Plane Estimation by Active Vision from Point Features and Image Moments

Riccardo Spica, Paolo Robuffo Giordano, François Chaumette

▶ **To cite this version:**

Riccardo Spica, Paolo Robuffo Giordano, François Chaumette. Plane Estimation by Active Vision from Point Features and Image Moments. IEEE Int. Conf. on Robotics and Automation, ICRA'15, May 2015, Seattle, United States. hal-01121631

HAL Id: hal-01121631
https://inria.hal.science/hal-01121631

Submitted on 2 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Plane Estimation by Active Vision from Point Features and Image Moments

Riccardo Spica, Paolo Robuffo Giordano, and François Chaumette

***Abstract*— In this paper we experimentally validate and compare three different methods for estimating the 3D parameters of a planar scene from a (possibly time-varying) set of feature points acquired by a moving monocular camera. The first method, based on the classical decomposition of the homography matrix, is meant to serve as a *baseline condition* classically used in many previous works. The other two methods exploit an active Structure from Motion (SfM) scheme for either extracting the plane from the reconstructed 3D position of all the tracked points, or for directly estimating the plane parameters by considering a set of *discrete image moments* as visual input. The possible loss/gain of point features during the camera motion is considered in all three methods by, in particular, introducing a suitable weighting strategy for the image moment case. Finally, the results of an experimental validation are presented with a comparative discussion of the pros/cons of the three methods.**

I. INTRODUCTION

Detection and estimation of 3D planes from the raw visual data acquired by a moving camera is a typical problem faced by, e.g., ground mobile robots or UAVs autonomously navigating in unknown indoor environments. Indeed, in many situations, and especially in artificial environments, the surrounding scene can be reasonably approximated as a collection of planes. This simple modelization is often sufficient for solving tasks such as navigating in a corridor or positioning inside a room [1].

Several methods have been proposed for identifying the 3D plane parameters from an image sequence. A widely used approach exploits the so-called *homography constraint*, that is, a geometric relationship linking two views of the same planar scene. Some examples in this context can be found in [2], [3]. Filtering techniques (such as EKF) can also be exploited to improve the estimation of the homography matrix as proposed in [4]. Alternative methods use special sensors (e.g., RGB-D cameras) or Structure from Motion (SfM) algorithms to first reconstruct a 3D point cloud of the surrounding environment for then extracting/classifying the planes present in the scene [1], [5]–[7].

The goal of this paper is to propose and experimentally validate/compare three different methods for estimating the 3D parameters of a planar scene having as input a (possibly time-varying) set of point features acquired by a moving camera. The first method is based on the classical and well-known decomposition of the homography matrix and is indeed included as a *baseline condition* to compare the other two methods against. The second and third methods are instead based on the *active* SfM algorithm proposed in [8] and tailored to the cases at hand. In particular, the second method exploits the active SfM scheme to first reconstruct the 3D positions of all the tracked points for then searching for the best fitting plane (in a least-square sense) in the resulting point cloud. This technique has already been introduced in [9] but only tested via numerical simulations. In this work we instead provide a full experimental validation of this approach in real conditions. Finally the last method (a novel contribution of this paper) tailors the active SfM scheme [8] for exploiting a set of *image moments* of the observed point features as visual input so as to directly yield the estimated plane parameters as output. As shown in the reported results, the advantages of this latter method lie in its higher robustness with respect to non-perfectly planar scenes thanks to its stronger 'filtering' action, as well as in its reduced complexity w.r.t. the second method (3 estimated states for any number of tracked points).

The possibility of losing/gaining point features during the camera motion (e.g., because of the limited camera fov) is also considered in all three methods. While this is easily accomplished for the first two methods, a non-trivial extension is instead needed in the third (moment-based) one, i.e., the introduction of a suitable *weighting function* in the definition of the image moments for smoothly taking into account new/lost features. This weighting strategy (and the associated weighted moment dynamics) is, to the best of our knowledge, a novel contribution and could also be exploited for dealing with dense and/or photometric image moments in the context of visual servoing [10].

Finally, the proposed machinery is termed *active* since it optimizes *online* the camera motion (as only a function of the available measurements) in order to maximize the convergence rate of the plane estimation error and thus obtain a higher estimation accuracy in a shorter time compared to a non-active case. This then additionally differentiates our contribution w.r.t. most of the previous literature which in general assumes a camera moving in a 'non-informed' way, i.e., without attempting to facilitate the plane estimation task during navigation. Finally, we remark that the active SfM approach adopted in this work has in fact several similarities with the notion of "sensor-based" or "ego-centric" Visual SLAM, see [11] for a recent overview. In both cases, a robot/camera builds a 3D model of the environment in its

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

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

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

own body/sensor frame via a filtering technique: a Kalman filter in [11] and similar works, and the deterministic (but with a fully characterized and *actively* optimized transient response) filter derived from [8] in our case.

The paper is organised as follows: in Sect. II we introduce the problem of plane estimation from measured point features and we briefly review the classical homography-based solution used for ground-truth comparison; in Sect. III we summarize the active SfM framework presented in [8] and describe how to apply it to the case of 3D reconstruction from a collection of feature points and from discrete image moments; in Sect. V we then present an experimental validation of the three proposed methods by discussing the various pros/cons; finally in Sect. VI we draw some conclusions and discuss possible future directions.

II. PROBLEM DESCRIPTION

Let $\mathcal{P} : \boldsymbol{n}^T\boldsymbol{E} + d = 0$ be the equation of a planar scene (expressed in the camera frame), with $\boldsymbol{n} \in \mathbb{S}^2$ being the unit normal vector and $d \in \mathbb{R}$ the distance to the plane. Let also $\boldsymbol{P}_k = (X_k, Y_k, Z_k)$ be a collection of N 3D points belonging to $\mathcal{P}$, and $\boldsymbol{p}_k = (x_k, y_k, 1) = (X_k/Z_k, Y_k/Z_k, 1)$ be the corresponding normalized feature positions measured on the image plane (the camera is assumed calibrated). The problem addressed in this work is how to estimate the 3D plane parameters $(\boldsymbol{n}, d)$ from the tracked N point features $\boldsymbol{p}_k$ gathered by a moving monocular camera (with N possibly time-varying). We stress that we are not interested in estimating *one* particular plane, but rather in identifying the parameters of the plane *currently* (i.e. within some time interval) dominating the scene observed by the camera.

