An Active Strategy for Plane Detection and Estimation with a Monocular Camera

Paolo Robuffo Giordano, Riccardo Spica, and François Chaumette

***Abstract*— Plane detection and estimation from visual data is a classical problem in robotic vision. In this work we propose a novel *active* strategy in which a monocular camera tries to determine whether a set of observed point features belongs to a common plane, and, if so, what are the associated plane parameters. The active component of the strategy imposes an optimized camera motion (as a function of the observed scene) able to maximize the convergence in estimating the scene structure. Based on this strategy, two methods are then proposed to solve the plane estimation task: a classical solution exploiting the homography constraint (and, thus, almost completely based on image correspondances across distant frames), and an alternative method fully taking advantage of the scene structure estimated incrementally during the camera motion. The two methods are extensively compared in several case studies by discussing the various pros/cons.**

I. INTRODUCTION

Plane detection and estimation from raw visual data is a classical problem in sensor-based robot control, especially in the context of mobile robotics. Indeed, planes are widespread in artificial (man-made) and natural environments, and therefore constitute the typical 3D structure one tries to segment in order to, e.g., plan safe paths among planar obstacles (e.g., vertical walls), or navigate by keeping a desired attitude or distance from special planes (e.g., ground plane for flying robots). The ability to classify planes in the perceived environment is therefore an important feature for several sensor-based applications.

When dealing with images taken by a (possibly moving) camera, a number of approaches has been developed for solving the problem of detecting and identifying planes from visual data. Several methods for instance exploit known correspondances across frames to identify point features (or directly pixels) as whether belonging to a common plane together with the associated plane parameters. These methods usually rely on special geometric constraints linking two views of a planar scene such as the well-known and widely used *homography constraint* [1]. Other methods, instead, attempt to directly measure (using special sensors such as the RGB-D) or recover (exploiting structure from motion algorithms) a 'depth map' of the observed images, for then dealing with the issue of clustering and extracting planes from clouds of 3D points.

P. Robuffo Giordano is with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr.

R. Spica is with the University of Rennes 1 at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France riccardo.spica@irisa.fr

F. Chaumette is with Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France francois.chaumette@irisa.fr

Examples from the first category can be found in [2]–[4] where the homography constraint in its various forms (discrete and/or continuous version) is employed. For instance, in [3] the homography constraint is exploited to classify features belonging to the (dominant) ground plane for a mobile robot equipped with a camera. The method however requires an initialization step to determine the location of the ground plane at the beginning of the motion. The authors in [2] address a similar problem but by including a Kalman filtering step that exploits the temporal correlation between consecutive images to improve the estimation of the homography matrix. Also, special constraints of the employed robot (a ground wheeled mobile robot) are used to simplify the problem. In [4] the *continuous* version of the homography constraint is instead used to segment the optical flow detected by a flying robot into clusters belonging to different planar patches. Finally, within the second category of methods dealing with 3D point clouds one can mention [5]–[7] and references therein. In these cases, the problem is rather on how to fit planes to sets of 3D points and on how to cluster them according to some reasonable 'planarity measure'.

With respect to this state-of-the-art, the problem addressed in this paper is the following: given a monocular camera observing a (possibly time-varying) set of N feature points in the scene, find an *active* strategy able to determine, in an optimized way, whether the N points belong to a common plane and what are the associated plane parameters (distance and normal vector). No special assumptions are made on the N points, nor special constraints are assumed for the camera motion. The sought strategy is termed *active* in the sense that, following the framework proposed in [8], it aims at controlling online the camera motion (as a function of the observed scene) in order to optimize the convergence rate of the plane estimation task. In this sense, our method differs from most of the previous literature which assumes a camera moving in an 'non-informed' way, i.e., without attempting to affect online its motion for facilitating the plane estimation.

The rest of the paper is then organized as follows: Section II briefly reviews the theoretical framework of [8], while Sect. III introduces and details the proposed strategy for plane detection and estimation. Here, two alternative methods are proposed: a first solution based on the classical homography decomposition, and a second solution fully exploiting the structure of the scene estimated exploiting the framework in [8]. The two methods are then extensively compared in Sect. V with several simulation case studies which point out the respective pros/cons. Interestingly, the (widely-used) homography decomposition results highly sensitive to non-

idealities in the scene, with the other proposed method being instead much more robust to real-world conditions. Sect. VI then concludes the paper.

II. REVIEW OF ACTIVE STRUCTURE FROM MOTION

In this section, we briefly summarize the active estimation framework proposed in [8]. Let $(\boldsymbol{s},\, \boldsymbol{\chi}) \in \mathbb{R}^{m+p}$ be the state of a dynamical system in the form

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{s}} & = & \boldsymbol{f}_m(\boldsymbol{s},\, \boldsymbol{u}) + \boldsymbol{\Omega}^T(t)\boldsymbol{\chi} \\ \dot{\boldsymbol{\chi}} & = & \boldsymbol{f}_u(\boldsymbol{s},\, \boldsymbol{\chi},\, \boldsymbol{u}) \end{array} \right. \tag{1}$$

where $\boldsymbol{s}$ and $\boldsymbol{\chi}$ represent, respectively, a *measurable* and *unmeasurable* component of the state, and $\boldsymbol{u} \in \mathbb{R}^v$ is the system input vector. In formulation (1) vector $\boldsymbol{\chi}$ is required to appear *linearly* in the dynamics of $\boldsymbol{s}$ (first equation), and matrix $\boldsymbol{\Omega}(t) \in \mathbb{R}^{p\times m}$ to be a *known* and sufficiently smooth time-varying quantity.

Structure from Motion (SfM) problems can be recast to formulation (1) by taking $\boldsymbol{s}$ as the set of visual features measured on the image plane, $\boldsymbol{u} = (\boldsymbol{v},\, \boldsymbol{\omega})$ as the camera linear/angular velocity in camera frame, and $\boldsymbol{\chi}$ as a suitable (and locally invertible) function of the unknown structure of the scene to be estimated[1]. Furthermore, for SfM problems one has $\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v}(t))$ with, in particular, $\boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{0}) \equiv \boldsymbol{0}$: as well-known, the camera linear velocity $\boldsymbol{v}$ plays a key role for the convergence of SfM algorithms.

