Learning the Shape of Image Moments for Optimal 3D Structure Estimation

Paolo Robuffo Giordano, Riccardo Spica, and François Chaumette

***Abstract*— The selection of a suitable set of visual features for an optimal performance of closed-loop visual control or Structure from Motion (SfM) schemes is still an open problem in the visual servoing community. For instance, when considering integral region-based features such as image moments, only heuristic, partial, or local results are currently available for guiding the selection of an appropriate moment set. The goal of this paper is to propose a novel learning strategy able to *automatically* optimize online the shape of a given class of image moments as a function of the observed scene for improving the SfM performance in estimating the scene structure. As case study, the problem of recovering the (unknown) 3D parameters of a planar scene from measured moments and known camera motion is considered. The reported simulation results fully confirm the soundness of the approach and its superior performance over more consolidated solutions in increasing the information gain during the estimation task.**

I. INTRODUCTION

The quest for finding a good set of features for visual control and 3D Structure from Motion (SfM) is a classical problem in the visual servoing community, and it has indeed attracted a large body of literature over the last decades. A number of approaches has been proposed over the years for exploiting *local* geometrical primitives, such as points or lines tracked on the image, as visual features to be measured/controlled. A comprehensive overview of these possibilities can be found in [1]. In parallel, another very successful line of research has considered the use of more 'integral' image descriptors able to encode the information contained over a *region* of interest on the image plane (e.g., enclosed by the contour of a tracked object). This indeed usually results in a (relatively) easier extraction, matching and spatio-temporal tracking across multiple frames of the region of interest, thus generally improving the robustness against image processing errors.

Image moments of binary dense closed regions or of discrete sets of points [2] are a typical (and by now classical) example of *integral* features exploited for visually controlling the camera pose [3]–[5]. Further extensions of these ideas have dealt with, e.g., the use of particular kernels for evaluating the image moments [6], or the direct use of pixel intensities [7] by processing the whole acquired image. Image moments from dense regions [8] or discrete point clouds [9], [10] have also been exploited as visual features for recovering the 3D structure of a planar scene via SfM schemes.

Despite this state-of-the-art, it is however worth noting that the selection of a good set of image moments for 6-dof visual control or SfM is still an open problem. Ideally, one would like to find a unique set of visual features resulting in the 'most linear' control problem with the largest convergence domain, or in maximum observability (i.e., information gain) for a given camera displacement in case of SfM tasks. However, to the best of our knowledge, only local, partial (e.g., depending of the particular shape of the object) or heuristic results are currently available. For instance, [2], [6], [7] propose different combinations of image moments able to only guarantee *local* 6-dof stability of the servoing loop around the desired pose, and with a basin of attraction to be heuristically determined case by case. As for what concerns the SfM case, the choice of which moments to exploit for allowing a converging estimation of the scene structure is also not straightforward. In [8], [11] the area a and barycenter coordinates (x_g, y_g) of a dense region are successfully fed to a SfM scheme based on the (intuitive) motivation that the same set (a, x_g, y_g) is also the typical choice for *controlling* the camera translational motion in a servoing loop [2]. However, this intuition breaks down when considering moments of a discrete point cloud: in this case, the typical choice for *controlling* the camera translational motion, that is, the set $(x_g, y_g, \mu_{20} + \mu_{02})$ (see [2]), is empirically shown in [10] to not provide enough information for allowing a converging estimation of the scene structure.

One could argue that the hope of finding a unique set of visual features optimal in all situations might eventually prove to be unrealistic, if not impossible, while it could just be more appropriate (and reasonable) to rely on an *automatic* and *online* selection of the best feature set (within a given class) tailored to the particular task at hand. Motivated by these considerations, the goal of this work is to propose an *automatic* learning strategy able to select *online* the 'best' set of image moments in order to optimize the SfM performance in recovering the (unknown) 3D parameters of a planar scene.

The rest of the paper is organized as follows: first some preliminary concepts of interests are reviewed in Sect. II, and then the definition of *weighted parametric moment*, central for the paper developments, is introduced in Sect. III. Section IV discusses a possible strategy for automatically adjusting the shape of the weighted moments so as to maximize an 'observability measure' (or information gain) of the chosen SfM task, and Sect. V reports the results of several

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

simulations where the benefits of the proposed approach can be clearly appreciated. Finally, Sect. VI concludes the paper and draws some future perspectives.

II. PRELIMINARIES

Consider a planar scene with equation $\mathcal{P}: \ \boldsymbol{n}^T\boldsymbol{E}+d=0$, with $\boldsymbol{n}\in\mathbb{S}^2$ being the unit normal vector and $d\in\mathbb{R}$ the plane distance from the camera center, and let $(\boldsymbol{v},\,\boldsymbol{\omega})\in\mathbb{R}^6$ represent the linear/angular camera velocity[1]. The $(i,\,j)$-th image moment m_{ij} of a collection of N point features $\boldsymbol{p}_k=(x_k,\,y_k,\,1)$ tracked on $\mathcal{P}$ is defined as

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

Furthermore, $x_g=m_{10}/N$ and $y_g=m_{01}/N$ denote the barycenter coordinates and

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

the $(i,\,j)$-th *centered* moment.

By letting $\boldsymbol{\chi}=-\boldsymbol{n}/d\in\mathbb{R}^3$, the dynamics of m_{ij} is known to take the following expression linear in $\boldsymbol{\chi}$

$$\dot{m}_{ij}=\boldsymbol{f}_{\omega_{ij}}(m_{kl},\,\boldsymbol{\omega})+\boldsymbol{f}_{\chi_{ij}}(m_{kl},\,\boldsymbol{v})\boldsymbol{\chi}, \tag{2}$$

see [2] for further details. In (2), m_{kl} stands for a generic $(k,\,l)$-th moment of order up to $i+j+1$ and $\boldsymbol{\chi}$ represents the 3D structure of the observed planar scene $\mathcal{P}$ (not directly measurable from sole image quantities). A conceptually equivalent derivation can also be obtained for the barycenter, for the centered moments, and for the case of image moments of dense planar regions [2].