Method A. Reconstruction from the homography constraint

A classical (and widely-used) possibility to recover the plane parameters $(\boldsymbol{n}, d)$ is to exploit the homography constraint linking two views of the same planar scene [12]: in brief, define ${}^0\mathcal{F}_C$ and $\mathcal{F}_C$ as the camera frames at the beginning of the motion and at the current time, and let ${}^0\boldsymbol{p}_k$ and $\boldsymbol{p}_k$ represent the measured locations of the k-th feature point in frames ${}^0\mathcal{F}_C$ and $\mathcal{F}_C$. By matching the N feature pairs $({}^0\boldsymbol{p}_k, \boldsymbol{p}_k)$, it is possible to algebraically compute the homography matrix relating the two views, and to then further decompose it via standard techniques for extracting the plane normal $\boldsymbol{n}$ in $\mathcal{F}_C$. In order to recover the plane distance d one needs to exploit some additional 'metric' information such as known translation between the two frames or the 3D coordinates of (at least) one of the tracked points $\boldsymbol{P}_k$. Assuming an estimation $\hat{\boldsymbol{P}}_k = \hat{Z}_k\boldsymbol{p}_k$ is available for all tracked points in $\mathcal{F}_C$, one simply has

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

This first possibility for recovering $(\boldsymbol{n}, d)$ from the homography decomposition will be denoted as *method A* throughout the rest of the paper.

We now discuss two additional methods based on an active SfM framework.

III. PLANE ESTIMATION FROM ACTIVE STRUCTURE FROM MOTION

We start by quickly reviewing the *active* SfM framework proposed in [8] and exploited in this work: let $\boldsymbol{s} \in \mathbb{R}^m$ be the set of visual features *measured* on the image plane, $\boldsymbol{\chi} \in \mathbb{R}^p$ a suitable (and locally invertible) function of the unknown structure of the scene *to be estimated* by the SfM algorithm, and $\boldsymbol{u} = (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^6$ the camera linear/angular velocity expressed in the camera frame. With these choices, one can show that the SfM dynamics takes the general form

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

where matrix $\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v}) \in \mathbb{R}^{p\times m}$ is a *known* quantity such that $\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{0}) \equiv \boldsymbol{0}$. Let now $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{\chi}}) \in \mathbb{R}^{m+p}$ be the estimated state, and define $\boldsymbol{\xi} = \boldsymbol{s} - \hat{\boldsymbol{s}}$ as the 'visual feedback' error (measured $\boldsymbol{s}$ vs. estimated $\hat{\boldsymbol{s}}$) and $\boldsymbol{z} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$ as the 3D structure estimation error. An estimation scheme for system (2) meant to build a converging estimation $\hat{\boldsymbol{\chi}}(t)$ from the measured $\boldsymbol{s}(t)$ (i.e., such that the estimation error $\boldsymbol{z}(t) \to 0$) can be devised as

$$\left\{\begin{array}{rcl} \dot{\hat{\boldsymbol{s}}} &=& \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{u}) + \boldsymbol{\Omega}^T(\boldsymbol{s}, \boldsymbol{v})\hat{\boldsymbol{\chi}} + \boldsymbol{H}\boldsymbol{\xi} \\ \dot{\hat{\boldsymbol{\chi}}} &=& \boldsymbol{f}_u(\boldsymbol{s}, \hat{\boldsymbol{\chi}}, \boldsymbol{u}) + \alpha\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v})\boldsymbol{\xi} \end{array}\right. \tag{3}$$

where $\boldsymbol{H} > 0$ and $\alpha > 0$ are suitable gains. We note that the scheme (3) *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 [8], it is possible to *characterize* the transient response of the SfM estimation error $\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$, as well as to *affect* it by acting *online* on the camera motion. One can indeed show that the convergence rate of $\boldsymbol{z}(t)$ results dictated by the norm of the square matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$, in particular by its smallest eigenvalue σ_1^2. For a given choice of gain α (a free parameter), the larger σ_1^2 the faster the error convergence, with in particular $\sigma_1^2 = 0$ if $\boldsymbol{v} = 0$ (as well-known, only a translating camera can estimate the scene structure). Being $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v})$, one has

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

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), see [8]. This relationship can then be inverted w.r.t. vector $\dot{\boldsymbol{v}}$ for affecting *online* $\sigma_1^2(t)$ during motion, e.g., in order to maximize its value for increasing the convergence rate of $\boldsymbol{z}(t)$. We note that this step represents the *active* component of the estimation 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 now apply this general active SfM framework to the problem of structure estimation for a planar scene.

Method B. Reconstruction from 3D points

If the (estimated) 3D points $\hat{\boldsymbol{P}}_k$ are available, a second possibility to recover the pair $(\boldsymbol{n}, d)$ is to directly search

for the best fitting plane in $\mathcal{F}_C$. By rearranging the plane constraint $\boldsymbol{n}^T\hat{\boldsymbol{P}}_k + d = 0$, $k = 1 \dots N$, one has

$$\begin{bmatrix} \hat{\boldsymbol{P}}_1^T & 1 \\ \vdots & \vdots \\ \hat{\boldsymbol{P}}_N^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{0} \tag{5}$$

with $\boldsymbol{A} \in \mathbb{R}^{N\times 4}$. Assuming $N \geq 4$ and $\mathrm{rank}(\boldsymbol{A}) = 3$, the linear system (5) has a unique solution (up to a scalar factor) for the pair $(\boldsymbol{n},\, d)$ which can be found by standard least-square techniques (svd decomposition of matrix $\boldsymbol{A}$).

As for the issue of optimally recovering the unknown depths Z_k for the N tracked point features $\boldsymbol{p}_k$, this can be addressed by exploiting the SfM scheme (3). As discussed in [9], let $\boldsymbol{s} = (x_1,\, y_1, \dots,\, x_N,\, y_N) \in \mathbb{R}^{2N}$ be the vector of measured visual features, and $\boldsymbol{\chi} = (1/Z_1, \dots,\, 1/Z_N) \in \mathbb{R}^N$ be the 3D structure to be estimated (the depths of all tracked points). This choice results in the matrix

$$\boldsymbol{\Omega}\boldsymbol{\Omega}^T = \mathrm{diag}(\sigma_{1,1}^2, \sigma_{1,2}^2, \dots \sigma_{1,N}^2), \tag{6}$$

with

$$\sigma_{1,k}^2 = (x_k v_z - v_x)^2 + (y_k v_z - v_y)^2 \tag{7}$$

being the eigenvalue determining the convergence speed of the k-th estimation error $z_k(t) = \chi_k(t) - \hat{\chi}_k(t) = 1/Z_k(t) - 1/\hat{Z}_k(t)$ for the k-th feature point. Exploiting (4), optimization of the convergence of the whole vector $\boldsymbol{z}(t)$ can then be obtained by, e.g., maximizing the minimum eigenvalue

$$\sigma_m^2 = \min_{k=1\dots N} \sigma_{1,k}^2 \tag{8}$$

w.r.t. the camera linear velocity $\boldsymbol{v}$.

