

Active Structure-from-Motion for 3D Straight Lines*

André Mateus, Omar Tahri, and Pedro Miraldo

Abstract—A reliable estimation of 3D parameters is a must for several applications like planning and control, in which is included Image-Based Visual Servoing. This control scheme depends directly on 3D parameters, e.g. depth of points, and/or depth and direction of 3D straight lines. Recently, a framework for *Active Structure-from-Motion* was proposed, addressing the former feature type. However, straight lines were not addressed. These are 1D objects, which allow for more robust detection, and tracking. In this work, the problem of *Active Structure-from-Motion* for 3D straight lines is addressed. An explicit representation of these features is presented, and a change of variables is proposed. The latter allows the dynamics of the line to respect the conditions for observability of the framework. A control law is used with the purpose of keeping the control effort reasonable, while achieving a desired convergence rate. The approach is validated first in simulation for a single line, and second using a real robot setup. The latter set of experiments are conducted first for a single line, and then for three lines.

I. INTRODUCTION

There are many robotic applications for which the recovery of 3D information is required. Among those is Image-Based Visual Servoing, which consists in using visual information (e.g. image points) to control a robot [1]. This kind of control is highly dependent on a good estimation of 3D parameters, for instance the depth of points, and/or the depth and direction vector of lines [2].

A way to estimate 3D information is Structure-from-Motion, which consists in estimating 3D parameters from images taken by a moving camera [3], [4]. It is a well studied problem in Computer Vision, with emphasis on points and lines. Even though those works present solutions to the problem, they suffer from considering either big displacement between views or more than two views. Thus, preventing their use in Visual Servoing, since it requires continuous estimation. In order to cope with these issues, some authors adopt filtering strategies, by taking measurements from consecutive camera images and the displacement of the camera between each frame (assumed to be known) [5], [6], [7], [8].

A non-linear state estimation scheme for unknown parameters is presented in [9], and is applied to the estimation of image points' depth, and focal length of a camera. The observer is proved to be stable, as long as, the *persistence of*

excitation condition is satisfied. More approaches, based on non-linear state estimators, are presented in [10], [11], [12], [13], [14], [15]. A comparison of a filtering solution with a non-linear estimation framework is presented in [16].

Some authors addressed the control of a camera to optimize 3D estimation (*Active Vision* [17]). In [18], a method to recover the 3D structure of points, cylinders, straight lines, and spheres is presented. It consists in using the Implicit Function Theorem, to obtain an explicit expression of the 3D information as a function of image measurements, their time derivative, and the camera's velocity. The active component (control law) attempts to simultaneously keep the features static in the image, and perform a desired trajectory (depending on the feature). This approach requires the features time derivative (on the image plane) to be known, which are not directly available, and thus require discrete approximation. Besides, the control law used does not give much insight on how it may affect the estimation error.

The optimization of the camera's motion, in order to have a desired estimation error response, is addressed in [19], where a framework for *Active Structure-from-Motion* is presented. This is a general framework, that can be used for a wide variety of visual features, and its convergence behavior is well defined. The framework has been applied to several feature types, like points, cylinders, spheres [20], planes from measured image moments [21], [22], and rotational invariants [23], [24]. A method for coupling this framework with a Visual Servoing control scheme is presented in [25].

Active Vision has been addressed for a large set of visual features, yet straight lines have not been explored. The main focus has been the estimation of a point's depth. However, in practice, it is useful to use richer features. In particular, straight lines which are 1D objects, and whose one-to-one association from the image to the world is easier to obtain. Besides, lines are also easier to detect in images (e.g. Hough Transform [26]), and allow for a more robust tracking [27]. To the best of our knowledge, active vision for lines has been previously addressed by two works. The first is [18], whose shortcomings have already been stated. The second is [20], where lines are only considered indirectly, in the estimation of a cylindrical target radius. In fact, the estimation scheme that allows a full line recovery is not explored, since the authors resort to a formulation specific to cylinders.

This work studies the problem of *Active Vision* for 3D straight lines. Previous works represented these features as the intersection of two planes [18]. That representation is implicit, and does not have a direct counterpart in the image plane. Thus, it is usually used together with the (ρ, θ) representation in the image plane, which even though is

*This work was partially supported by the Portuguese FCT grants PD/BD/135015/2017 (through the NETSys Doctoral Program) & SFRH/BPD/111495/2015, and ISR/LARSyS Strategic Funding by the FCT project PEST-OE/EEI/LA0009/2013.