Equation (2) is at the core of virtually all algorithms for 3D structure estimation with image moments playing the role of measurements. For instance, owing to the linearity of (2) w.r.t. $\boldsymbol{\chi}$, several SfM schemes meant to recover the (unmeasurable) 3D structure $\boldsymbol{\chi}$ from the measured $\boldsymbol{s}(t)$ and the known camera motion $(\boldsymbol{v},\,\boldsymbol{\omega})$ have been proposed in [8]–[10]. However, as discussed in the Introduction, a satisfactory SfM performance requires the *preliminary identification* of a suitable set of image moments 'rich enough' for ensuring observability of the scene structure. Specifically, given a collection of m measured moments $\boldsymbol{s}=(m_{i_1j_1},\ldots,\,m_{i_mj_m})$ and defining (using (2))

$$\boldsymbol{\Omega}=\left[\boldsymbol{f}^T_{\chi_{i_1j_1}}(m_{kl},\,\boldsymbol{v})\ \ldots\ \boldsymbol{f}^T_{\chi_{i_nj_n}}(m_{kl},\,\boldsymbol{v})\right]\in\mathbb{R}^{3\times m}, \tag{3}$$

the 'observability matrix' $\boldsymbol{\Omega}\boldsymbol{\Omega}^T\in\mathbb{R}^{3\times3}$ must remain full rank during the camera motion [11].

The weighted parametric moments introduced hereafter are meant to provide an *adjustable* visual feature set that can be tuned online for automatically coping with this observability requirement.

[1]All quantities are expressed in the camera frame, and the camera is assumed calibrated.

III. DEFINITION OF WEIGHTED PARAMETRIC IMAGE MOMENTS

Let $w=w(x,\,y,\,\boldsymbol{\theta})$ be a smooth function of the coordinates $(x,\,y)$ on the image plane and of a vector of parameters $\boldsymbol{\theta}\in\mathbb{R}^p$. One can generalize (1) and define a *weighted parametric* image moment for N observed features $\boldsymbol{p}_k$

$$m_w(\boldsymbol{\theta})=\sum_{k=1}^{N}w(x_k,\,y_k,\,\boldsymbol{\theta}), \tag{4}$$

with, obviously, $m_w=m_{ij}$ for $w(x,\,y,\,\boldsymbol{\theta})=x^iy^j$. Function $w(x,\,y,\,\boldsymbol{\theta})$ can be seen as the *class* of all the considered image moments (e.g., a quadratic form in $x,\,y$) parameterized by vector $\boldsymbol{\theta}$ (e.g., the coefficients of the quadratic form). Consider now the following additional definitions

$$\begin{aligned}
m^x_{w_{ij}}(\boldsymbol{\theta}) &= \sum_{k=1}^{N}x_k^iy_k^j\left.\frac{\partial w(x,\,y,\,\boldsymbol{\theta})}{\partial x}\right|_{(x_k,\,y_k)}\\
m^y_{w_{ij}}(\boldsymbol{\theta}) &= \sum_{k=1}^{N}x_k^iy_k^j\left.\frac{\partial w(x,\,y,\,\boldsymbol{\theta})}{\partial y}\right|_{(x_k,\,y_k)}\\
\boldsymbol{m}^{\boldsymbol{\theta}}_w(\boldsymbol{\theta}) &= \sum_{k=1}^{N}\left.\frac{\partial w(x,\,y,\,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{(x_k,\,y_k)},
\end{aligned} \tag{5}$$

and note that $\boldsymbol{m}^{\boldsymbol{\theta}}_w$ is a row vector of dimension p. Following the derivations in [2], it is easy to show that the dynamics of $m_w(\boldsymbol{\theta})$ takes the expression (reminiscent of (2))

$$\begin{aligned}
\dot{m}_w(\boldsymbol{\theta})=&[m_A(\boldsymbol{v},\,\boldsymbol{\theta})\quad m_B(\boldsymbol{v},\,\boldsymbol{\theta})\quad m_C(\boldsymbol{v},\,\boldsymbol{\theta})]\boldsymbol{\chi}\\
&+[m_{\omega_x}(\boldsymbol{\theta})\quad m_{\omega_y}(\boldsymbol{\theta})\quad m_{\omega_z}(\boldsymbol{\theta})]\boldsymbol{\omega}+\boldsymbol{m}^{\boldsymbol{\theta}}_w(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}
\end{aligned} \tag{6}$$

with

$$\begin{cases}
m_A=-m^x_{w_{10}}v_x-m^y_{w_{10}}v_y+(m^x_{w_{20}}+m^y_{w_{11}})v_z\\
m_B=-m^x_{w_{01}}v_x-m^y_{w_{01}}v_y+(m^x_{w_{11}}+m^y_{w_{02}})v_z\\
m_C=-m^x_{w_{00}}v_x-m^y_{w_{00}}v_y+(m^x_{w_{10}}+m^y_{w_{01}})v_z\\
m_{\omega_x}=(m^x_{w_{11}}+m^y_{w_{02}}+m^y_{w_{00}})\\
m_{\omega_y}=(-m^x_{w_{00}}-m^x_{w_{20}}-m^y_{w_{11}})\\
m_{\omega_z}=(m^x_{w_{01}}-m^y_{w_{10}})
\end{cases}, \tag{7}$$

and $\boldsymbol{\chi}=(A,\,B,\,C)=-\boldsymbol{n}/d$.

One can then exploit (4–7) for implementing a visual control or estimation algorithm as in the classical case, but with the *additional* possibility of acting on vector $\boldsymbol{\theta}$ (a free parameter) for optimizing any criterium of interest (e.g., the norm of the observability matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ during an estimation task as it will be discussed in Sect. IV).

A. Design of the weighting function $w(x,\,y,\,\boldsymbol{\theta})$

Clearly, there exist many possibilities for designing the weighting function $w(\cdot)$, i.e., the class of moments spanned by vector $\boldsymbol{\theta}$. A convenient choice, in our opinion, is to take $w(\cdot)$ as some *polynomial* basis in x and y with $\boldsymbol{\theta}$ being the vector of coefficients. Indeed, in this way the weighted moments (4), the expressions in (5) and, eventually, all the terms in (7) will reduce to linear combinations of the *unweighted moments* m_{ij} in (1). The overall computational complexity will then result equivalent to the classical case [2].