We finally note that this method does not require the exact matching of point features across distant frames (initial and current ones) as it is instead the case for method A, but it only needs a frame-by-frame tracking. As a consequence, the method can straightforwardly cope with loss/gain of feature points because of, e.g., limited field of view: new estimated points $\hat{\boldsymbol{P}}_k$ can be added to system (5) by initializing the corresponding estimated depth $\hat{Z}_k$ so as to belong to the current estimation of the plane $\mathcal{P}$. The only assumption (common to all the methods) is that all the tracked points seen by the moving camera belong to a common plane[1].

In the following, this second possibility for recovering $(\boldsymbol{n},\, d)$ will be denoted as *method B*.

Method C. Plane estimation from image moments

Having summarized methods A and B, we now propose a third novel possibility based on the machinery of *point-based* image moments originally introduced in [13]. This method, hereafter denoted as *method C*, can be seen as a further improvement of method B in that it exploits the active estimation scheme (3) for directly estimating the pair $(\boldsymbol{n},\, d)$ (3 independent quantities) instead of the N depths Z_k of the N considered point features $\boldsymbol{p}_k$ for then algebraically solving system (5). Thus, the complexity of the SfM scheme results reduced w.r.t. method B as the number of estimated states is independent of the number of tracked points. Furthermore, since $(\boldsymbol{n},\, d)$ are directly estimated via a filtering process, one can expect method C to be more robust than method B w.r.t. non perfectly planar scenes as no algebraic step is involved (contrarily to method B that still requires the least-square solution of the linear system (5)). Indeed, these considerations are also supported by the experimental results of Sect. V.

Consider then the $(i,\, j)$-th moment m_{ij} evaluated on the collection of N observed feature points $\boldsymbol{p}_k = (x_k,\, y_k\, 1)$

$$m_{ij} = \sum_{k=1}^{N} x_k^i y_k^j.$$

From [13], the dynamics of m_{ij} takes the expression

$$\dot{m}_{ij} = f_{\omega_{ij}}(m_{kl},\, \boldsymbol{\omega}) + \boldsymbol{f}_{\chi_{ij}}(m_{kl},\, \boldsymbol{v})\boldsymbol{\chi} \tag{9}$$

where m_{kl} stands for the generic $(k,\, l)$-th moment of order up to $i + j + 1$, and $\boldsymbol{\chi} = -\boldsymbol{n}/d \in \mathbb{R}^3$. Analogous considerations hold for the centered moments

$$\mu_{ij} = \sum_{k=1}^{N} (x_k - x_g)^i (y_k - y_g)^j$$

with $x_g = m_{10}/m_{00}$ and $y_g = m_{01}/m_{00}$ being the barycenter coordinates, and $m_{00} = N = const$ in this case. Furthermore, it is (see, e.g., [8])

$$\dot{\boldsymbol{\chi}} = \boldsymbol{\chi}\boldsymbol{\chi}^T\boldsymbol{v} - [\boldsymbol{\omega}]_\times\boldsymbol{\chi} = \boldsymbol{f}_u(\boldsymbol{\chi},\, \boldsymbol{u}).$$

The estimation scheme (3) can then be directly applied for recovering $\boldsymbol{\chi} = -\boldsymbol{n}/d$ by including in $\boldsymbol{s}$ a suitable collection of $m \geq 3$ image moments $\boldsymbol{s} = (m_{i_1 j_1}, \dots, m_{i_m j_m})$, and thus letting $\boldsymbol{f}_m = [f_{\omega_{i_1 j_1}} \dots f_{\omega_{i_m j_m}}]^T \in \mathbb{R}^m$ and

$$\boldsymbol{\Omega} = [\boldsymbol{f}_{\chi^{i_1 j_1}}^T \dots \boldsymbol{f}_{\chi^{i_m j_m}}^T] \in \mathbb{R}^{3\times m}. \tag{10}$$

This estimation strategy, however, lacks the possibility of taking into account the loss/gain of feature points over time (as it is instead the case with method B). When a feature point leaves visibility, a practical workaround could be to just redefine a moment m_{ij} as the sum over the remaining $N - 1$ points and feed the estimation scheme (9) with this new measurement (and analogously for new points entering visibility). However, this would clearly introduce a discontinuity in the measured m_{ij} — a discontinuity not modeled by the dynamics (9) which predicts the moment evolution as only a function of the camera own motion $(\boldsymbol{v},\, \boldsymbol{\omega})$. Therefore, we now propose a redefinition of *weighted image moments* meant to explicitly cope with this issue.

Assume presence of a countable number of feature points on the plane $\boldsymbol{p}_k = (x_k,\, y_k,\, 1)$, $k = 1 \dots \infty$, and define the $(i,\, j)$-th weighted moment as

$$m_{ij} = \sum_{k=1}^{\infty} w(x_k,\, y_k,\, t - t_k) x_k^i y_k^j, \tag{11}$$

[1]The results of Sect. V will nevertheless test the robustness of the methods against this hypothesis.

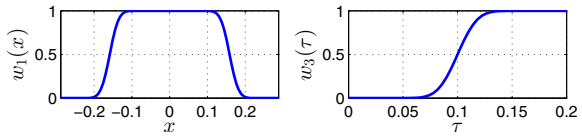

Fig. 1: Left: shape of weight $w_1(x)$ with limits $x_{min} = -x_{max} = 0.2884$ (normalized size of the image plane). Right: shape of weight $w_3(\tau)$.

where the *weighting function* $w(x,\, y,\, \tau) : \mathbb{R}^3 \rightarrow [0,\, 1]$ is a sufficiently smooth map, and t_k represents the time at which the point feature $\boldsymbol{p}_k$ is considered for the first time.