A. Mateus & P. Miraldo are with Institute for Systems and Robotics (ISR/IST), LARSyS, Instituto Superior Técnico, Univ. Lisboa, Portugal {andre.mateus, pedro.miraldo}@tecnico.ulisboa.pt.

O. Tahri is with INSA Centre Val de Loire, Université d'Orléans, PRISME EA 4229, Bourges, France omar.tahri@insa-cvl.fr.

minimal, is also periodical [2]. Furthermore, the coupling of this two representations leads to complex dynamics. In this work lines are represented by *binormalized Plücker coordinates* [2], which are presented in Sec. II-A, along with their dynamics. This coordinates are explicit (a set of coordinates defines a single line); define lines everywhere in space (except if they contain the camera optical center); and have a direct representation in the image plane (the moment vector is the normal of the line's interpretation plane)¹.

State estimation is addressed by the framework in [19], which is presented in Sec. II-B. Since, the dynamics of the *binormalized Plücker coordinates* do not respect the requirements for convergence, a change of variables is presented in Sec. III. Followed by the dynamics in the new coordinates, and the non-linear observer for 3D straight lines structure.

The proposed observer is validated in simulation for a single line, whose setup and results are presented in Sec. IV-A. The control law used for active structure estimation is also presented. This was designed to keep the control effort relatively low (in terms of the norm of the velocity vector), while achieving a desired convergence rate. Those results are replicated with a real robotic platform [28], in Sec. IV-B. In Sec. IV-B.2, an application of the observer for recovering the 3D structure of three lines is presented. Finally, the conclusions are presented in Sec. V.

II. BACKGROUND

This section presents the theoretical background and concepts required for this work. First, the representation of 3D straight lines and their dynamics are presented, followed by the framework for *Active Structure-from-Motion*.

A. Dynamics of the 3D Straight Lines

Plücker coordinates are a common way to represent 3D lines [29], given that they can represent lines everywhere in the space (except if they intersect the camera's optical center). Besides, they have a direct representation in the image plane, since they contain the normal vector to the interpretation plane. This coordinates are defined as

$$\mathcal{L} \sim \begin{bmatrix} \mathbf{u} \\ \mathbf{m} \end{bmatrix}, \quad \text{with } \mathcal{L} \subset \mathcal{P}^5 \quad (1)$$

where \mathcal{P}^5 is the five-dimensional projective space, $\mathbf{u} \in \mathbb{R}^3$, and $\mathbf{m} \in \mathbb{R}^3$ represent the line's direction and moment respectively, up to a scale factor, and satisfy

$$\mathbf{u}^T \mathbf{m} = 0. \quad (2)$$

By normalizing the direction vector, we obtain the *Euclidean Plücker coordinates* [2]

$$\begin{cases} \mathcal{L} = \begin{bmatrix} \mathbf{d} \\ \mathbf{n} \end{bmatrix} \\ \mathbf{d}^T \mathbf{n} = 0 \\ \mathbf{d}^T \mathbf{d} = 1, \end{cases} \quad (3)$$

where $\mathbf{d} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$, and $\mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{u}\|}$. This representation is explicit, since a set of coordinates represents a single line.

Keep in mind that \mathbf{n} represents the normal vector to the interpretation plane, thus it is possible to represent the projection of the line in the image \mathbf{h} [30] as

$$\mathbf{h} = \frac{\mathbf{n}}{\|\mathbf{n}\|}. \quad (4)$$

Let $l = \|\mathbf{n}\|$ be the depth of the line. If we recall that for every point $\mathbf{p} \in \mathbb{R}^3$ in the line it holds

$$\mathbf{n} = \mathbf{p} \times \mathbf{d}, \quad (5)$$

and since $\|\mathbf{d}\| = 1$. Then $\|\mathbf{n}\| = \|\mathbf{p}\| \sin(\phi)$, with ϕ being the angle between the point and the direction. If the depth of the line is considered to be the distance of the point closest to the origin, then the norm of \mathbf{n} will always be equal to the depth. Finally, the *binormalised Plücker coordinates* [2] can be defined as

$$\mathcal{L} = \begin{bmatrix} \mathbf{d} \\ l\mathbf{h} \end{bmatrix}. \quad (6)$$