For a system in form (1), a possible estimation scheme can be devised as follows: letting $(\hat{\boldsymbol{s}}\,, \hat{\boldsymbol{\chi}}) \in \mathbb{R}^{m+p}$ be the estimated state, $\boldsymbol{\xi} = \boldsymbol{s} - \hat{\boldsymbol{s}}$, $\boldsymbol{z} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$, consider the following observer

$$\left\{ \begin{array}{rcl} \dot{\hat{\boldsymbol{s}}} & = & \boldsymbol{f}_m(\boldsymbol{s},\, \boldsymbol{u}) + \boldsymbol{\Omega}^T(t)\hat{\boldsymbol{\chi}} + \boldsymbol{H}\boldsymbol{\xi} \\ \dot{\hat{\boldsymbol{\chi}}} & = & \boldsymbol{f}_u(\boldsymbol{s},\, \hat{\boldsymbol{\chi}},\, \boldsymbol{u}) + \alpha\boldsymbol{\Omega}(t)\boldsymbol{\xi} \end{array} \right. \tag{2}$$

with $\boldsymbol{H} > 0$ and $\alpha > 0$. Note also that observer (2) *does not* require knowledge of $\dot{\boldsymbol{s}}$ (i.e., measurement of velocities on the image plane), but it only needs measurement of $\boldsymbol{s}$ (the 'visual features') and of $(\boldsymbol{v},\, \boldsymbol{\omega})$ (the camera linear/angular velocity in the camera frame).

Following the derivations in [8], one can show that, by a proper (state-dependent) choice of gain $\boldsymbol{H}$, the dynamics of the estimation error $\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$ (the error in estimating the structure of the scene $\boldsymbol{\chi}$) is equivalent to that of the following second-order linear and diagonal system

$$\ddot{\boldsymbol{\eta}} + 2\sqrt{\alpha}\boldsymbol{S}\dot{\boldsymbol{\eta}} + \alpha\boldsymbol{S}^2\boldsymbol{\eta} = 0, \tag{3}$$

where $\boldsymbol{\eta} \in \mathbb{R}^p$, $\boldsymbol{S} = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{p\times p}$, and $0 \leq \sigma_1^2 \leq \ldots \leq \sigma_p^2$ are the p eigenvalues of the square matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$. The convergence rate of system (3) is then dictated by the quantity $\alpha\sigma_1^2$, with σ_1^2 being the smallest eigenvalue of $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$. Since in the SfM case $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v})$, one can show that [8]

$$\dot{(\sigma_1^2)} = \boldsymbol{J}_v\dot{\boldsymbol{v}} + \boldsymbol{J}_s\dot{\boldsymbol{s}}, \tag{4}$$

[1]For instance, in the point feature case [9], $\boldsymbol{\chi}$ can be taken as the *inverse* of the feature depth Z, and, for image moments of planar scenes, $\boldsymbol{\chi}$ can be taken as the normal vector of the observed plane scaled by its distance from the camera optical center [10].

where the Jacobian matrices $\boldsymbol{J}_v \in \mathbb{R}^{1\times 3}$ and $\boldsymbol{J}_s \in \mathbb{R}^{1\times m}$ have a *closed form* expression function of $(\boldsymbol{s},\, \boldsymbol{v})$ (known quantities). It is then possible to invert (4) w.r.t. vector $\dot{\boldsymbol{v}}$ so as to act on $\sigma_1^2(t)$, e.g., for maximizing its value. We note that this step represents the *active* component of the strategy since, in the general case, inversion of (4) will yield a camera velocity $\boldsymbol{v}(t)$ function of the system measured state $\boldsymbol{s}(t)$.

We conclude by providing the explicit expressions of the above machinery for the point feature case (which is the case considered in the next developments). Assume a *calibrated* pin-hole camera, and let $\boldsymbol{s} = (x,\, y) = (X/Z,\, Y/Z)$ be the normalized perspective projection of a 3D point $(X,\, Y,\, Z)$ onto the image plane. Formulation (1) can be applied by taking $\chi = 1/Z$ with, thus, $m = 2$ and $p = 1$, and

$$\left\{ \begin{array}{l} \boldsymbol{f}_m\,(\boldsymbol{s},\, \boldsymbol{u}) = \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix}\boldsymbol{\omega} \\ \boldsymbol{\Omega}\,(\boldsymbol{s},\, \boldsymbol{v}) = \begin{bmatrix} xv_z - v_x & yv_z - v_y \end{bmatrix} \\ f_u\,(\boldsymbol{s}, \chi, \boldsymbol{u}) = v_z\chi^2 + (yw_x - xw_y)\,\chi \end{array} \right. . \tag{5}$$

Furthermore, one has

$$\sigma_1^2 = \boldsymbol{\Omega}\boldsymbol{\Omega}^T = (xv_z - v_x)^2 + (yv_z - v_y)^2 \tag{6}$$

as the single eigenvalue of $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$, and in (4)

$$\left\{ \begin{array}{rl} \boldsymbol{J}_v & = 2\begin{bmatrix} v_x - xv_z & v_y - yv_z & (xv_z - v_x)\,x + (yv_z - v_y)\,y \end{bmatrix} \\ \boldsymbol{J}_s & = 2\begin{bmatrix} (xv_z - v_x)v_z & (yv_z - v_y)v_z \end{bmatrix} \end{array} \right. . \tag{7}$$

III. DETECTION AND ESTIMATION OF A PLANE FROM A SET OF POINT FEATURES

Let $\boldsymbol{P}_i = (X_i,\, Y_i,\, Z_i)$ be N 3D points expressed in the camera frame, and let $\boldsymbol{p}_i = (x_i,\, y_i,\, 1) = (X_i/Z_i,\, Y_i/Z_i,\, 1)$ be the corresponding normalized feature positions on the (assumed calibrated) camera image plane. The problem addressed in this paper is how to *optimally* determine whether the N points $\boldsymbol{P}_i$ belong to a common plane, and what are the associated plane parameters (normal and distance). We now detail two possible strategies to achieve this goal.