The weight w can be exploited to assign a 'quality' measure to each feature point so as to enforce a smooth change in m_{ij} whenever a tracked feature leaves visibility or a new feature is taken into consideration (regardless of its position on the image plane). In particular, we design the weight $w(x,\, y,\, \tau)$ as the product of three scalar functions

$$w(x,\, y,\, \tau) = w_1(x) w_2(y) w_3(\tau).$$

Weights $w_1(x)$ and $w_2(y)$ are designed so as to vanish at the image borders and are meant to smoothly take into account features entering/exiting the image plane. Weight $w_3(\tau)$ is finally intended to smoothly take into account the introduction of a newly detected feature point p_k when already within visibility (for instance, when starting to track a new point close to the image center). Figure 1 shows a possible shape for $w_1(x)$ (also representative of $w_2(y)$) and $w_3(\tau)$.

Exploiting the definition (11), it is easy (although tedious) to obtain an expression conceptually equivalent to (9) for the dynamics of the weighted moments $\dot{m}_{ij}$. Some details in this sense are reported in the Appendix. This then allows to directly apply the SfM scheme (3) to the case of weighted moments. We finally note that, in practice, the summation (11) is clearly evaluated only on the (*finite* but time-varying) set of currently tracked point features since $w(x_k,\, y_k,\, t - t_k) = 0$ for any $\boldsymbol{p}_k$ not visible or not considered at any time $t \leq t_k$.

As for which moments to consider for the estimation of $\boldsymbol{\chi}$, after some experimental tests we opted for

$$\boldsymbol{s} = (x_g,\, y_g,\, \mu_{20},\, \mu_{02},\, \mu_{11}) \in \mathbb{R}^m, \qquad m = 5. \tag{12}$$

This choice is partially motivated by [13] which proposed the triple $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ as a good set of features for controlling the camera translational dofs in a visual servoing loop. However, we empirically found this latter set to be ill-conditioned for what concerns the estimation of $\boldsymbol{\chi}$, with instead (12) providing enough information (i.e., full rankness of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$) for the estimation convergence[2].

Finally, analogously to the previous case, optimization of the structure estimation convergence from image moments can be achieved by maximizing w.r.t. $\boldsymbol{v}$ the smallest eigenvalue σ_1^2 of the square 3×3 matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (10).

[2]Alternatively, one could also resort to an adaptive/online selection of the best set of image moments as discussed in [14].

IV. DISCUSSION

Summarizing, although all presented methods are able to solve the plane estimation problem, the latter two have some advantages w.r.t. the (more classical) method A under several aspects. Indeed, as already explained, method B and method C, contrarily to method A, do not require the exact matching of point features across distant frames (initial and current ones), but only a frame-by-frame tracking. It is of course possible to reinitialize method A whenever the number of matched features becomes too small (i.e., by redefining ${}^0\mathcal{F}_C$ as the current camera frame), but this necessarily introduces an erratic transient phase due to the initial limited baseline.

In addition, method B and method C ought to be more robust w.r.t. non perfectly planar scenes because of their inherent filtering nature as compared to method A (a pure algebraic procedure). This is even more true for method C since method B still requires the algebraic resolution of (5). Finally, method C has also the additional advantage over method B of a reduced complexity in terms of number of estimated quantities: *three* (vector $\hat{\boldsymbol{\chi}}$) regardless of the number of tracked points, whereas method B needs to estimate N quantities for N tracked point features.

An advantage of method A lies, instead, in its "convergence rate": method A is (in principle) able to yield a good estimation of $\boldsymbol{n}$ as soon as a non-negligible displacement has taken place between the two frames ${}^0\mathcal{F}_C$ and $\mathcal{F}_C$, while the accuracy of both method B and method C clearly depends on the convergence rate of the estimation scheme (3).

V. EXPERIMENTAL RESULTS

This section reports some experimental results meant to illustrate and compare the three plane estimation methods introduced in the previous section. The experiments were run by employing a greyscale camera with a resolution of 640×480 px and a framerate of 30 fps. The camera was mounted on the end-effector of a 6-dofs Gantry robot commanded in velocity at a frequency of 100 Hz. Image processing and feature tracking were realized via the open-source ViSP library [15]. In the first set of experimets (Sects. V-A and V-B) a simple dotted pattern was used for feature extraction and matching. This solution was meant to reduce as much as possible the variability between each experimental run by ensuring tracking of the very same set of points across all trials. In the last experiment (Sect. V-C) a more realistic scene was considered with the Pyramidal Lucas Kanade feature tracker implemented in OpenCV used for tracking points on the surface of a (planar) topographic map (see Fig. 2). A video of the experiments is also attached to the paper.

As explained, the convergence rate of methods B and C was optimized by *actively* maximizing the minimum eigenvalue σ_m^2 in (8) for method B, and the smallest eigenvalue σ_1^2 of the 3×3 matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (10) for method C. Exploiting (4), and recalling that the Jacobian $\boldsymbol{J}_v$ can be computed in closed form in both cases, the following update

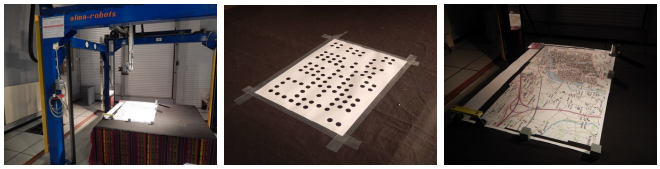

Fig. 2: Experimental set-up with the dotted pattern and the topographic map used for feature extraction and tracking.

rule was implemented for the camera linear velocity $\boldsymbol{v}$

$$\dot{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} k_1 \left(\kappa_{des} - \kappa\right) + k_2 \left(\boldsymbol{I} - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2}\right) \boldsymbol{J}_v^T \qquad (13)$$

with $\kappa(t) = \frac{1}{2}\|\boldsymbol{v}(t)\|^2$, $\kappa_{des} = \frac{1}{2}\|\boldsymbol{v}(t_0)\|^2$, $k_1 > 0$ and $k_2 \geq 0$. The first term in (13) asymptotically enforces $\|\boldsymbol{v}(t)\|^2 = \|\boldsymbol{v}(t_0)\|^2$ while the second term projects $\boldsymbol{J}_v^T$ onto the null-space of the constraint $\|\boldsymbol{v}\| = const$. Therefore, by means of (13) the direction of the camera velocity is optimized so as to maximize the SfM estimation convergence while keeping $\|\boldsymbol{v}(t)\| = const$.