As for which polynomial basis to exploit, many choices are possible depending on the constraints/requirements of the particular application. Within the scope of this work, two possibilities are considered:

1) Polynomial basis of fixed degree: first, one can take $w(\cdot)$ as a polynomial in x and y of a given *degree* $\delta \in \mathbb{N}^+$, that is,

$$w(x,\, y,\, \boldsymbol{\theta}) = \sum_{j=1}^{\delta} \sum_{k=0}^{j} \theta_{T_j+k} x^{(j-k)} y^k \tag{8}$$

with $T_j = \binom{j+1}{2}$ and $\boldsymbol{\theta} = (\theta_1, \ldots \ldots \theta_{T_\delta+\delta}) \in \mathbb{R}^{T_\delta+\delta}$. Indeed, this allows (4) to span all the moment linear combinations of order up to δ with coefficients in vector $\boldsymbol{\theta}$. As illustration, by choosing $\delta = 2$ in (8), one obtains the following quadratic polynomial

$$w(x,\, y,\, \boldsymbol{\theta}) = \theta_1 x + \theta_2 y + \theta_3 x^2 + \theta_4 xy + \theta_5 y^2$$

that, when plugged in (4), yields

$$m_w(\boldsymbol{\theta}) = \theta_1 m_{10} + \theta_2 m_{01} + \theta_3 m_{20} + \theta_4 m_{11} + \theta_5 m_{02}. \tag{9}$$

The class (9) can then specialize into, e.g., the barycenter coordinate x_g for $\boldsymbol{\theta} = (1/N,\, 0,\, 0,\, 0,\, 0)$, the centered moment μ_{02} for $\boldsymbol{\theta} = (0,\, -y_g,\, 0,\, 0,\, 1)$, and so on. Clearly, the larger the value of the degree δ, the richer the basis representation power in encoding the scene geometry, but at the (well-known) cost of an increasing noise level with the moment order.

2) Constrained polynomial basis: a second possibility is to design a *constrained* polynomial basis for coping with the possible loss/gain of point features during the camera motion because of the limited camera field of view (fov). Indeed, by imposing that $w(\cdot)$ vanishes (with vanishing derivative) at the image borders, any point feature close to the limits will *smoothly* enter or leave the image plane and, thus, prevent any discontinuity in the moment dynamics (6).

Let then $x_{min} < x_{max}$ and $y_{min} < y_{max}$ represent the limits of a rectangular image plane, and consider a weighting function $w(\cdot)$ partitioned as

$$w(x,\, y,\, \boldsymbol{\theta}) = w^x(x, \boldsymbol{\theta}^x) w^y(y, \boldsymbol{\theta}^y), \tag{10}$$

where $w^x(x, \boldsymbol{\theta}^x)$ and $w^x(x, \boldsymbol{\theta}^y)$ are polynomial bases and $\boldsymbol{\theta} = (\boldsymbol{\theta}^x,\, \boldsymbol{\theta}^y) \in \mathbb{R}^{p_x+p_y}$, $p_x + p_y = p$, is the vector of coefficients. Assuming $p^x \geq 4$ and imposing

$$\begin{cases} w^x(x_{min}, \boldsymbol{\theta}^x) &= w^x(x_{max}, \boldsymbol{\theta}^x) = 0 \\ \left.\dfrac{\partial w^x(x, \boldsymbol{\theta}^x)}{\partial x}\right|_{x_{min}} &= \left.\dfrac{\partial w^x(x, \boldsymbol{\theta}^x)}{\partial x}\right|_{x_{max}} = 0 \end{cases}, \tag{11}$$

one can solve for a set of 4 parameters in vector $\boldsymbol{\theta}^x$ for shaping $w^x(x, \boldsymbol{\theta}^x)$ as desired. For instance, by taking

$$w^x(x,\, \boldsymbol{\theta}^x) = \theta_1^x x^5 + \theta_2^x x^4 + \theta_3^x x^3 + \theta_4^x x^2 + \theta_5^x x + \theta_6^x \tag{12}$$

and by (arbitrarily) choosing the pair $(\theta_1^x,\, \theta_2^x)$ as free parameters in vector $\boldsymbol{\theta}^x$, system (11) yields

$$\begin{cases} \theta_3^x &= (-3x_{max}^2 - 3x_{min}^2 - 4x_{max}x_{min})\theta_1^x \\ &\quad +(-2x_{min} - 2x_{max})\theta_2^x \\ \theta_4^x &= (2x_{max}^3 + 8x_{max}^2 x_{min} + 8x_{max}x_{min}^2 + 2x_{min}^3)\theta_1^x \\ &\quad +(x_{max}^2 + 4x_{max}x_{min} + x_{min}^2)\theta_2^x \\ \theta_5^x &= (-7x_{max}^2 x_{min}^2 - 4x_{min}^3 x_{max} - 4x_{max}^3 x_{min})\theta_1^x \\ &\quad +(-2x_{max}x_{min}^2 - 2x_{max}^2 x_{min})\theta_2^x \\ \theta_6^x &= (2x_{max}^3 x_{min}^2 + 2x_{max}^2 x_{min}^3)\theta_1^x + x_{max}^2 x_{min}^2 \theta_2^x \end{cases}. \tag{13}$$

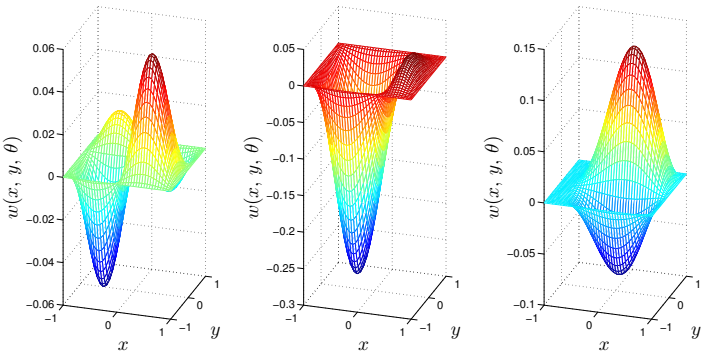


Fig. 1: Three examples of the constrained polynomial basis $w(x,\, y,\, \boldsymbol{\theta})$ in (10–13). Note how $w(x,\, y,\, \boldsymbol{\theta})$ smoothly vanishes at the image borders of size $[-1,\, 1] \times [-1,\, 1]$

Imposing analogous conditions to function $w^y(y, \boldsymbol{\theta}^y)$ at y_{min} and y_{max} (with again $p^y \geq 4$) will then constrain a total of 8 parameters in vector $\boldsymbol{\theta}$, with the remaining $p-8$ coefficients still free to be exploited for optimization purposes. For the sake of illustration, Fig. 1 shows three examples of weighting functions $w(\cdot)$ smoothly vanishing at the borders of an image plane of size $[-1,\, 1] \times [-1,\, 1]$ and obtained by picking at random three values for the free parameters in vector $\boldsymbol{\theta}$.