A. Plane reconstruction from estimated 3D points

Assume that an estimation $\hat{Z}_i$ of the unknown depth Z_i of each point is available. Then, from each measured point feature $\boldsymbol{p}_i$ one can recover an estimation $\hat{\boldsymbol{P}}_i = \hat{Z}_i\boldsymbol{p}_i$ of the corresponding 3D point $\boldsymbol{P}_i$ in the current camera frame. Let $\mathcal{P}:\ \boldsymbol{n}^T\boldsymbol{E} + d = 0$ be the equation of the sought plane, with $\boldsymbol{n} \in \mathbb{S}^2$ and $d \in \mathbb{R}$ representing the unit normal vector and distance in camera frame. For the estimated points $\hat{\boldsymbol{P}}_i$ to belong to $\mathcal{P}$, it must hold

$$\boldsymbol{n}^T\hat{\boldsymbol{P}}_i + d = 0, \quad i = 1\ldots N. \tag{8}$$

Equation (8) can be rearranged in matrix form as

$$\begin{bmatrix} \hat{\boldsymbol{P}}_1^T & 1 \\ \vdots & \vdots \\ \hat{\boldsymbol{P}}_N^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{0} \tag{9}$$

with $\boldsymbol{A} \in \mathbb{R}^{N\times 4}$. Assuming $N \geq 4$, the linear system (9) has a unique solution (up to a scalar factor) iff $\mathrm{rank}(\boldsymbol{A}) = 3$. Let $\boldsymbol{U}_A\boldsymbol{S}_A\boldsymbol{V}_A^T = \boldsymbol{A}$ be the singular value decomposition

of matrix $\boldsymbol{A}$, with $\sigma^2_{1,A} \leq \cdots \leq \sigma^2_{4,A}$ being the associated singular values. The inverse of the condition number $\sigma_A = \sigma^2_{1,A}/\sigma^2_{4,A}$ can be taken as a normalized measure of the planarity of the N points $\hat{\boldsymbol{P}}_i$ ($\mathrm{rank}(\boldsymbol{A}) = 3 \iff \sigma_A = 0$). Furthermore, as well-known, a (least-square) solution of the homogeneous system (9) is given by $\bar{\boldsymbol{v}} = (\bar{v}_1, \ldots \bar{v}_4)$, the column of $\boldsymbol{V}_A$ associated to $\sigma^2_{1,A}$. From $\bar{\boldsymbol{v}}$ one can then recover

$$\begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \pm \frac{\bar{\boldsymbol{v}}}{\sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2}}, \tag{10}$$

i.e., by imposing $\|\boldsymbol{n}\| = 1$. The final sign ambiguity can be resolved by fixing the sign of d according to the adopted convention.

Summarizing, given a collection of N (estimated) 3D points $\hat{\boldsymbol{P}}_i$, one can obtain a measure of their planarity by computing σ_A ($\sigma_A = 0$ if they belong to the same plane, $\sigma_A > 0$ otherwise), and then from (10) obtain a *unique* solution for the plane parameters $(\boldsymbol{n}, d)$ which best fits the N points. Clearly, one still faces the issue of obtaining the N estimations $\hat{\boldsymbol{P}}_i$ from the measured feature points $\boldsymbol{p}_i$. Section IV explains how to optimally solve this problem exploiting the active estimation framework of the previous Sect. II.

B. Plane reconstruction from the homography decomposition

As a (well-known and widely-used) alternative method, one can also exploit the homography constraint for recovering the plane normal $\boldsymbol{n}$ from a moving camera observing N feature points, see [1]. In short, let ${}^0\mathcal{F}_C$ be the camera frame of reference at the beginning of the motion, $\mathcal{F}_C$ the camera frame at the current time (i.e., after some displacement has taken place), and $(\boldsymbol{R}, \boldsymbol{T})$, $\boldsymbol{R} \in SO(3)$, $\boldsymbol{T} \in \mathbb{R}^3$, be the rotation matrix from ${}^0\mathcal{F}_C$ to $\mathcal{F}_C$ and the position of ${}^0\mathcal{F}_C$ w.r.t. $\mathcal{F}_C$ and expressed in $\mathcal{F}_C$, respectively. Let also ${}^0\boldsymbol{p}_i$ and $\boldsymbol{p}_i$ represent the measured locations of the i-th feature point in frames ${}^0\mathcal{F}_C$ and $\mathcal{F}_C$.

Assuming again a planar scene $\mathcal{P}: \ \boldsymbol{n}^T\boldsymbol{E} + d = 0$, it is well-known that the following relationship holds for all the N image pairs $({}^0\boldsymbol{p}_i, \boldsymbol{p}_i)$

$$[\boldsymbol{p}_i]_\times \boldsymbol{\mathcal{H}} {}^0\boldsymbol{p}_i = 0, \quad i = 1 \ldots N, \tag{11}$$

where $[\boldsymbol{x}]_\times \boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$ and $\boldsymbol{\mathcal{H}} = \boldsymbol{R} + \frac{\boldsymbol{T}}{d}\boldsymbol{n}^T \in \mathbb{R}^{3\times 3}$ is the so-called *homography matrix*, which encodes the structure of the scene (plane parameters) and the displacement among the two frames. Equation (11) can be rearranged as $\boldsymbol{b}_i^T \boldsymbol{\mathcal{H}}^s = 0$ where $\boldsymbol{b}_i = {}^0\boldsymbol{p}_i \otimes [\boldsymbol{p}_i]_\times \in \mathbb{R}^{9\times 3}$ (with $\otimes$ indicating the Kronecker product), and $\boldsymbol{\mathcal{H}}^s = (\boldsymbol{\mathcal{H}}_{11}, \boldsymbol{\mathcal{H}}_{21}, \ldots, \boldsymbol{\mathcal{H}}_{33}) \in \mathbb{R}^9$. By now letting $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N) \in \mathbb{R}^{3N\times 9}$ be the collection of all the N $\boldsymbol{b}_i$, one can compactly rewrite equation (11) for all measured pairs as

$$\boldsymbol{B}\boldsymbol{\mathcal{H}}^s = \boldsymbol{0}. \tag{12}$$

Similarly to before, equation (12) has a unique solution, up to a scalar factor, iff $\mathrm{rank}(\boldsymbol{B}) = 8$. Letting then $\sigma^2_{1,B} \leq \ldots \leq \sigma^2_{9,B}$ be the singular values of B, one can take again the inverse of the condition number $\sigma_B = \sigma^2_{1,B}/\sigma^2_{9,B}$ as an alternative measure of planarity besides the previously introduced σ_A.