As for the angular velocity $\boldsymbol{\omega}$, it was exploited, in the experiments in Sects. V-A and V-B, to keep the centroid of the tracked point features at the center of the image and, in the experiments in Sect. V-C, to align the camera optical axis with the (estimated) plane normal $\hat{\boldsymbol{n}}$. Indeed, we remark that matrix $\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v})$ in (2) *does not* depend on $\boldsymbol{\omega}$, see, e.g., (6–7) for the point feature case. Thus, when employing the SfM filter (3) for recovering the scene structure, one can freely choose the camera angular velocity without affecting the estimation convergence (which is essentially dictated by the norm of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ as discussed in Sect. III).

A. Experiments of plane estimation from 3D points (method B)

We report here the results in estimating the plane parameters $(\boldsymbol{n}, d)$ with method B. The experiment started from an initial guess $(\hat{\boldsymbol{n}}(t_0), \hat{d}(t_0))$ with an error of 40 deg w.r.t the true $\boldsymbol{n}(t_0)$ and a relative error of 50% w.r.t. the true $d(t_0)$. The initial depths of all the tracked points $\boldsymbol{p}_k$ were initialized so as to force $\hat{\boldsymbol{P}}_k(t_0)$ to belong to the estimated plane described by $(\hat{\boldsymbol{n}}(t_0), \hat{d}(t_0))$. In order to demonstrate the importance of the active camera velocity optimization, we first ran a set of four exeriments starting from the same initial conditions but using different initial camera velocities with the same norm $\|\boldsymbol{v}(t_0)\| = 0.0224m/s$. In these experiments we used $k_1 = 10$ in (13) but we either substituted $k_2\boldsymbol{J}_v^T$ with a random acceleration vector (purple dashed line) or we set $k_2 = 0$, thus keeping a $\boldsymbol{v}(t) = \boldsymbol{v}(t_0) = const$ during motion (green, red and cyan dashed lines). Finally we started the experiment again from the same initial camera velocity as in the experiment that performed worst in the previous set (cyan line) and we adopted the update rule (13) with $k_1 = 10$ and $k_2 = 1$.

Finally, for the sake of allowing a *fair* comparison between the convergence rates of method B and method C, we first collected all the data during a first execution of all trajectories, and then ran the two estimation schemes offline on the collected dataset by properly adjusting the estimation gains α_B and α_C of both methods. Indeed, let $\bar{\sigma}_m^2 = \frac{1}{T}\int_{t_0}^{t_0+T} \sigma_m^2(\tau)\mathrm{d}\tau$ and $\bar{\sigma}_1^2 = \frac{1}{T}\int_{t_0}^{t_0+T} \sigma_1^2(\tau)\mathrm{d}\tau$ be the average values of the eigenvalues $\sigma_m^2(t)$ and $\sigma_1^2(t)$ during motion in the active estimation cases, with T representing the experiment duration (blue lines in Figs. 3 and 4). After having computed $\bar{\sigma}_m^2$ and $\bar{\sigma}_1^2$ during the first run, the estimation gains α_B and α_C were chosen so as to satisfy $\alpha_B\bar{\sigma}_m^2 = \alpha_C\bar{\sigma}_1^2$ for imposing the same *closed-loop dynamics* to both method B and method C[3]. This resulted in gain $\alpha_B = 1043.4$ (used in these experiments), and in gain $\alpha_C = 20000$ (used in the experiments of the next Sect. V-B).

Fig. 3(a) shows the behavior of the norm of the estimation error $\boldsymbol{z}$ between the real and estimated inverse feature depths, normalized w.r.t. its initial value. The normalization is meant to allow a comparison of this plot with the analogous one in Fig. 4(a). The angle between vectors $\boldsymbol{n}(t)$ and $\hat{\boldsymbol{n}}(t)$ and the relative error $(\hat{d}(t) - d(t))/d(t)$ are also shown in the bottom plots. We can then note how the plane estimation task is solved in all cases (the estimation errors converge towards zero) but, clearly, in the active case (blue line) the error convergence is significantly faster than in the other experiments. This is further evident from Fig. 3(b) where the value of the $\alpha_B\sigma_m(t)$ is shown for all experiments (same color code): thanks to the active optimization of the direction of $\boldsymbol{v}(t)$, during the active experiment, $\alpha_B\sigma_m(t)$ results approximately 12.5 times larger than in the worst experiment (cyan) which started from the same initial camera velocity. Finally, Fig. 3(c) depicts the camera trajectory in all cases with arrows indicating the direction of the camera optical axis. The green patch represents the location of the plane to be estimated. We encourage the reader to also look at the attached video for better appreciating the effects of the active strategy on the camera trajectory.

B. Experiments of plane estimation from image moments (method C)

In this second set of experiments we show the results of using the weighted discrete image moments for the estimation of the plane parameters. As before the initial guess for $\hat{\boldsymbol{\chi}}(t_0)$ has an error of approximately 40° w.r.t. $\boldsymbol{n}(t_0)$ and a relative error of around 50% w.r.t. $d(t_0)$. Again, we first ran a set of four exeriments starting from the same initial conditions but using different initial camera velocities with the same norm $\|\boldsymbol{v}(t_0)\| = 0.0206m/s$ and using (13) with $k_1 = 10$ and either substituting $k_2\boldsymbol{J}_v^T$ with a random acceleration vector (purple dashed line) or setting $k_2 = 0$, thus keeping a $\boldsymbol{v}(t) = \boldsymbol{v}(t_0) = const$ during motion (green, red and cyan dashed lines). Finally, in the active case we started again from the initial camera velocity of the experiment that performed worst in the previous set (cyan line), and we adopted the update rule (13) with $k_1 = 10$ and $k_2 = 1$. In all cases, as explained in Sect. V-A, we set $\alpha_C = 20000$.

We show again in Fig. 4(a) the behavior of the normalized norm of the estimation error $\boldsymbol{z}$, the angle between the actual

[3]As explained in [8], the convergence rate of the SfM scheme (3) is actually dictated by the smallest eigenvalue of $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ *times* the chosen estimation gain α.