We conclude by noting that, compared to the previous case (8), this latter possibility necessitates of a polynomial basis (10) with a degree of at least 7. Indeed, as explained, the vanishing conditions at the image border will constrain $4+4$ coefficients in $\boldsymbol{\theta}^x$ and $\boldsymbol{\theta}^y$, thus forcing both $w^x(x, \boldsymbol{\theta}^x)$ and $w^y(y, \boldsymbol{\theta}^y)$ to have a degree of (at least) 3 for ensuring $p_x \geq 4$ and $p_y \geq 4$ as required (see (12)). However, for any *optimization* of the coefficient vector $\boldsymbol{\theta}$ to be possible, either $p_x > 4$ or $p_y > 4$ must hold for allowing presence of at least *one free coefficient* to be optimized besides those already constrained by the vanishing conditions. On the other hand, if either $p_x > 4$ or $p_y > 4$, the final polynomial basis (10) will necessarily result of at least degree 7.

Therefore, the use of higher-order moments (of at least order 7) is the 'price to pay' for smoothly taking into account the loss/gain of point features during the camera motion[2]. In contrast, the degree of the polynomial basis in (8) can be chosen at will and thus adjusted, if necessary, for limiting the noise level in the measured moments.

IV. OPTIMIZATION OF THE WEIGHTED PARAMETRIC IMAGE MOMENTS

Let again $\boldsymbol{s} \in \mathbb{R}^m$ represent the set of measured image moments. As explained in Sect. II (and discussed in more detail in [11]), when attempting to estimate the 3D structure $\boldsymbol{\chi}$ from the measured $\boldsymbol{s}$ (and the known camera motion $(\boldsymbol{v},\, \boldsymbol{\omega})$), full rankness of the square matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (3) plays the role of a necessary (observability) condition for ensuring a converging estimation. Since in all SfM problems one has $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v})$ (see again [11]), it is possible to act on the camera linear velocity $\boldsymbol{v}$ in order to maximize some

[2]Of course, the use of different functional bases, also non-polynomial, could be possible.

conditioning measure of $\mathbf{\Omega\Omega}^T$ for increasing the information gain during the camera motion. This insight has motivated some recent work in the context of *active* Structure from Motion for planar and 3D scenes [10]–[12].

Consider now a set of $m \geq 3$ *weighted* moments

$$\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1), \ldots, m_w(\boldsymbol{\theta}_m)) \in \mathbb{R}^m$$

with $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) \in \mathbb{R}^p$ being the stack of all parameters. Plugging the weighted moment dynamics (6–7) in the definition (3), one has

$$\mathbf{\Omega}(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}) = \begin{bmatrix} m_A(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_A(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \\ m_B(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_B(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \\ m_C(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_C(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \end{bmatrix} \in \mathbb{R}^{3\times m}. \tag{14}$$

Therefore, when employing the weighted parametric moments (4) instead of the classical moments (1), one gains the additional possibility of *also* acting on vector $\boldsymbol{\theta}$ (i.e., on the 'moment shape') for affecting matrix $\mathbf{\Omega\Omega}^T$.

Different scalar quantities can be taken as a measure of the conditioning of the square (and semi-positive definite) matrix $\mathbf{\Omega\Omega}^T$. For instance, the analysis in [11] shows that its smallest eigenvalue σ_1^2 directly affects the convergence rate of the employed estimator, and thus a reasonable choice is to maximize σ_1^2 over time. However, the evaluation of the derivative/gradient of an eigenvalue is unfortunately not well-defined for repeated eigenvalues [13]. In order to avoid this issue, in this work we chose to take the quantity

$$\rho = \det(\mathbf{\Omega\Omega}^T) \tag{15}$$

as a conditioning measure for matrix $\mathbf{\Omega\Omega}^T$. Indeed, from classical linear algebra [14] the following relationship holds for a square matrix $\boldsymbol{A}$

$$\frac{\mathrm{d}\det(\boldsymbol{A})}{\mathrm{d}t} = \mathrm{tr}\left(\mathrm{adj}(\boldsymbol{A})\frac{\mathrm{d}(\boldsymbol{A})}{\mathrm{d}t}\right) \tag{16}$$

with $\mathrm{tr}(\cdot)$ and $\mathrm{adj}(\cdot)$ being the *trace* and *adjugate* operators, respectively. Contrarily to the derivative of an eigenvalue, the relationship (16) is always well-defined with, in particular, no possible ill-conditioning due to repeated eigenvalues. By then applying (16) to matrix $\mathbf{\Omega\Omega}^T$ and expanding the various terms, one obtains

$$\begin{aligned}\dot{\rho} = &\sum_i \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial v_i}\right)\dot{\boldsymbol{v}}_i + \sum_i \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial \theta_i}\right)\dot{\theta}_i \\ &+ \sum_i \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial s_i}\right)\dot{s}_i = \boldsymbol{J}_v\dot{\boldsymbol{v}} + \boldsymbol{J}_\theta\dot{\boldsymbol{\theta}} + \boldsymbol{J}_s\dot{\boldsymbol{s}}\end{aligned} \tag{17}$$

where the Jacobian matrixes

$$\begin{aligned}\boldsymbol{J}_v &= \left[\ldots \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial v_i}\right)\ldots\right] \in \mathbb{R}^{1\times 3} \\ \boldsymbol{J}_\theta &= \left[\ldots \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial \theta_i}\right)\ldots\right] \in \mathbb{R}^{1\times p} \\ \boldsymbol{J}_s &= \left[\ldots \mathrm{tr}\left(\mathrm{adj}\left(\mathbf{\Omega\Omega}^T\right)\frac{\partial(\mathbf{\Omega\Omega}^T)}{\partial s_i}\right)\ldots\right] \in \mathbb{R}^{1\times m}\end{aligned} \tag{18}$$

are function of $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta})$ (all available quantities). We stress that all the terms in (18) can be computed in closed-form.