A (least-square) solution $\boldsymbol{\mathcal{H}}^s$ of (12) can again be found by exploiting the singular value decomposition of $\boldsymbol{B}$ (by taking the column of $\boldsymbol{V}_B$ associated to $\sigma^2_{1,B}$). Using standard algorithms [1], it is finally possible to decompose the associated recovered homography $\boldsymbol{\mathcal{H}}$ into *two* physically possible solutions $(\boldsymbol{R}_1, \boldsymbol{T}_1/d, \boldsymbol{n}_1)$ and $(\boldsymbol{R}_2, \boldsymbol{T}_2/d, \boldsymbol{n}_2)$. However, the ambiguity among these two solutions can only be resolved by exploiting prior knowledge of the scene (e.g., approximated known direction of $\boldsymbol{n}$ in one of the two frames, or comparison against the homography estimated from a third frame).

We note that the above machinery only involves observed image pairs $({}^0\boldsymbol{p}_i, \boldsymbol{p}_i)$, and thus allows to compute σ_B and to recover the plane normal $\boldsymbol{n}$ without requiring knowledge of the (unknown) depths Z_i as in the previous case. Knowledge of the scene 3D structure is nevertheless still needed for obtaining the plane distance d: from the recovered $\boldsymbol{n}$ and the estimated $\hat{\boldsymbol{P}}_i$, a simple possibility from (8) is indeed

$$d = -\frac{\sum_{i=1}^N \boldsymbol{n}^T \hat{\boldsymbol{P}}_i}{N}. \tag{13}$$

C. Final considerations

Let us denote with *method A* (based on the estimated $\hat{\boldsymbol{P}}_i$) and *method B* (based on the classical homography decomposition) the two techniques discussed in the previous Sect. III-A and Sect. III-B. It is interesting to draw the following comparison:

1) Assumptions: while method A only relies on a frame-by-frame tracking of the point features, method B requires the correspondences of several points across distant frames (the initial and the current ones). Furthermore, method A can straightforwardly cope with the loss/gain of feature points (e.g., because of visibility constraints as shown in Sect. V-C), while method B needs to either always keep (a subset of) the initial features within visibility, or to periodically reinitialize the initial frame at the current one (thus, temporarily suffering from a small baseline);

2) Complexity: the planarity measure σ_A is obtained from the svd decomposition of the $N \times 4$ matrix $\boldsymbol{A}$ in (9), while evaluation of σ_B requires the decomposition of the $3N \times 9$ matrix $\boldsymbol{B}$ in (12). For large N one can then prefer σ_A (method A) in terms of reduced algebraic computational load;

3) Convergence rate: method A relies, in all its steps, on the estimated 3D points $\hat{\boldsymbol{P}}_i$. Therefore, during the transient phase of the depth estimation error (i.e., when $Z_i - \hat{Z}_i$ is still large), no reliable results can be expected by method A, while method B can *in principle* be successfully employed for recovering $\boldsymbol{n}$ and computing σ_B as soon as the camera has undergone a sufficient displacement w.r.t. its initial pose;

4) Accuracy: Method A provides a *unique* solution for $\boldsymbol{n}$ (eq. (10)), while method B results in *two* physically possible $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ which must then be disambiguated. Finally, and as it will be shown extensively in Sect. V, the homography decomposition of method B results *highly sensitive* to non-idealities of the observed scene (e.g., when the observed

points $\boldsymbol{P}_i$ are approximately, but not exactly, planar), with method A being instead much more robust to these issues.

In conclusion method A results superior to method B in all aspects apart from the potentially slower convergence rate, which is anyway traded for a much higher robustness w.r.t. non-idealities as those found in real-world conditions.

IV. OPTIMAL DEPTH ESTIMATION FOR A SET OF POINT FEATURES

We now address the issue of optimally recovering the unknown depths Z_i of the N observed point features $\boldsymbol{p}_i$ by optimizing the camera motion. We stress, again, that this 'optimized depth estimation' only relies on the measured $\boldsymbol{p}_i$ and on the (assumed known) camera velocity $(\boldsymbol{v}, \boldsymbol{\omega})$. We also note that the N points $\boldsymbol{P}_i$ are *not* required to be planar for applying the following strategy, but they can be arranged in any spatial configuration (as long as they can be tracked).

In case of N point features, one can directly apply observer (2) by defining $\boldsymbol{s} = (\boldsymbol{p}_1, \dots \boldsymbol{p}_N) \in \mathbb{R}^m$ and $\boldsymbol{\chi} = (1/Z_1, \dots, 1/Z_N) \in \mathbb{R}^p$, with $m = 2N$ and $p = N$. Since the dynamics of each $\boldsymbol{s}_i = \boldsymbol{p}_i$ is only affected by its associated unknown $\chi_i = 1/Z_i$, this is equivalent to implementing in parallel N instances of (2), one for each tracked feature (i.e., N instances with $m = 2$ and $p = 1$). Indeed, we can also note that, for N points, matrix $\boldsymbol{\Omega}$ takes the expression

$$\boldsymbol{\Omega} = \begin{bmatrix} x_1 v_z - v_x & 0 & 0 & \dots & 0 \\ y_1 v_z - v_y & 0 & 0 & \dots & 0 \\ 0 & x_2 v_z - v_x & 0 & \dots & 0 \\ 0 & y_2 v_z - v_y & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & x_N v_z - v_x \\ 0 & 0 & 0 & \dots & y_N v_z - v_y \end{bmatrix}^T, \quad (14)$$

with, therefore,

$$\boldsymbol{\Omega}\boldsymbol{\Omega}^T = \begin{bmatrix} \sigma_{1,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{1,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{1,N}^2 \end{bmatrix} = \mathrm{diag}(\sigma_{1,i}^2),$$

where

$$\sigma_{1,i}^2 = (x_i v_z - v_x)^2 + (y_i v_z - v_y)^2 \quad (15)$$

is the eigenvalue determining the convergence speed of the estimation error $z_i(t) = \chi_i(t) - \hat{\chi}_i(t) = 1/Z_i(t) - 1/\hat{Z}_i(t)$ for the i-th feature point, see (5–6).