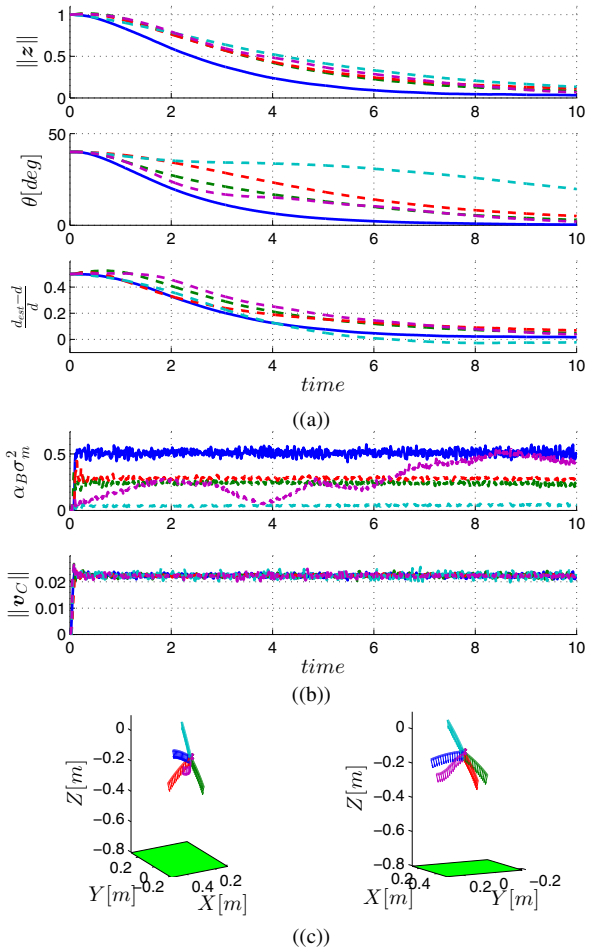

Fig. 3: Experimental results for the estimation of the plane parameters using 3D points (method B) with an active strategy (blue lines) or a random acceleration (purple line) or a constant linear velocity (green, red and cyan lines): ((a)) normalized norm of the estimation error $\boldsymbol{z}$, angle between the estimated and actual normal $\boldsymbol{n}$ and relative error between estimated and actual distance d; ((b)) smallest eigenvalue σ_m^2 of the $N \times N$ matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ multiplied by α_B and linear velocity norm; ((c)) geometric 3d trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

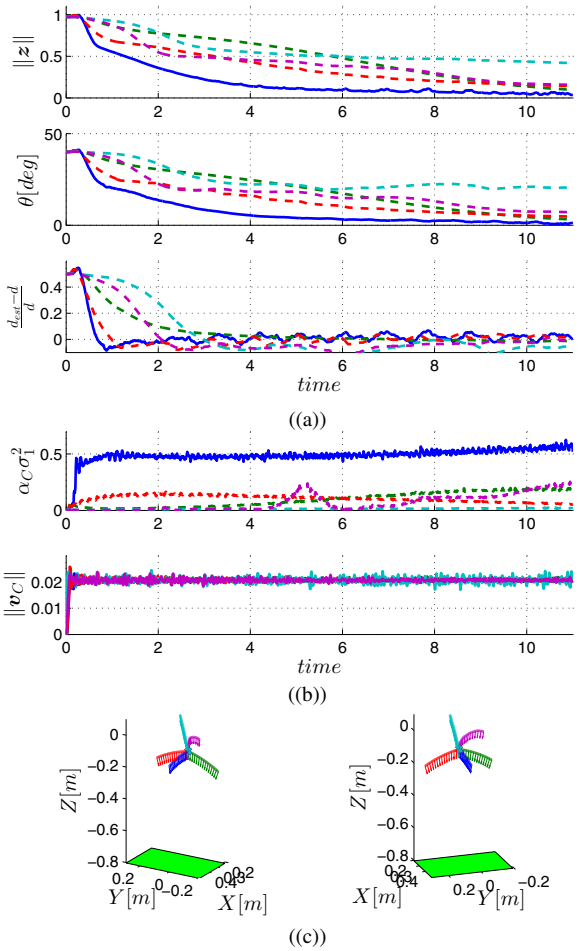

Fig. 4: Experimental results for the estimation of the plane parameters using image moments (method C) with an active strategy (blue lines) or a random acceleration (purple line) or a constant linear velocity (green, red and cyan lines): ((a)) normalized norm of the estimation error $\boldsymbol{z}$, angle between the estimated and actual normal $\boldsymbol{n}$ and relative error between estimated and actual distance d; ((b)) smallest eigenvalue σ_1^2 of the 3×3 matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ multiplied by α_C and linear velocity norm; ((c)) geometric 3d trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

and estimated normal direction $\boldsymbol{n}(t)$ and the relative error on $d(t)$. As evident from the plots, the active strategy results again in a faster convergence of the estimation error, as also clear from Fig. 4(b) where the behavior of $\alpha_C \sigma_1^2(t)$ is shown for all cases. The trajectory of the camera in the various experiments is finally shown in Fig. 4(c) (and as well in the attached video).

C. Comparison of the three methods A, B and C

This final set of experiments is meant to provide a comparative analysis of the differences between methods B and C against the classical method A taken as a *baseline condition*. As already discussed in Sect. IV, the 'convergence' of method A for the estimation of the plane normal direction is, in general, faster w.r.t. the other two methods due to its algebraic nature (no filtering process is present in this case). On the other hand the use of an estimation scheme in method B and method C allows for the possibility of tuning the estimation gain α (a free parameter) against the noise level present in the system (i.e., trading off convergence speed for noise robustness).

In order to test the three methods in a more challenging scenario, we added to the scene a small planar picture with a non-negligible inclination w.r.t. the main plane (see Fig. 5) so as to introduce the presence of some 'outliers' w.r.t. the main dominant plane[4]. The picture was located to be in visibility at the beginning of the experiment and to leave the camera field of view shortly after. The camera linear velocity was

[4]Of course one could utilize a RANSAC-based classification (exploiting the homography constraint) for preliminarily segmenting the two planes so as to only consider the points belonging to the main dominant plane for the estimation task. However, in a real situation, the accuracy of any classification method can never be perfect and some outliers will fail to be detected. Therefore, in this experiment we *intentionally* decided to not include any preliminary RANSAC-based pruning in order to just assess the "intrinsic" robustness of the proposed algorithms (which would clearly be improved by any preliminary outlier rejection step).

