that the cost will grow in less structured and/or less controlled environments.

The cost of the HOHCT can be optimized still when trying to reach real time performance. As the HOHCT already tries to search and test the minimal number of hypotheses, the most time consuming part of the algorithm is during the estimation of Mahalanobis distance. At this step, which is repeated for each hypothesis, a matrix inversion takes place. A strategy to compute the Mahalanobis distances in an incremental way could be developed, allowing for the exploitation of techniques to optimally compute the inversion of iteratively growing matrices, as demonstrated in [26].

Pairing Rejected	Incidences	Relative Frequency
$r = 3$	6	0.05%
$r = 2$	72	0.5%
$r = 1$	663	5.2%
Compatible	12009	94.18%
Total	12750	100%

Table 2. Results of HOHCT failed tests and number of matching deemed incompatible for outdoor sample.

5. Conclusions

A method to address the problem of batch validation within the context of the monocular SLAM technique [20] has been presented in this paper, within the context of the DI-D, although the proposed technique, the HOHCT, could in fact be used in any feature based SLAM approach. The considered monocular SLAM technique, the inverse depth delayed initialization, presents a series of particular features and characteristics due to its estimation methodology.

Landmarks are only introduced into the Kalman filter once the available depth estimation is accurate enough and fewer landmarks are introduced than in the un-delayed approaches. Despite these particularities, a data association gating technique is needed, mainly when the estimation process fails to converge to a good solution. Thus a batch validation technique based on statistical thresholding is introduced. This batch validation technique, the Highest Order Hypothesis Compatibility Test, has been shown to greatly improve the monocular SLAM results under certain circumstances. These circumstances include the emergence of 'false' landmarks, difficult illumination conditions and irregular trajectories with non-smooth changes to linear or angular velocities. Additionally, tests show that the algorithm responsible for finding the best hypothesis will rarely go beyond the quadratic cost with respect to the set of observed features, in fact being more prone to linear cost. This occurs when the batch validation test fails despite having a worst case scenario cost which is almost exponential.

Future works are expected to progress along two main lines of research. First, a method to deal with bigger maps is to be introduced, exploiting the characteristics of the inverse depth delayed initialization as the SLAM technique. A view was presented in [19], though new approaches are being studied in order to determine which would better fit the technique. The second line will work towards the production of a real-time implementation of the techniques proposed. This shall help evaluate the perspectives for the introduction of monocular SLAM as a reliable replacement of high end sensors, such as laser range finders.

6. Acknowledgment

This work has been funded by the Spanish Ministry of Science Project DPI2010-17112.

7. References

[1] Auat F, De la Cruz C, Carelli R, Bastos T (2011). Navegación Autónoma Asistida Basada en SLAM para una silla de Ruedas Robotizada en Entornos

Restringidos. Revista Iberoamericana de Automática e Informática Industrial. v. 8, pp.81-92. (in Spanish).
[2] Vázquez-Martín R, Núñez P, Bandera A, Sandoval F (2009). Curvature-based environment description for robot navigation using laser range sensors. Sensors. v.8, pp. 5894-5918.
[3] Jin H, Favaro P, Soatto S (2003). A Semi-Direct Approach to Structure from Motion. The Visual Computer. v. 19(6), pp. 377- 394.
[4] Fitzgibbon AW, Zisserman A (1998). Automatic Camera Recovery for Closed or Open Image Sequences. Proceedings of the European Conference on Computer Vision.
[5] Strelow S, Sanjiv D (2003). Online motion estimation from image and inertial measurements. Workshop on Integration of Vision and Inertial Sensors INERVIS.
[6] Kwok NM, Dissanayake G (2003). Bearing-only SLAM in indoor environments. Australasian Conference on Robotics and Automation.
[7] Kwok NM, Dissanayake G (2005). Bearing-only SLAM using a SPRT based Gaussian sum filter. IEEE International Conference on Robotics and Automation.
[8] Davison A, Gonzalez Y, Kita N (2004). Real-Time 3D SLAM with wide-angle vision. IFAC Symposium on Intelligent Autonomous Vehicles.
[9] Montiel JMM, Civera J, Davison A (2006). Unified inverse depth parameterization for monocular SLAM. Robotics: Science and Systems Conference.
[10] Durrant-Whyte H, Bailey T (2006). Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine. v.13, pp. 99-106.
[11] Durrant-Whyte H, Bailey T (2006). Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine. v.13, pp. 15-23.
[12] Neira J, Tardos JD (2001). Data association in stochastic mapping using the joint compatibility test. IEEE Transaction on Robotics and Automation. v.17(6), pp. 890–897.
[13] Bailey T (2002). Mobile robot localisation and mapping in extensive outdoor environments. Ph.D. dissertation, Univ. Sydney, Australian Ctr. Field Robotics.
[14] Lategahn H, Geiger A, Kitt B (2011). Visual SLAM for autonomous ground vehicles. IEEE International Conference on Robotics and Automation. pp.1732-1737.
[15] Caccia M, Bruzzone G, Ferreira F, Veruggio G (2009). Online video mosaicing through SLAM for ROVs. OCEANS 2009 – EUROPE. pp.1-6.
[16] Weiss S, Scaramuzza D, Siegward R (2011). Monocular SLAM–Based Navigation for Autonomous Micro Helicopters in GPS-Denied Environments. Journal of Field Robotics, 28(6), pp. 854-874.
[17] Klein G, Murray D (2007). Parallel Tracking and Mapping for Small AR Workspaces. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225-234.
[18] Strasdat H, Montiel JMM, Davison AJ (2010). Real-time monocular SLAM: Why filter?. IEEE International Conference on Robotics and Automation. pp. 2657-2664.
[19] Munguia R, Grau A (2009). Closing Loops With a Virtual Sensor Based on Monocular SLAM. IEEE Trans. on Instrumentation and Measurement, 58(8), pp. 2377-2384.
[20] Munguia R, Grau A (2008). Delayed Inverse Depth Monocular SLAM**.** 17th IFAC World Congress.
[21] Chatterjee C, Roychowdhury V (2000). Algorithms for coplanar camera calibration. Machine Vision and Applications. v. 12, pp. 84-97.
[22] Fishler M, Bolles RC (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM. v.24, pp. 381–395.
[23] Ganapathy S (1984). Decomposition of Transformation Matrices for Robot Vision. Proceedings of International Conference on Robotics and Automation. pp 130–139.
[24] Chiuso A, Favaro P, Jin H, Soatto S (2000). MFm: 3-D Motion from 2-D Motion Causally Integrated over Time. Proceedings of the European Conference on Computer Vision.
[25] Davison A, Murray D (1998). Mobile Robot Localisation using Active Vision. Proceedings of the European Conference on Computer Vision.
[26] David A (1998). Matrix Algebra from a Statistician's Perspective, Springer-Verlag. New York, NY, USA.