Let $\mathbf{v}_c = [\nu_c, \omega_c]^T \in \mathbb{R}^6$ be the camera velocity, with $\nu_c \in \mathbb{R}^3$ and $\omega_c \in \mathbb{R}^3$ being the linear and angular velocities respectively. Then the dynamics of (6) are

$$\dot{\mathbf{d}} = \omega_c \times \mathbf{d} \quad (7)$$

$$\dot{\mathbf{h}} = \omega_c \times \mathbf{h} - \frac{\nu_c^T \mathbf{h}}{l} (\mathbf{d} \times \mathbf{h}) \quad (8)$$

$$\dot{l} = \nu_c^T (\mathbf{d} \times \mathbf{h}). \quad (9)$$

Let us consider the coordinates of a straight line to be our state vector. Thus, (7), (8), and (9) give us the state equations. From those the state estimation scheme can be designed, whose background is presented in the next section.

B. Active Structure-From-Motion

This section presents the framework for *Active Structure-from-Motion* [19]. Let $\mathbf{x} = [\mathbf{s}, \boldsymbol{\chi}]^T \in \mathbb{R}^{m+p}$ be the state vector, whose dynamics are given by

$$\begin{cases} \dot{\mathbf{s}} = \mathbf{f}_m(\mathbf{s}, \omega_c) + \boldsymbol{\Omega}^T(\mathbf{s}, \nu_c) \boldsymbol{\chi} \\ \dot{\boldsymbol{\chi}} = \mathbf{f}_u(\mathbf{s}, \boldsymbol{\chi}, \nu_c), \end{cases} \quad (10)$$

where $\mathbf{s} \in \mathbb{R}^m$ is the vector of the measurable components of the state, and $\boldsymbol{\chi} \in \mathbb{R}^p$ the unknown components. Besides, let $\hat{\mathbf{x}} = [\hat{\mathbf{s}}, \hat{\boldsymbol{\chi}}]^T$ be the estimated state, and $\mathbf{e} = [\tilde{\mathbf{s}}, \tilde{\boldsymbol{\chi}}]^T$ be the state estimation error, with $\tilde{\mathbf{s}} = \mathbf{s} - \hat{\mathbf{s}}$ and $\tilde{\boldsymbol{\chi}} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$. Then, to recover the unknown quantities, the following estimation scheme can be used

$$\begin{cases} \dot{\hat{\mathbf{s}}} = \mathbf{f}_m(\mathbf{s}, \omega_c) + \boldsymbol{\Omega}^T(\mathbf{s}, \nu_c) \hat{\boldsymbol{\chi}} + \mathbf{H} \tilde{\mathbf{s}} \\ \dot{\hat{\boldsymbol{\chi}}} = \mathbf{f}_u(\mathbf{s}, \hat{\boldsymbol{\chi}}, \nu_c) + \alpha \boldsymbol{\Omega}(\mathbf{s}, \nu_c) \tilde{\mathbf{s}}, \end{cases} \quad (11)$$

with \mathbf{H} positive definite, and $\alpha > 0$.

According to the *persistence of excitation condition* [9], convergence of the estimation error to zero is possible, iff the matrix $\boldsymbol{\Omega} \boldsymbol{\Omega}^T$ is full rank. Then, it is possible to prove that, the convergence rate will depend on the norm of that matrix, and in particular, on its smallest eigenvalue σ_1^2 . This depends on the measurable components of the state and

¹Plane containing the line and the optical center of the camera.

on the camera's linear velocity, which can be optimized to control the convergence rate.

Let $\mathbf{U}\Sigma\mathbf{V} = \mathbf{\Omega}$ be the singular value decomposition of matrix $\mathbf{\Omega}$, where $\Sigma = [\mathbf{S} \quad \mathbf{0}]$, with $\mathbf{S} = \text{diag}(\{\sigma_i\})$, $i = 1, \dots, k$. Then, $\mathbf{H} \in \mathbb{R}^{m \times m}$ may be chosen as

$$\mathbf{H} = \mathbf{V} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \mathbf{V}^T, \quad (12)$$

where $\mathbf{D}_1 \in \mathbb{R}^{p \times p}$ is a function of the singular values of $\mathbf{\Omega}$, and $\mathbf{D}_2 \in \mathbb{R}^{(m-p) \times (m-p)}$ is set to be the identity matrix. Following [19], the former is defined as $\mathbf{D}_1 = \text{diag}(\{c_i\})$, $c_i > 0$ with $c_i = 2\sqrt{\alpha}\sigma_i$, and $i = 1, \dots, k$. This choice prevents oscillatory modes, thus trying to achieve a critically damped transient behavior.