In order to optimize the error convergence of the whole error vector $\boldsymbol{z}(t)$, one can simply aim at maximizing the mean $\sigma^2 = \sum_{i=1}^N \sigma_{1,i}^2/N$. Let $\boldsymbol{J}_{v,i}$ and $\boldsymbol{J}_{s,i}$ be the Jacobian matrixes associated to each $\sigma_{1,i}^2$ as given in (7). One then has

$$\dot{(\sigma^2)} = \frac{1}{N}\left(\sum_{i=1}^N \boldsymbol{J}_{v,i}\dot{\boldsymbol{v}} + \sum_{i=1}^N \boldsymbol{J}_{s,i}\dot{\boldsymbol{s}}_i\right).$$

Maximization of σ^2 can then be obtained by choosing $\dot{\boldsymbol{v}}$ as

$$\dot{\boldsymbol{v}} = \dot{\boldsymbol{v}}_\sigma = \frac{k_\sigma}{N}\sum_{i=1}^N \boldsymbol{J}_{v,i}^T - \left(\sum_{i=1}^N \boldsymbol{J}_{v,i}\right)^\dagger \sum_{i=1}^N \boldsymbol{J}_{s,i}\hat{\dot{\boldsymbol{s}}}_i, \quad k_\sigma > 0, \quad (16)$$

where † denotes the pseudo-inverse operator. Note that in (16) the (not directly measured) $\dot{\boldsymbol{s}}_i$ is replaced by an estimation $\hat{\dot{\boldsymbol{s}}}_i$ obtained by evaluating (1) on $\hat{\chi}_i$. This, indeed, avoids the need of obtaining the (possibly noisy) image velocity $\dot{\boldsymbol{s}}_i$ by, e.g., numerical differentiation.

V. RESULTS

In this section we report five simulated case studies meant to illustrate the proposed machinery for plane detection and estimation. In all simulations, we considered a free-flying camera delivering images at 30 Hz (the update rate for obtaining vector $\boldsymbol{s}$) and controlled at 100 Hz (the update rate for imposing the camera velocity commands $(\boldsymbol{v}, \boldsymbol{\omega})$). The vector of visual features $\boldsymbol{s}$ was corrupted, component-wise, with a uniformly distributed random noise of 2 pixels.

In all simulations we considered presence of a plane $\mathcal{P}$ with $\boldsymbol{n} = (0, 0, 1)$ and $d = -1$ [m] in ${}^0\mathcal{F}_C$, and tested methods A and B in five different combinations: (i) cases I–II involve a *perfectly planar scene* made of $N = 10$ points $\boldsymbol{P}_i$ lying exactly on $\mathcal{P}$ and randomly generated from a uniform distribution of radius 0.2 [m] (therefore, all points start with $Z_i(t_0) = 1$ [m]). The camera is also assumed to have an unlimited field of view (fov). Case I implements the optimization action on σ^2, while case II does not implement any optimization action; (ii) cases III and IV involve a (more realistic) *approximately planar scene* made again of $N = 10$ points generated as in cases I–II, but by then corrupting their position with an additional uniformly distributed noise of amplitude 0.05 [m] along the direction of $\boldsymbol{n}$ (thus, simulating presence of an uncertainty of 5% in the planarity assumption of points $\boldsymbol{P}_i$ w.r.t. the initial camera pose). The camera is again assumed to have an unlimited fov. Case III implements the optimization of σ^2 while case IV does not implement it; (iii) the last case V involves again an *approximately planar scene* as in the previous cases III–IV, but considers a *limited* camera fov. The possibility of losing/gaining features over time when exiting/entering the camera fov is then taken into account. This case implements the optimization action on σ^2.

As for the optimization of σ^2, we note that each $\sigma_{1,i}^2$ in (15) depends on the norm of the camera linear velocity $\boldsymbol{v}$ (the larger the camera speed $\sim \|\boldsymbol{v}\|$, the faster the estimation error convergence for a given set of gains). In order to obtain a fair comparison among all cases, we thus considered the maximization of σ^2 under the constraint $\|\boldsymbol{v}\| = const$. This was obtained as follows: letting $\kappa = \frac{1}{2}\|\boldsymbol{v}\|^2$, $\kappa_{des} = \frac{1}{2}\|\boldsymbol{v}(t_0)\|^2$, we replaced (16) with

$$\dot{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} k_1 (\kappa_{des} - \kappa) + k_2 \left(\boldsymbol{I} - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2}\right)\dot{\boldsymbol{v}}_\sigma \quad (17)$$

with $k_1 > 0$ and $k_2 \geq 0$. Cases I–III–V were then implemented with $k_1 > 0$ and $k_2 > 0$ (maximization of σ^2), while cases II–IV with $k_1 > 0$ and $k_2 = 0$ (no action on σ^2). Finally, the camera angular velocity $\boldsymbol{\omega}$ was exploited in all cases for keeping the centroid of the observed point features $\boldsymbol{p}_i$ at the center of the image plane (as $\sigma^2 = \sigma^2(\boldsymbol{v}, \boldsymbol{s})$, one can freely chose $\boldsymbol{\omega}$ to fulfil additional goals).

A. Camera with unlimited field of view and perfect planar scene (cases I-II)

For illustrating the results of cases I–II, let us denote with $(\hat{\boldsymbol{n}}_A, \hat{d}_A)$ and $(\hat{\boldsymbol{n}}_B, \hat{d}_B)$ the estimation of the plane parameters $(\boldsymbol{n}, d)$ obtained with method A and method B, respectively, and then define $e_{d_A} = d - \hat{d}_A$, $e_{d_B} = d - \hat{d}_B$, $e_{n_A} = \arccos(\boldsymbol{n}^T \hat{\boldsymbol{n}}_A)$, $e_{n_B} = \arccos(\boldsymbol{n}^T \hat{\boldsymbol{n}}_B)$ as the corresponding estimation errors[2]. The following values were used in the simulations: $k_1 = 10$ and $k_2 = 50$ (for case II) in (17), $\alpha = 200$ in (2), and $\boldsymbol{v}(t_0) = (-0.05, 0.05, 0.1)$ with, thus, $\|\boldsymbol{v}(t_0)\|^2 = 0.015$. The initial values for the estimations $\hat{Z}_i(t_0)$ were taken as the real $Z_i(t_0)$ plus a uniformly distributed random noise of amplitude 0.5 [m].