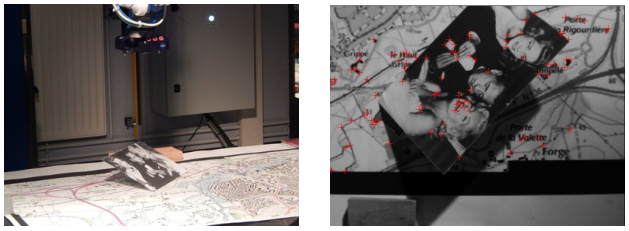

Fig. 5: Experimental setup for the estimation of the plane parameters in presence of outlier measurements. Note the introduction of the inclined picture in the observed scene on the right.

optimized via (13) by maximizing the smallest eigenvalue σ_1^2 of the matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ for the image moment case, and then the same trajectory was used for the other two methods. This resulted in a non-optimal, but still observable, trajectory for method B.

As done in the previous experimental sections, for the sake of obtaining a fair comparison between the convergence rates of method B and method C, we adjusted the estimation gains of both methods in such a way that $\alpha_B \bar{\sigma}_m^2 = \alpha_C \bar{\sigma}_1^2$, where $\bar{\sigma}_m^2$ and $\bar{\sigma}_1^2$ are the average values of the smallest eigenvalues for the two estimators along the (this time common) trajectory. This resulted in $\alpha_B = 200$ for method B and $\alpha_C = 26179$ for method C.

The behavior of the estimation error on the plane parameters is depicted in Fig. 6(a) for method A (green lines), method B (blue lines) and method C (red lines). In Fig. 6(b) the products $\alpha_B \sigma_m^2(t)$ and $\alpha_C \sigma_1^2(t)$ are plotted. It can be noticed that in all three cases at the beginning of the experiment (i.e. when the 'outlier' effect of the inclined image over the main planar scene is more present) the error in the estimation of the normal is significant although not diverging. All methods estimate a plane with an intermediate normal direction (as one would expect). Subsequently, the estimation errors for method C and method B start converging toward zero at $t \approx 8\boldsymbol{s}$ (first dashed vertical line), that is, when the outlier image starts leaving the image plane. However, note how the homography method still yields a very noisy estimation during this phase. Furthermore, once all the outliers are lost ($t \approx 20.3\boldsymbol{s}$ and second vertical dashed lines in the plots) all the methods yield a converging estimation error. However we can still notice two facts: (*i*) method C results in the fastest convergence. This is also because the weight w of the outliers starts approaching 0 as they get close to the image border (and thus their disturbing effect is more quickly discarded); (*ii*) method A has a faster convergence rate w.r.t. method B once all the outliers are lost, but it also yields a noisier estimation until the end of the experiment. In particular one can notice the presence of considerable "jumps" in the estimation of method A due to the reinitialization performed each time the number of matched features falls below a given threshold.

In order to demonstrate the effectiveness of the adopted weighting functions in the computation of the discrete moments, the behavior of $m_{00}(t)$ is shown in Fig. 6(b) (bottom plot) . This is meant to illustrate how the number of active points changes over time due to losses at the image border

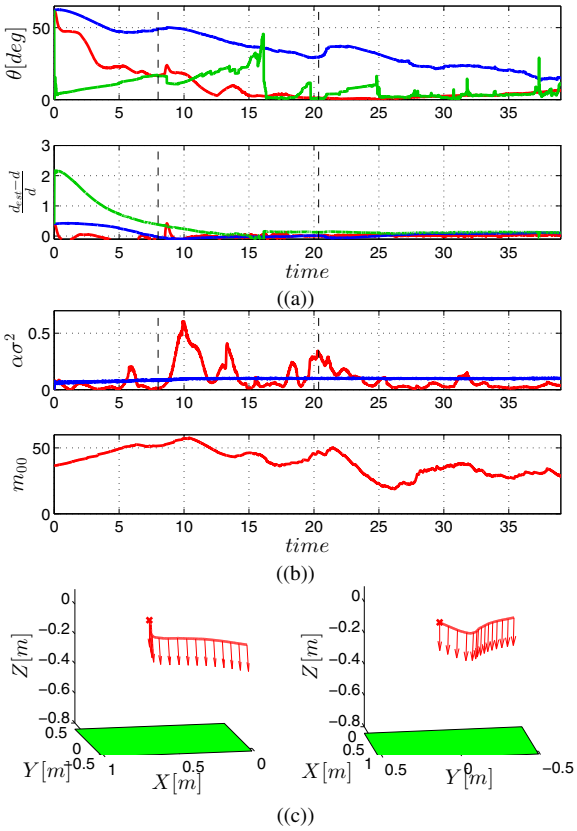

Fig. 6: Experimental results for the estimation in presence of outlier measurements using homography decomposition (method A – green lines), 3D points estimation (method B – blue lines) or image moments (method C – red lines): ((a)) angle between the estimated and actual normal $\boldsymbol{n}$ and relative error between estimated and actual distance d; ((b)) product $\alpha\sigma$ method B and method C and evolution of the image moment m_{00}; ((c)) geometric 3d trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

or detection of new features. The presence of the weighting strategy function guarantees the desired continuity of $m_{00}(t)$ (and similarly of all other image moments not plotted here).

Finally Fig. 7 shows the evolution of the individual switching functions $w_1(x)$, $w_2(y)$ and $w_3(\tau)$, and of their product $w = w_1 w_2 w_3$ for three representative point features. At $t \approx 3\,s$ the red feature starts leaving the image plane (first along the x direction and then along the y direction) and its total weight goes to zero by the action of both $w_1(x)$ and $w_2(y)$. After the feature has completely left the plane, the tracker detects a new feature (the blue one) at $t \approx 12\,s$. Being far from the image border, it is smoothly taken into account thanks to the effect of weight $w_3(t)$ (note the zoomed views on the right side of the plots where the smooth rise of $w_3(t)$ can be seen). Finally, the green feature is close to the border of the image at the time of detection. In this case, even if weight $w_3(t)$ is rising towards 1, the total wight of the feature is kept small by the action of $w_2(y)$.