The relation (17) can then be exploited for affecting $\rho(t)$ over time by acting on $\dot{\boldsymbol{v}}$ (the camera linear acceleration) *and/or* $\dot{\boldsymbol{\theta}}$ (the parameter vector). Among the many possibilities, we considered here the following update rules

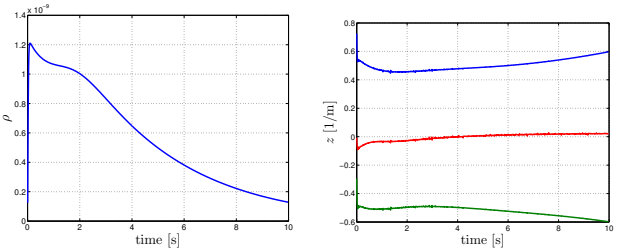
$$\begin{cases} \dot{\boldsymbol{v}} &= k_v\left(\boldsymbol{I} - \dfrac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\right)\boldsymbol{J}_v^T \\ \dot{\boldsymbol{\theta}} &= k_\theta\left(\boldsymbol{I} - \dfrac{\boldsymbol{\theta}\boldsymbol{\theta}^T}{\boldsymbol{\theta}^T\boldsymbol{\theta}}\right)\boldsymbol{J}_\theta^T \end{cases}, \qquad k_v > 0,\ k_\theta > 0 \tag{19}$$

which are meant to maximize $\rho(t)$ by following its gradient w.r.t. $(\boldsymbol{v}, \boldsymbol{\theta})$ projected on the null-spaces of the constant-norm constraints $\|\boldsymbol{v}(t)\| = const$ and $\|\boldsymbol{\theta}(t)\| = const$. As explained in [11], the constraint $\|\boldsymbol{v}(t)\| = const$ is meant to prevent a better conditioning of matrix $\mathbf{\Omega\Omega}^T$ *only* due to a faster camera motion while observing the scene. Indeed, since, roughly speaking, $\|\mathbf{\Omega\Omega}^T\|$ is monotonically increasing with $\|\boldsymbol{v}\|^2$, the faster the camera motion the faster the SfM convergence regardless of any other optimization action. The second constraint $\|\boldsymbol{\theta}(t)\| = const$ is motivated by similar arguments: an increasing $\|\boldsymbol{\theta}(t)\|$ would (artificially) magnify $\rho(t)$ at the cost of an increased noise level (all the terms in (7) would just result amplified).

The optimization action (19) will then maximize the observability measure $\det(\mathbf{\Omega\Omega}^T)$ for the SfM task at hand by (i) adjusting the direction of the camera linear velocity $\boldsymbol{v}$ and, *at the same time*, (ii) by adapting the shape of the m weighted moments $\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1), \ldots, m_w(\boldsymbol{\theta}_m))$ as only a function of the perceived scene and camera motion. We also remark that (19) (or any other equivalent strategy) assumes the possibility of acting at will on the direction of the linear camera velocity $\boldsymbol{v}$. There could be cases where this is not (fully) possible, and $\boldsymbol{v}$ (or components of it) are given (for example during a combined estimation/servoing loop as in [15]). In all these cases, it is obviously still possible to just keep on optimizing $\boldsymbol{\theta}(t)$ during the (given/known) camera motion in order to adapt, as best as possible, the moment shape. Finally, since $\mathbf{\Omega}(t)$ (and thus $\rho(t)$) does *not* depend on the camera angular velocity, one can freely choose $\boldsymbol{\omega}$ to fulfil any additional goal of interest.

V. SIMULATION RESULTS

All the following simulations consider a free-flying camera observing a planar scene $\mathcal{P}$ consisting of $N = 30$ points, and with plane parameters $\boldsymbol{n} = (0, 0, -1)$ and $d = 1.5$ [m] in ${}^0\mathcal{F}_C$ (the initial camera frame at $t = t_0$). The initial estimations of the plane normal and distance are always taken as $\hat{\boldsymbol{n}}(t_0) = (-0.87,\ 0,\ -0.49)$ and $\hat{d}(t_0) = 1$ [m], thus representing an initial incertitude of ≈ 60 [deg] on the real normal direction and of 0.5 [m] on the real distance to the plane. Finally, the point features $\boldsymbol{p}_k$, $k = 1 \ldots N$, are sampled at 60 Hz and then corrupted component-wise by a uniformly distributed random noise of magnitude 2 pixels before being processed for evaluating the image moments. The camera motion (and the optimization (19)) is instead updated at 100 Hz.



((a)) ((b))

Fig. 2: Results obtained by employing the classical moment set $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ and optimizing for the camera linear velocity $\boldsymbol{v}$. In this case $\rho(t)$ keeps very close to zero (the scale of Fig. (a) is 10^{-9}) and as a consequence the estimation error $\boldsymbol{z}(t)$ does not converge (Fig. (b))

As for the SfM algorithm providing an estimation $\hat{\boldsymbol{\chi}}(t) = -\hat{\boldsymbol{n}}(t)/\hat{d}(t)$ of the unknown 3D structure $\boldsymbol{\chi}(t) = -\boldsymbol{n}(t)/d(t)$, we made use of the general scheme recently discussed in [11] in the context of *active* SfM and already exploited in, e.g., [9], [10], [12] in a number of different applications. The reader is referred to these works for full details on the inner machinery of the algorithm: for our goals it is just worth mentioning that, for a given choice of the estimation gains, the convergence rate of the structure estimation error defined as

$$\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$$

is directly determined by the norm of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$. Thus, the larger the value of $\rho(t)$ from (15) during the camera motion, the faster the expected convergence of $\boldsymbol{z}(t) \rightarrow \boldsymbol{0}$.