In order to get a desired rate, one can either tune the gain α and/or act on the input to lead the eigenvalue to a desired value. In this work, we focus on the latter. Let us compute the total time derivative of the eigenvalues of matrix $\mathbf{\Omega}\mathbf{\Omega}^T$. Using the results in [19], we conclude that

$$(\dot{\sigma}_i^2) = \mathbf{J}_{\nu_c, i} \dot{\nu}_c + \mathbf{J}_{s, i} \dot{\mathbf{s}} \quad \text{with } i = 1, \dots, k, \quad (13)$$

where matrices $\mathbf{J}_{\nu_c, i}$, $\mathbf{J}_{s, i}$ are the Jacobian matrices yielding the relationship between the eigenvalues of $\mathbf{\Omega}\mathbf{\Omega}^T$, and the linear velocity and the measurable components of the state respectively. In particular

$$\mathbf{J}_{\nu_c, i} = \left[\mathbf{v}_i^T \frac{\partial \mathbf{\Omega}\mathbf{\Omega}^T}{\partial \nu_x} \mathbf{v}_i, \mathbf{v}_i^T \frac{\partial \mathbf{\Omega}\mathbf{\Omega}^T}{\partial \nu_y} \mathbf{v}_i, \mathbf{v}_i^T \frac{\partial \mathbf{\Omega}\mathbf{\Omega}^T}{\partial \nu_z} \mathbf{v}_i \right], \quad (14)$$

with \mathbf{v}_i being the normalized eigenvector associated with the i^{th} eigenvalue of $\mathbf{\Omega}\mathbf{\Omega}^T$. A differential inversion technique can be used to regulate the eigenvalues, by acting on vector $\dot{\nu}_c$. Notice that, when applying an inversion technique, it may not be possible to compensate for the second term on the right-hand side of (13). A way to compensate, for the effect of $\dot{\mathbf{s}}$ in the dynamics, is to enforce $\dot{\mathbf{s}} \simeq \mathbf{0}$. In this work, that effect is compensated making use of the angular velocity as described in Sec. IV-A.

III. ACTIVE STRUCTURE-FROM-MOTION USING 3D STRAIGHT LINES

Sec. II-B presents the framework for *Active Structure-from-Motion*. This consists of a full state observer, whose state is partially measured. In this work, the measurement consists of vector \mathbf{h} , which is the normal vector to the interpretation plane of the line. Besides \mathbf{h} , our representation consists of the direction vector and the line's depth. These will only be retrieved from the estimated state. A requirement is that the unknown variables appear linearly on the dynamics of the measurable ones. From (8), we can conclude that this does not happen. In order to cope with this issue, the change of variables

$$\chi = \frac{\mathbf{d}}{l}, \quad (15)$$

can be considered. Using (9) and (7), it is easy to conclude that

$$\dot{\chi} = \omega_c \times \chi - \chi \nu_c^T (\chi \times \mathbf{h}). \quad (16)$$

Then, from (11) and (8), one can define

$$\mathbf{\Omega} = -\nu^T \mathbf{h} [\mathbf{h}]_{\times}, \quad (17)$$

where $[\mathbf{h}]_{\times}$ is a 3×3 skew-symmetric matrix that linearizes the cross product ($\chi \times \mathbf{h} = -[\mathbf{h}]_{\times} \chi$).

Convergence of the estimation error \mathbf{e} to 0 is possible iff the square matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ is full rank. From (17), this is not the case, since its rank is equal to 2 (this happens because of the multiplication by the skew-symmetric matrix, which has, by definition, rank 2). In order to deal with this issue, the orthogonality of the *Plücker* coordinates is explored. The obtained system belongs then to the large family of Differential Algebraic Systems [31]. Since, the algebraic condition is linear, a solution consists in expressing the coordinate of χ , corresponding to the coordinate of \mathbf{h} with highest absolute value, as a function of the others. This will not require a switching strategy², since the condition $\dot{\mathbf{h}} \simeq \mathbf{0}$ is enforced. Thus, the coordinate with the highest absolute value is always the same throughout the task. Notice that, by enforcing $\dot{\mathbf{h}} \simeq \mathbf{0}$, the angular velocity is used to maintain the interpretation plane of the line constant. This allows us to compensate for the effects of $\dot{\mathbf{h}}$ in the dynamics of σ_i^2 , and thus the linear velocity of the camera can be used to obtain a desired convergence rate.