The results of the four cases are reported in Figs. 1(a–l) from which we can then draw the following considerations: first of all, note how the convergence speed of the depth estimation error $\boldsymbol{z}(t)$ is much slower in case I w.r.t. case II (Fig. 1(c) vs. Fig. 1(d)). Thanks to the active maximization of σ^2 in the latter case, convergence of the depth estimation errors is approximately reached in about 4 [s]. We recall that in both cases the camera was traveling with the same linear velocity norm $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}(t_0)\|$: the faster convergence of $\boldsymbol{z}(t)$ was then only due to the 'active optimization' of the direction of $\boldsymbol{v}$ as dictated by (17). One can find a similar pattern in the behavior of the N eigenvalues $\sigma^2_{1,i}$ in Figs. 1(e–f): note how in case I (Fig. 1(e)) all eigenvalues stay below 0.001, with some of them decreasing over time, while in case II (Fig. 1(f)) the eigenvalues are actively regulated towards the common value $\|\boldsymbol{v}(t_0)\|^2 = 0.015$ (as explained in [8], when $\boldsymbol{p}_i \simeq (0, 0)$ one has $\max \sigma^2_{1,i} = \|\boldsymbol{v}\|^2$).

Coming to the planarity test and the estimation of the plane parameters $(\boldsymbol{n}, d)$, from Figs. 1(g–h) one can see that $\sigma_B(t) \simeq 0$, $\forall t$ (red solid line), while $\sigma_A(t)$ converges to zero only when vector $\boldsymbol{z}(t)$ has reached convergence (compare Figs. 1(g–h) with Figs. 1(c–d)). Both criteria then successfully recognize the N points as belonging to a common plane but, clearly, σ_B outperforms σ_A. This is not surprising since, as explained in Sect. III-C, the quantity σ_B is obtained in terms of sole image measurements $({}^0\boldsymbol{p}_i, \boldsymbol{p}_i)$, while σ_A requires a good enough knowledge of the estimated 3D points $\hat{\boldsymbol{P}}_i$ (and, thus, provides a reliable answer only when $\boldsymbol{z}(t)$ is close to convergence).

A similar behavior can be found in Figs. 1(i–j) for the errors e_{n_A} (solid blue line) and e_{n_B} (dashed red line). The estimated normal $\hat{\boldsymbol{n}}_B$ of method B (homography decomposition) coincides almost immediately with $\boldsymbol{n}$. The initial 'noisy' behavior of $e_{n_B}(t)$ is due to the lack of a large enough baseline w.r.t. the simulated image noise added to the measurement $\boldsymbol{s}$ which negatively affects the homography decomposition. Finally, having obtained a good normal estimation $\hat{\boldsymbol{n}}_B$ allows to symmetrically obtain a good estimation $\hat{d}_B$ from (13). This is shown in Figs. 1(k–l) where the errors (e_{d_A}, e_{d_B}) are reported. We can again note how $e_{d_B}(t)$ (dashed red line) quickly converges to 0 when compared to $e_{d_A}(t)$ (solid blue line).

[2]Here, for simplicity we always assumed ability to disambiguate among the two physically possible solutions of method B.

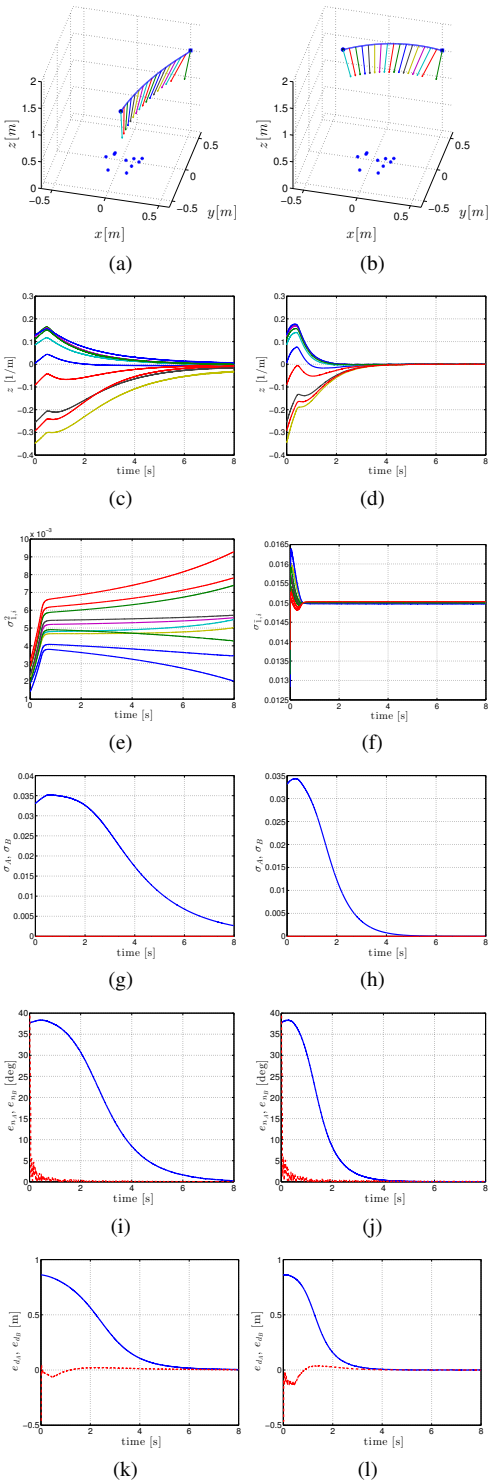

Fig. 1: Simulations with unlimited camera fov and perfect planar scene (case I on the left, case II on the right). (a–b) camera 3D trajectory; (c–d) behavior of the depth estimation error $\boldsymbol{z}(t)$; (e–f) behavior of the N eigenvalues $\sigma^2_{1,i}$; (g–h) behavior of the planarity measures σ_A (solid blue line) and σ_B (solid red line); (i–j) behavior of e_{n_A} (solid blue line) and e_{n_B} (dashed red line); (k–l) behavior of e_{d_A} (solid blue line) and e_{d_B} (dashed red line)

Summarizing, these results clearly show two facts: on the one hand, they illustrate the benefits of the optimization

action on σ^2 for improving the convergence of $\boldsymbol{z}(t)$. On the other hand, they also indicate method B (based on the classical homography decomposition) as apparently superior for what concerns both the planarity test and the estimation of the plane parameters $(\boldsymbol{n}, d)$. This latter conclusion is, however, completely different when considering the (more realistic) situation of an approximately planar scene of the next cases III–IV.