VI. CONCLUSIONS

In this paper we presented and critically compared via several experiments three methods for estimating the 3D

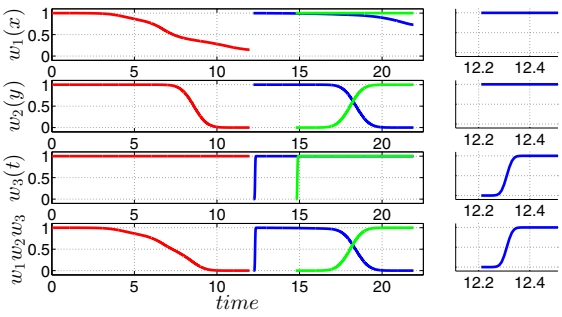

Fig. 7: Evolution of the switching funtions and of their product for three representative point features. On the right: detailed views of the corresponding plots on the left in the time interval immediately following the introduction of the blue feature in the estimator.

parameters $(\boldsymbol{n},\, d)$ of a plane from a set of (possibly time-varying) point features tracked by a moving camera. The first one (the *baseline* method) is based on the classical decomposition of the homography constraint, while the other two methods exploit an active SfM algorithm for estimating the 3D structure of the scene: the depths of the tracked point features for then extracting $(\boldsymbol{n},\, d)$ in one case, and directly the plane parameters $(\boldsymbol{n},\, d)$ from the measured image moments in the other case. In both these latter methods an active strategy is also presented for controlling *online* the camera motion in order to optimize the estimation convergence speed. Furthermore, the possible loss/gain of point features over time because of the limited camera fov is considered in all three methods. For the third (moment-based) case this required the extension to the case of *weighted* image moments to smoothly take into account new/lost features.

The reported experiments confirmed the effectiveness of these methods in estimating $(\boldsymbol{n},\, d)$ in real conditions, also in the (intentional) presence of some outliers w.r.t. the main dominant plane. In particular, the third (moment-based) method resulted the most robust against outliers because of its better filtering capabilities (as expected). Motivated by these findings, we are currently investigating the possibility of extending our results to the case of dense photometric moments. Moreover it would also be interesting to combine the proposed methods with some (RANSAC-based) classification technique for outlier rejection and plane clustering so as to address the issue of estimating the parameters of multiple planes at the same time.

APPENDIX

Let

$$\begin{cases} m^x_{ij} &= \displaystyle\sum_{k=1}^{\infty} \frac{\partial w}{\partial x}(x_k,\, y_k,\, t-t_k) x_k^i y_k^j \\ m^y_{ij} &= \displaystyle\sum_{k=1}^{\infty} \frac{\partial w}{\partial y}(x_k,\, y_k,\, t-t_k) x_k^i y_k^j \\ m^t_{ij} &= \displaystyle\sum_{k=1}^{\infty} \frac{\partial w}{\partial t}(x_k,\, y_k,\, t) x_k^i y_k^j \end{cases} \tag{14}$$

and $\boldsymbol{\chi} = \boldsymbol{n}/d = (A,\, B,\, C)$. By leveraging on the developments of [13], the dynamics of the $(i,\, j)$-th weighted moment (11) is given by

$$\dot{m}_{ij} = [m_{vx}\; m_{vy}\; m_{vz}\; m_{wx}\; m_{wy}\; m_{wz}] \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} + m^t_{ij} \tag{15}$$

with

$$\begin{aligned} m_{vx} &= A(-im_{i,j} - m^x_{i+1,j}) + B(-im_{i-1,j+1} - m^x_{i,j+1}) \\ &\quad + C(-im_{i-1,j} - m^x_{i,j}) \\ m_{vy} &= A(-jm_{i+1,j-1} - m^y_{i+1,j}) + B(-jm_{i,j} - m^y_{i,j+1}) \\ &\quad + C(-jm_{i,j-1} - m^y_{i,j}) \\ m_{vz} &= A(jm_{i+1,j} + im_{i+1,j} + m^x_{i+2,j} + m^y_{i+1,j+1}) \\ &\quad + B(im_{i,j+1} + jm_{i,j+1} + m^x_{i+1,j+1} + m^y_{i,j+2}) + \\ &\quad + C(jm_{i,j} + im_{i,j} + m^x_{i+1,j} + m^y_{i,j+1}) \\ m_{wx} &= jm_{i,j+1} + im_{i,j+1} + jm_{i,j-1} + m^x_{i+1,j+1} + m^y_{i,j} + m^y_{i,j+2} \\ m_{wy} &= -im_{i+1,j} - jm_{i+1,j} - im_{i-1,j} - m^x_{i,j} - m^y_{i+1,j+1} - m^x_{i+2,j} \\ m_{wz} &= im_{i-1,j+1} - jm_{i+1,j-1} - m^y_{i+1,j} + m^x_{i,j+1}. \end{aligned}$$

We can note that the dynamics (15) involves moments of order up to $(i+j+2)$ associated to the terms m^x_{ij} and m^y_{ij}. Also, it is easy to check that in the unweighted case ($w \equiv 1$ and $m^x_{ij} = m^y_{ij} = m^t_{ij} \equiv 0$), one obviously recovers the classical moment dynamics reported in [13].

REFERENCES

[1] K. Watanabe, R. Kawanishi, A. Yamashita, Y. Kobayashi, T. Kaneko, and H. Asama, "Mobile Robot Navigation in Textureless Unknown Environment Based on Plane Estimation by Using Single Omni-Directional Camera," in *System Integration (SII), 2012 IEEE/SICE International Symposium on*, Dec 2012, pp. 37–42.

[2] 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.

[3] 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.

[4] 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.

[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, pp. 7647–7653.

[9] P. Robuffo Giordano, R. Spica, and F. Chaumette, "An Active Strategy for Plane Detection and Estimation for a Monocular Camera," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014.

[10] M. Bakthavatchalam, F. Chaumette, and O. Tahri, "An Improved Modelling Scheme for Photometric Moments with Inclusion of Spatial Weights for Visual Servoing with Partial Appearance/Disappearance," in *2015 IEEE Int. Conf. on Robotics and Automation*, 2015.

[11] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Globally Asymptotically Stable Sensor-Based Simultaneous Localization and Mapping," *IEEE Trans. on Robotics*, vol. 29, no. 6, pp. 1380–1395, Dec 2013.

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

[13] O. Tahri and F. Chaumette, "Point-Based and Region-Based Image Moments for Visual Servoing of Planar Objects," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.

[14] P. Robuffo Giordano, R. Spica, and F. Chaumette, "Learning the Shape of Image Moments for Optimal 3D Structure Estimation," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015.

[15] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.