Let us consider the case where the z coordinate of χ is fixed. Solving the orthogonality constraint for χ_z , we obtain

$$\chi_z = -\frac{\chi_x h_x + \chi_y h_y}{h_z}. \quad (18)$$

By replacing (18) in (8) and (16), and changing the state to $\mathbf{x} = [\mathbf{h}, \chi_x, \chi_y]^T$, the dynamics become

$$\dot{\mathbf{h}} = \omega_c \times \mathbf{h} + \nu_c^T \mathbf{h} \begin{bmatrix} -\frac{h_x h_y}{h_z} & -h_z - \frac{h_y^2}{h_z} \\ h_z + \frac{h_x^2}{h_z} & \frac{h_x h_y}{h_z} \\ -h_y & h_x \end{bmatrix} \begin{bmatrix} \chi_x \\ \chi_y \end{bmatrix} \quad (19)$$

$$\begin{aligned} \dot{\chi}_x = & -\omega_{c_z} \chi_y - \omega_{c_y} \frac{h_x \chi_x + h_y \chi_y}{h_z} - \dots \\ & - \nu_{c_x} \left(h_z \chi_x \chi_y + \frac{h_y \chi_x (h_x \chi_x + h_y \chi_y)}{h_z} \right) + \dots \\ & + \nu_{c_y} \left(h_z \chi_x^2 + \frac{h_x \chi_x (h_x \chi_x + h_y \chi_y)}{h_z} \right) - \dots \\ & - \nu_{c_z} (h_y \chi_x^2 - h_x \chi_y \chi_x) \end{aligned} \quad (20)$$

$$\begin{aligned} \dot{\chi}_y = & \omega_{c_z} \chi_x + \omega_{c_x} \frac{\chi_x h_x + \chi_y h_y}{h_z} - \dots \\ & - \nu_{c_x} \left(h_z \chi_y^2 + \frac{h_y \chi_y (h_x \chi_x + h_y \chi_y)}{h_z} \right) + \dots \\ & + \nu_{c_y} \left(h_z \chi_x \chi_y + \frac{(h_x \chi_y (h_x \chi_x + h_y \chi_y))}{h_z} \right) + \dots \\ & + \nu_{c_z} (h_y \chi_x^2 - h_y \chi_y \chi_x). \end{aligned} \quad (21)$$

²A switching strategy (in this context) is a change of the state variables in runtime, i.e., if the moment vector is not constant the coordinate with the highest absolute value is not always the same, and thus it would be necessary to fix a different coordinate of χ .