B. Camera with unlimited field of view and approximately planar scene (cases III-IV)

The results are reported in Figs. 2(a–l). Here, Figs. 2(g–h) show again the planarity criteria σ_A and σ_B: being the scene not exactly planar, both measures correctly reach a *constant non-zero value* which represents the confidence level in considering the points $\boldsymbol{P}_i$ as belonging to the same plane. Note, however, that now σ_B (solid red line) has a transient behavior qualitatively equivalent to σ_A (solid blue line). Thus, both quantities provide a similar level of information before reaching their respective 'steady-state'. Furthermore, when considering the errors e_{n_A} and e_{n_B} in Figs. 2(i–j) we can note another interesting result: while $e_{n_B}(t)$ (dashed red line) has a highly erratic behavior (indicating a quite unreliable estimation of $\hat{\boldsymbol{n}}_B$), $e_{n_A}(t)$ (solid blue line) converges towards 0 (here, the error is computed w.r.t. the normal $\boldsymbol{n}$ best fitting the real N points $\boldsymbol{P}_i$). Analogous considerations also hold for the errors e_{d_A} (solid blue line) and e_{d_B} (dashed red line) in Figs. 2(k–l): $e_{d_A}(t)$ correctly converges to 0 while $e_{d_B}(t)$ does not converge at all.

These results then allow to conclude that, in the more realistic condition of an approximately planar scene (with an error of $\approx 5\%$ w.r.t. the camera initial distance to the plane), method B is not able to provide a reliable estimation of the plane parameters $(\boldsymbol{n}, d)$ as opposite to method A which, instead, shows an unsatisfactory performance. This is most likely because the homography decomposition of method B only relies on image correspondences and is, thus, highly sensitive to non-idealities such as image noise or non perfect planarity of the observed points, while method A exploits the estimation of the point depths $\hat{Z}_i$ for then drawing conclusions from an (estimated) cloud of 3D points in the *current* camera frame.

C. Camera with limited field of view (case V)

In this last simulation we considered a camera with a limited fov for introducing the possibility of losing/gaining point features during the camera motion because of the visibility constraint. This was meant to test the robustness of the proposed estimation strategy when relaxing the (often unrealistic) assumption of being able to track the same set of point features over time. The simulation involved a total of $N = 15$ points arranged in an approximately planar scene, and the active optimization of σ^2. Only method A was employed for recovering the plane parameters $(\hat{\boldsymbol{n}}_A, \hat{d}_A)$ since, as explained, method B requires keeping a sufficient number feature points always within the camera fov.

Whenever during motion a new point feature $\boldsymbol{p}_i$ entered visibility, its estimated depth $\hat{Z}_i$ was initialized so as to force

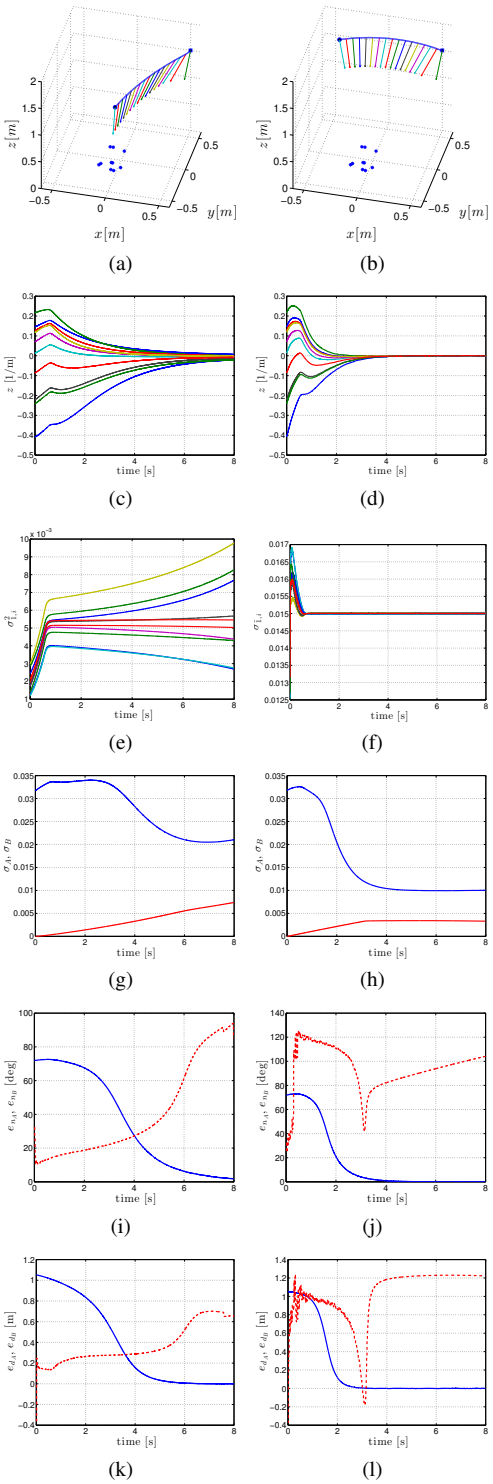

Fig. 2: Results of the simulations with unlimited camera fov and approximately planar scene (case III on the left, case IV on the right). Case III is shown on the left side and case IV on the right side. The pattern of the plots is the same as in Figs. 1(a–l).