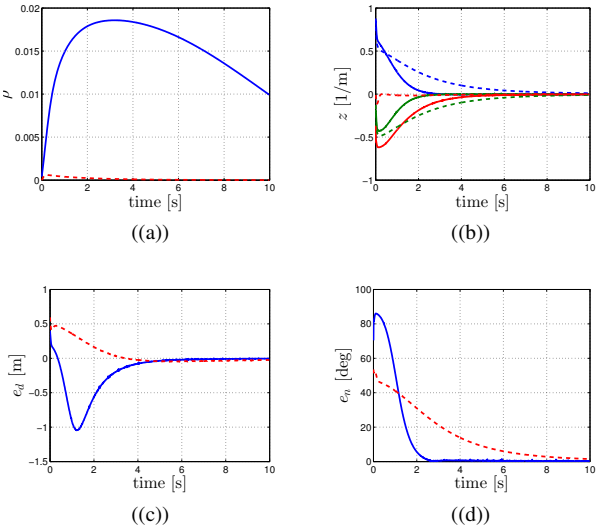
A. Unconstrained polynomial basis

We start with the results obtained by making use of the unconstrained polynomial basis (8) of fixed degree δ introduced in Sect. III-A.1. In particular, we tested our method by considering a set of $m = 3$ weighted moments $\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1),\, m_w(\boldsymbol{\theta}_2),\, m_w(\boldsymbol{\theta}_3)) \in \mathbb{R}^3$ with degree $\delta = 2$ defined as in (9), with then $\boldsymbol{\theta}_i \in \mathbb{R}^5$, $i = 1 \ldots 3$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2,\, \boldsymbol{\theta}_3) \in \mathbb{R}^p$, $p = 15$. This choice was meant to provide a direct comparison against the use of:

1) the more 'classical set' $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ that, as explained, is known to be an optimal choice for *controlling* the camera translational motion but also to yield poor results when employed for SfM purposes;
2) the set of *five* moments $(x_g,\, y_g,\, \mu_{20},\, \mu_{11},\, \mu_{02})$ which, as reported in [10], does allow for a converging estimation but at cost of an increased complexity (need of propagating five image moments).

The goal of the comparison is to prove that estimation of vector $\boldsymbol{\chi}$ is, instead, fully possible when a suitable combination of just *three* moments of order up to 2 is selected.

Figures 2(a–b) start showing the results obtained by employing the set $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ for estimating vector $\boldsymbol{\chi}$ while, at the same time, optimizing the camera linear velocity $\boldsymbol{v}$ by implementing the first row of (19) with $k_v = 1$. The linear velocity was initially set to $\boldsymbol{v}(t_0) = [0\ 0.1\ 0]^T$ [m/s] with then $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}(t_0)\| = 0.1$ [m/s] during the camera motion. As expected, and even despite the velocity optimization, the value of $\rho(t)$ keeps (numerically) very close to 0 with a maximum of $\approx 1.2 \cdot 10^{-9}$ (Fig. 2(a)). Thus, the chosen set $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ is not able to provide enough information for allowing convergence of the SfM scheme, and indeed the estimation error $\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$ even starts diverging (Fig. 2(b)).



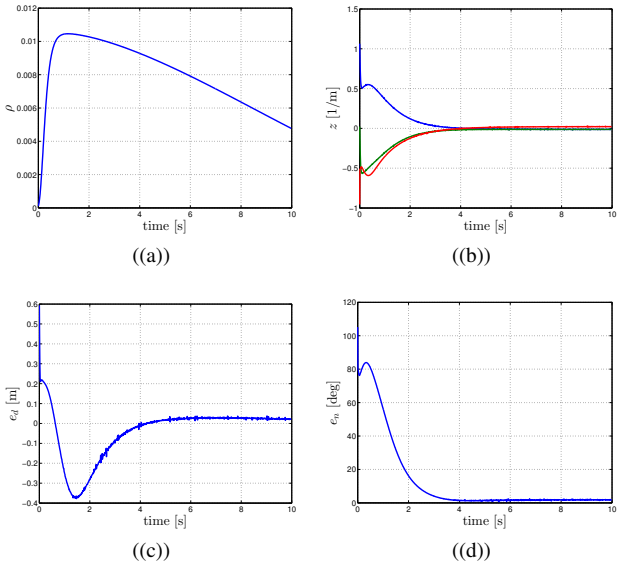
((a)) ((b))

((c)) ((d))

Fig. 3: *Solid* lines: results obtained by employing three weighted moments of degree $\delta = 2$ defined as in (9), and optimizing for both vector $\boldsymbol{\theta}$ and the camera linear velocity $\boldsymbol{v}$. Fig. (a): the value of ρ is now $\approx 10^7$ times larger than in the previous case of Fig. 2(a). This then allows for a quick convergence of the error quantities $\boldsymbol{z}(t)$, $e_d(t)$ and $e_n(t)$ (Figs. (b–d)). For a comparison, in all plots *dashed* lines correspond to the use of the *five* moments $(x_g,\, y_g,\, \mu_{20},\, \mu_{11},\, \mu_{02})$: it is worth noting how, despite the increased measurement set (five moments vs. only three), the estimation convergence results still slower than in the weighted moment case

On the other hand, exploiting the three weighted moments of degree 2 yields a much more satisfactory estimation performance: Figs. 3(a–d) report in *solid lines* the results obtained by implementing (19) with $k_v = 1$ and $k_\theta = 3$, and by taking again $\boldsymbol{v}(t_0) = [0\ 0.1\ 0]^T$ [m] as in the previous case. The parameter vector $\boldsymbol{\theta}$ was instead chosen at random under the constraint $\|\boldsymbol{\theta}_i(t_0)\| = 1$, $i = 1 \ldots 3$.