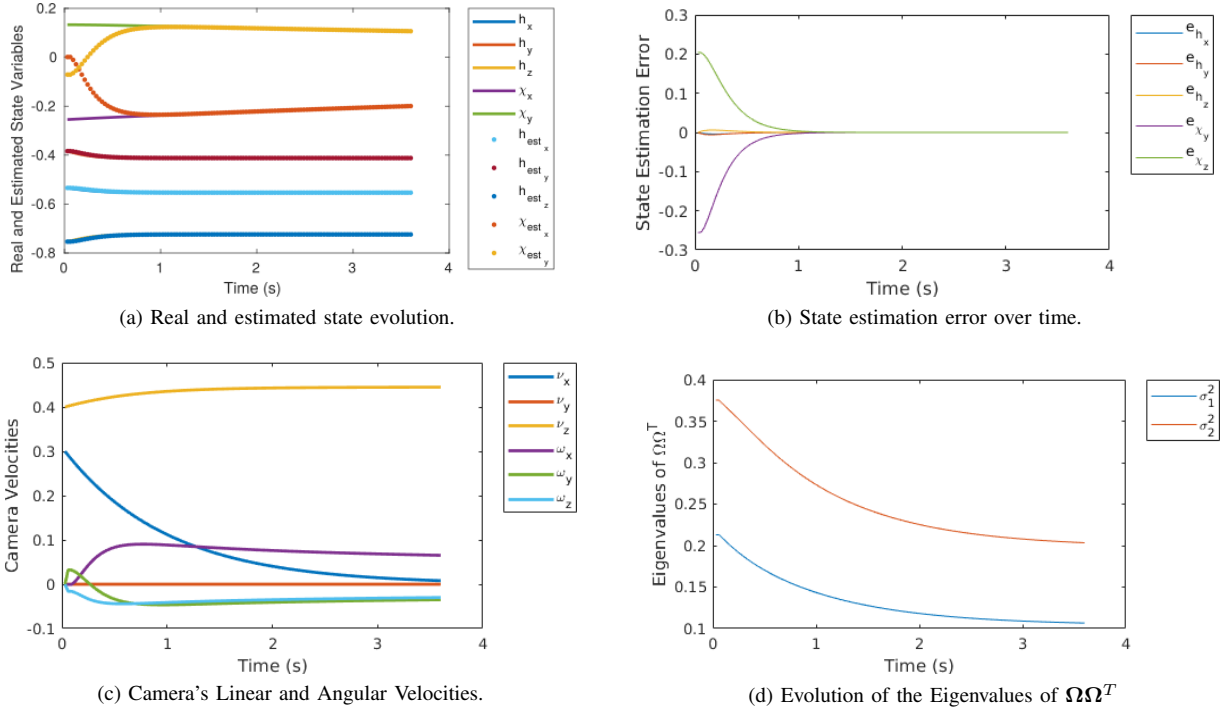


Fig. 1: Simulation results for a single line. The real and the estimated state are presented in the top left plot, and the corresponding state estimation error in the top right plot. The velocities of the camera throughout the task are presented in the bottom left plot. Finally the eigenvalues of the matrix $\Omega\Omega^T$ are presented in the bottom right plot.

For simplicity sake, let us define $\dot{\chi}_x = f_{u_x}(\mathbf{h}, \chi_x, \chi_y, \mathbf{v}_c)$, and $\dot{\chi}_y = f_{u_y}(\mathbf{h}, \chi_x, \chi_y, \mathbf{v}_c)$. These definitions will be used henceforward to refer the dynamics of the unknown variables. In this formulation

$$\Omega = \nu_c^T \mathbf{h} \begin{bmatrix} -\frac{h_x h_y}{h_z} & h_z + \frac{h_x^2}{h_z} & -h_y \\ -h_z - \frac{h_y^2}{h_z} & \frac{h_x h_y}{h_z} & h_x \end{bmatrix}, \quad (22)$$

and $\Omega\Omega^T$ is full rank, as long as the linear velocity and the moment vector are not orthogonal (this restriction is included in (17) also). The observer can thus be written as

$$\begin{cases} \dot{\mathbf{h}} = \omega_c \times \mathbf{h} + \Omega^T \begin{bmatrix} \hat{\chi}_x \\ \hat{\chi}_y \end{bmatrix} + \mathbf{H}\tilde{\mathbf{h}} \\ \begin{bmatrix} \dot{\hat{\chi}}_x \\ \dot{\hat{\chi}}_y \end{bmatrix} = \begin{bmatrix} f_{u_x}(\mathbf{h}, \hat{\chi}_x, \hat{\chi}_y, \mathbf{v}_c) \\ f_{u_y}(\mathbf{h}, \hat{\chi}_x, \hat{\chi}_y, \mathbf{v}_c) \end{bmatrix} + \alpha \Omega \tilde{\mathbf{h}}. \end{cases} \quad (23)$$

Notice that, the same approach can be replicated for both the x and y coordinates of \mathbf{h} , yielding matrices Ω compliant with the requirements for convergence. The observer is validated with simulation tests, which will be presented next.

IV. RESULTS

We start by validating our approach, resorting to simulation tests for a single line. Then, we apply the method in a real mobile robot [28], considering a single line, and finally three lines simultaneously.