$\hat{\boldsymbol{P}}_i = \hat{Z}_i \boldsymbol{p}_i$ to belong to the current estimation of the plane given by $(\hat{\boldsymbol{n}}_A, \hat{d}_A)$, i.e., by choosing

$$\hat{Z}_i = -\hat{d}_A / (\hat{\boldsymbol{n}}_A^T \boldsymbol{p}_i). \tag{18}$$

The identity of each feature exiting visibility was not retained

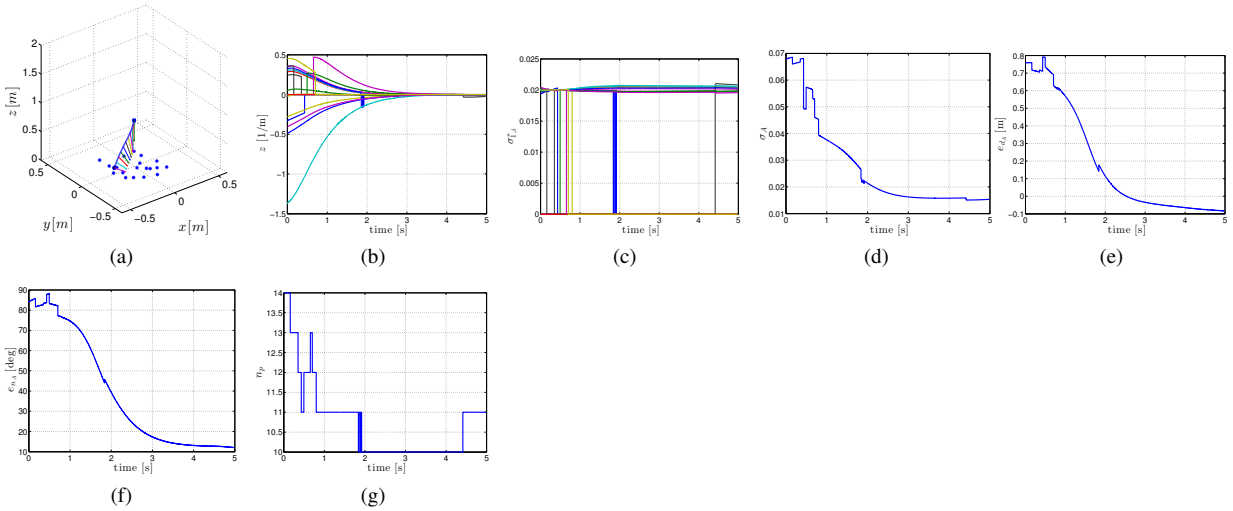

Fig. 3: Results of the simulations with limited camera fov (case V). (a) camera 3D trajectory; (b) behavior of the depth estimation error $\boldsymbol{z}(t)$; (c) behavior of the N eigenvalues $\sigma^2_{1,i}$; (d) behavior of the planarity measures σ_A; (e) behavior of e_{n_A}; (f) behavior of e_{d_A} ; (g) behavior of n_p, the total number of features in visibility. The vertical jumps in the various plots indicate loss/gain of point feature which exit/enter the image plane

and, when re-entering in the image plane, it was treated as a newly acquired point. The simulation was run with the same gains of the previous cases and by taking $\boldsymbol{v}(t_0) = (-0.1,\, 0.1,\, 0)$ with, thus, $\|\boldsymbol{v}(t_0)\|^2 = 0.02$.

Figures 3(a–g) show the results of the simulation: Fig. 3(a) depicts the camera trajectory in space with the arrow indicating its optical axis. Figure 3(b) shows the convergence of the estimation error $\boldsymbol{z}(t)$ and Fig. 3(c) the behavior of the $N = 15$ eigenvalues $\sigma^2_{1,i}$ over time. Note, again, how all $\sigma^2_{1,i}$ approximately reach and keep the value $\|\boldsymbol{v}(t_0)\|^2$ thanks to the active optimization of σ^2. The vertical jumps in the plots represent features which have left/entered the image plane, with their estimated depths being either discarded ($\hat{Z}_i = 0$) or reset as in (18).

Figure 3(d) depicts the behavior of the planarity measure $\sigma_A(t)$, and Figs. 3(e–f) the estimation errors $e_{d_A}(t)$ and $e_{n_A}(t)$. We can again note how method A is able to successfully determine the (approximate) planarity of the observed points and the associated plane parameters despite the additional visibility constraint which forces features to randomly enter/exit the camera fov (note, again, the several jumps in the plots indicating loss/gain of some features). Finally, Fig. 3(g) shows the total number of tracked features n_p over time (which keeps varying as expected).

VI. CONCLUSIONS

In this paper we have presented and critically compared two methods for determining whether a set of point features belongs to a common plane and the associated plane parameters. The first method exploits an estimation of the point depths for retrieving the 'best plane' fitting a set of 3D points, while the second method is based on the classical homography decomposition and, thus, strongly depends on correspodances across distant image frames (initial and current ones) or on a reinitialization of the initial frame. Both methods also rely (to different extents) on a newly developed active SfM strategy which allows to optimize online the camera trajectory in order to maximize the SfM convergence rate. An extensive set of simulation results in realistic conditions was then presented for assessing the pros/cons of both methods: the results showed the poorer performance of the classical homography-based approach w.r.t. the other approach.

We are currently aiming for an experimental validation of this approach by also investigating the use of different kinds of visual features (e.g., dense/discrete image moments), as well as more sophisticated strategies able to cope with outliers and/or presence of multiple planes in the scene.

REFERENCES

[1] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An invitation to 3D vision*. Springer, 2003.

[2] J. Arróspide, L. Salgado, M. Nieto, and R. Mohedano, "Homography-based ground plane detection using a single on-board camera," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 149–160, 2010.

[3] C.-H. Lin, S.-Y. Jiang, Y.-J. Pu, and K.-T. Song, "Robust Ground Plane Detection for Obstacle Avoidance of Mobile Robots Using a Monocular Camera," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 3706–3711.

[4] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 2153–2159.

[5] N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3D range images," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3378–3383.

[6] Y. Suttasupa, A. Sudsang, and N. Niparnan, "Plane Detection for Kinect Image Sequences," in *2011 Int. Conf. on Robotics and Biomimetics*, 2011, pp. 970–975.

[7] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a new Accumulator Design," *3D Res*, vol. 2, pp. 32:1–32:13, 2011.

[8] R. Spica and P. Robuffo Giordano, "A Framework for Active Estimation: Application to Structure from Motion," in *52nd IEEE Conf. on Decision and Control*, 2013.

[9] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[10] P. Robuffo Giordano, A. De Luca, and G. Oriolo, "3D structure identification from image moments," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, may 2008, pp. 93–100.