Looking at Fig. 3(a) one can then verify how now $\rho(t)$ attains an overall quite larger value compared to Fig. 2(a), with a maximum of $\approx 1.8 \cdot 10^{-2}$ (thus, more than 10^7 times larger than in the previous case). As a result, the estimation error $\boldsymbol{z}(t)$ is able to quickly converge towards $\boldsymbol{0}$ in about 4 seconds (Fig. 3(b)). For a better appreciation of the estimation performance, Figs. 3(c–d) also report the behavior of $e_d(t) = d(t) - \hat{d}(t)$ (the error in estimating the plane distance d) and $e_n = \arccos(\boldsymbol{n}^T(t)\hat{\boldsymbol{n}}(t))$ (the angular error in estimating the direction of the plane normal $\boldsymbol{n}$) with $\hat{d} = 1/\|\hat{\boldsymbol{\chi}}\|$ and $\hat{\boldsymbol{n}} = -\hat{\boldsymbol{\chi}}/\|\hat{\boldsymbol{\chi}}\|$. Finally, Figs. 3(a–d) superimpose in *dashed lines* the behavior of $\rho(t)$ and of the estimation errors when instead relying on the set of $m = 5$ moments $(x_g,\, y_g,\, \mu_{20},\, \mu_{11},\, \mu_{02})$ for estimating $\boldsymbol{\chi}$:



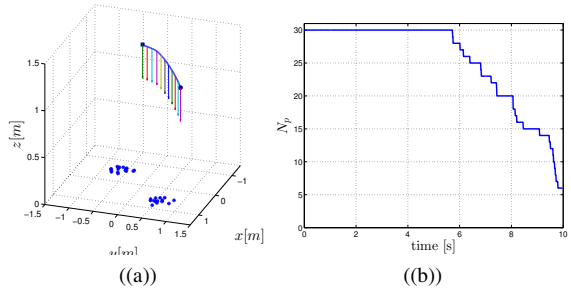
((a)) ((b)) ((c)) ((d))

Fig. 4: Results obtained by employing three weighted moments of degree $\delta = 2$ defined as in (9) and *only* optimizing for vector $\boldsymbol{\theta}$. Fig.(a): note how $\rho(t)$ still reaches a range of values comparable with the previous case of Fig. 3(a) despite the lack of any optimization of the camera velocity $\boldsymbol{v}$. Figs. (b–d): behavior of the error quantities $\boldsymbol{z}(t)$, $e_d(t)$ and $e_n(t)$

in this case, the estimation error does actually converge (as expected), but nevertheless at a slower rate compared to the weighted moment case (indeed, the maximum value of $\rho(t)$ is now 'only' $\approx 5.9 \cdot 10^{-4}$). We then believe these results clearly show the advantages of the proposed approach: the SfM scheme has its best performance when relying on the optimization (19) for automatically selecting (online) the best combination of *three* moments of order up to 2.

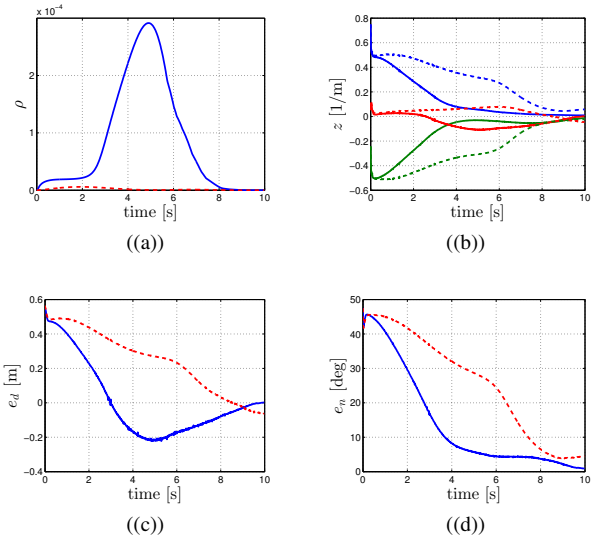
As an additional evaluation, Figs. 4(a–d) show the results obtained when *only* optimizing the parameter vector $\boldsymbol{\theta}$ while keeping a *constant* linear velocity $\boldsymbol{v}(t) = \boldsymbol{v}(t_0)$ during the whole motion (thus, by setting $k_v = 0$ in (19)). This case is meant to assess the optimization performance in a situation in which the camera velocity cannot be arbitrarily adjusted but it must be considered as 'given' by an external source. Thus, the only possibility for improving the conditioning of the observability matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ is to act on vector $\boldsymbol{\theta}$, i.e., on the moment shape. Nevertheless also in this situation $\rho(t)$ still reaches a range of values comparable with the previous case, with indeed $\max \rho(t) \approx 1.05 \cdot 10^{-2}$ against the previous $1.6 \cdot 10^{-2}$ (thus, still $\approx 10^7$ times larger than when employing the classical set $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$). As a result, vector $\boldsymbol{z}(t)$ keeps converging to $\mathbf{0}$ in about 4 [s] (Fig. 4(b)) even if slightly more slowly w.r.t. the previous case of Fig. 3(b) (as one could expect because of the smaller value of $\rho(t)$). In any case, we believe it is worth noting how the *sole* optimization of the moment shape (via vector $\boldsymbol{\theta}$) is still able to yield a very satisfactory SfM performance even for a non-optimal camera motion.

We finally remark that in all simulations the camera angular velocity $\boldsymbol{\omega}$ was exploited for keeping the centroid of the observed point features $\boldsymbol{p}_i$ at the center of the image



((a)) ((b))

Fig. 5: Results obtained by employing the constrained weighted moments (10–11), a camera with limited fov, and optimizing for both vector $\boldsymbol{\theta}$ and the linear velocity $\boldsymbol{v}$. Fig. (a): camera trajectory and direction of the optical axis during the estimation task. Fig. (b): number N_p of tracked point features over time



((a)) ((b)) ((c)) ((d))

Fig. 6: *Solid* lines: results obtained by employing the constrained weighted moments (10–11), a camera with limited fov, and optimizing for both vector $\boldsymbol{\theta}$ and the linear velocity $\boldsymbol{v}$. Fig. (a): behavior of $\rho(t)$ which reaches a maximum of $\approx 2.9 \cdot 10^{-4}$ before starting to decrease because of the fewer tracked points. Figs. (b–d): behavior of the error quantities $\boldsymbol{z}(t)$, $e_d(t)$ and $e_n(t)$. Note how all quantities keep behaving smoothly despite the frequent loss of tracked points. In *dashed* lines, the behavior that all quantities would have had in case no optimization of $\boldsymbol{\theta}$ had been performed

plane (we recall that matrix $\boldsymbol{\Omega}$ and, thus, $\rho(t)$ do not depend on $\boldsymbol{\omega}$ that can then be freely chosen without affecting the estimation performance).

B. Constrained polynomial basis

We now address the case of the constrained polynomial basis (10–11) described in Sect. III-A.2 and meant to smoothly take into account the loss/gain of point features because of the camera limited fov. We consider again a set of $m = 3$ weighted moments $\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1),\, m_w(\boldsymbol{\theta}_2),\, m_w(\boldsymbol{\theta}_3)) \in \mathbb{R}^3$ with both functions $w^x(\cdot)$ and $w^y(\cdot)$ taken as the fifth-order polynomials given in (12) with, therefore, a total of $p^x + p^y - 8 = 4$ parameters to be optimized. The initial camera velocity $\boldsymbol{v}(t_0)$ was set as in the previous cases, and the optimization action (19) was again implemented with $k_v = 1$ and $k_\theta = 3$. The camera angular velocity $\boldsymbol{\omega}$ was

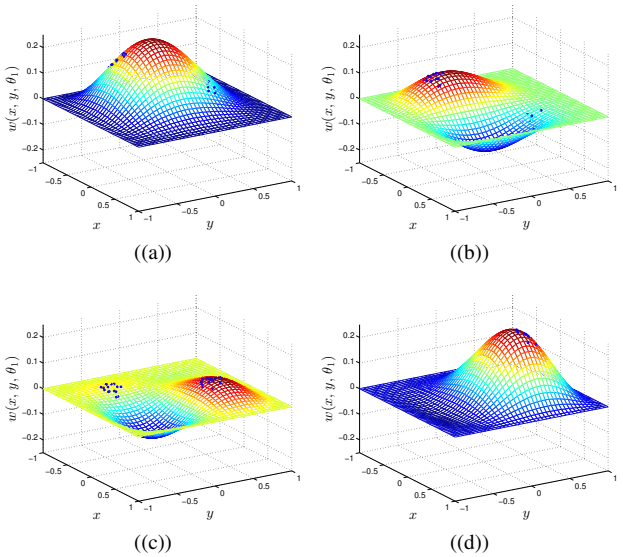


Fig. 7: Four snapshots of the weighting function $w(x,\, y,\, \boldsymbol{\theta}_1)$ taken during the camera motion. It is interesting to visualize how function $w(\cdot)$ automatically adjusts its shape as a function of the observed scene (e.g., it tends to peak around clusters of points)

instead kept null for facilitating the loss or point features during motion.

Figure 5(a) shows the camera trajectory during the estimation task, and Fig. 5(b) reports the number $N_p(t)$ of tracked features over time: after about 5 [s] some points start being lost, dropping from a total of 30 to a minimum of 6 at $t = 10$ [s]. Nevertheless, thanks to the adopted *constrained* weighted moments, the scene structure is still correctly estimated without suffering from discontinuities or numerical instabilities because of the lost features. Figure 6(a) reports again the behavior of $\rho(t)$ (blue solid line) that reaches a maximum of $\approx 2.9 \cdot 10^{-4}$ before starting to decrease at $t \approx 5$ [s] because of the fewer tracked points. As a comparison, Fig. 6(a) also reports the superimposed behavior of $\rho(t)$ in case no optimization of vector $\boldsymbol{\theta}$ had been performed (the almost horizontal red dashed line). In this case, the maximum attained value for $\rho(t)$ would have been $\approx 1.2 \cdot 10^{-6}$ (100 times smaller), thus proving again the importance of properly optimizing the shape of the chosen weighting function $w(\cdot)$. Figures 6(b–d) then show (in *solid* lines) the behavior of the estimation error $\boldsymbol{z}(t)$ and of the corresponding quantities $e_d(t)$ and $e_n(t)$ that *smoothly* reach convergence in about 10 [s] of motion despite the loss of point features. Again, for a comparison, Figs. 6(b–d) also report the superimposed behavior (in *dashed* lines) of the estimation errors in case of no optimization of vector $\boldsymbol{\theta}$ (all quantities have a slower convergence rate as expected).

Finally, Fig. 7 depicts four snapshots of the shape of function $w(x,\, y,\, \boldsymbol{\theta}_1)$ used to compute the first constrained weighted moment. The reader can then appreciate how the function shape evolves over time and, in particular, *automatically* polarizes its peaks around the location of the tracked point features. A video showing an animation of the reported simulations is also attached to the paper.

VI. CONCLUSIONS

In this paper we addressed the problem of *automatically* selecting an optimal set of image moments for improving the performance in estimating the 3D structure of a planar scene. This is achieved by replacing the classical definition of *unweighted moments* with the novel concept of *parametric weighted moments* that can span a whole class of image moments as a function of a parameter vector. By then optimizing *online* this parameter vector, the moment shape can be automatically adjusted as a function of the perceived scene. A number of simulation results, involving the estimation of the 3D parameters of a planar scene from measured moments (and known camera motion), fully confirms the effectiveness of the proposed approach. We are currently working towards an experimental validation of these results, as well as the application of these ideas in the context of image-based visual servoing.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] 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.

[3] F. Hoffmann, T. Nierobisch, T. Seyffarth, and G. Rudolph, "Visual servoing with moments of SIFT features," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2006, pp. 4262–4267.

[4] E. Bugarin and R. Kelly, "Direct visual servoing of planar manipulators using moments of planar targets," in *Robot Vision*, A. Ude, Ed. INTECH, 2010.

[5] C. Copot, C. Lazar, and A. Burlacu, "Predictive control of nonlinear visual servoing systems using image moments," *IET control theory & applications*, vol. 6, no. 10, pp. 1486–1496, 2012.

[6] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan, "Kernel-based visual servoing," in *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 1975–1980.

[7] M. Bakthavatchalam, F. Chaumette, and E. Marchand, "Photometric moments: New promising candidates for visual servoing," in *2013 IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 5521–5526.

[8] 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.

[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] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Experiments of plane estimation by active vision from point features and image moments," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015.

[11] 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.

[12] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere and Cylinder," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.

[13] M. I. Friswell, "The derivatives of repeated eigenvalues and their associated eigenvectors," *Journal of Vibration and Acoustics*, vol. 118, pp. 390–397, 1996.

[14] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed. Princeton University Press, 2009.

[15] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Coupling Visual Servoing with Active Structure from Motion," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 3090–3095.