A. Simulation Results

This section presents the simulation results for active observation of vector χ . The goal here is to keep the norm of the linear velocity relatively low, so the control effort does not increase excessively, while maintaining the possibility to lead the eigenvalues of $\Omega\Omega^T$ to a desired value (σ_{des}^2). Let $\mathbf{J}_{\nu_c} = [\mathbf{J}_{\nu_c,1}^T, \mathbf{J}_{\nu_c,2}^T]^T \in \mathbb{R}^{2 \times 3}$ be the Jacobian matrix, which relates $\dot{\nu}_c$ with the time derivative of the eigenvalues of $\Omega\Omega^T$, and $\sigma^2 = [\sigma_1^2, \sigma_2^2]^T \in \mathbb{R}^2$ be the current eigenvalues. Then, the control law used is

$$\dot{\nu}_c = k_1 \mathbf{J}_{\nu_c}^\dagger (\sigma_{des}^2 - \sigma^2) + k_2 (\mathbf{I}_2 - \mathbf{J}_{\nu_c}^\dagger \mathbf{J}_{\nu_c}) \nu_c, \quad (24)$$

which as been proposed in [19], where $\mathbf{I}_2 \in \mathbb{R}^{2 \times 2}$ is an identity matrix, $k_1 > 0$ and $k_2 > 0$ are positive constants, and $\mathbf{J}_{\nu_c}^\dagger$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix.

In order to compensate the effect of $\dot{\mathbf{h}}$ in the dynamics of the eigenvalues, the condition $\dot{\mathbf{h}} \simeq \mathbf{0}$ is enforced. This is achieved using the angular velocity, which is obtained by setting (8) to zero and solving for ω_c yielding

$$\omega_c = (\nu_c^T \mathbf{h}) \hat{\chi}. \quad (25)$$

The observer in (23) was simulated in MATLAB, considering a perspective camera, with the intrinsic parameters matrix given by $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$. All six degrees-of-freedom (dof) are assumed to be controllable. The true coordinates of the line are obtained by generating a random point in a cube with $3m$ side in front of the camera, then a unit direction is

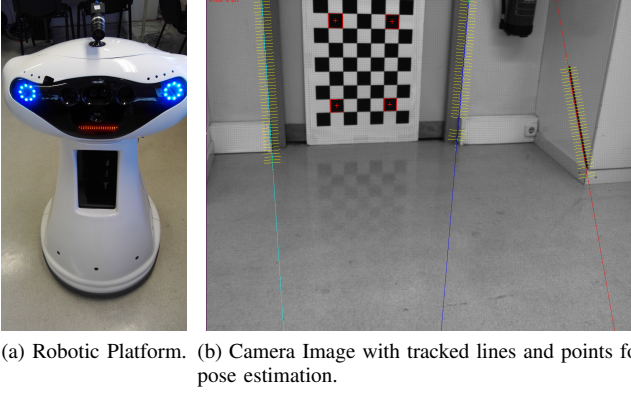


Fig. 2: On the left the robotic platform used in the experimental results is shown. It is an omnidirectional platform, and thus with 3 dof (2 linear, and 1 angular). On top there is a Pointgrey Flea3 USB3 [35] camera, which is the visual sensor used in the experiments. On the right an image from the camera with the four points, of the chessboard, used for pose estimation (to obtain the true coordinates of the lines) and the lines tracked with the moving-edges tracker [33].

randomly selected, and the moment vector is computed using (5). Finally, the change of variables (presented in Sec. III) is applied. The initial estimate of $(\hat{\chi})$ is also generated randomly, $\hat{\mathbf{h}}$ is initialized with its true value, since it is available from the measurements. Fig. 1 presents the simulation results for this case, with the following gains $k_1, k_2 = 1$, $\alpha = 2000$, and $\sigma_{des}^2 = [0.1, 0.2]^T$ (see (24) and (23)). Fig. 1(a) presents the state's evolution for both the real system and its estimate, where we can observe that the objective $\hat{\mathbf{h}} \simeq \mathbf{0}$ was achieved. Fig. 1(b) presents the state error over time, where we can see that convergence is achieved in about 1 second. Fig. 1(c) presents the velocities. Notice that the velocity norm is kept constant. Finally, Fig. 1(d) presents the eigenvalues over time. Notice that they reach their desired values. The total error of the *Plücker* coordinates is $\|\mathcal{L} - \mathcal{L}_{est}\| = 0.0019$, where \mathcal{L} and \mathcal{L}_{est} , are the real coordinates, and the estimated coordinates of the line as define in (6) respectively.

B. Real Experiments

Finally, the experimental results with a real robotic platform [28] (see Fig. 2(a)) are presented. It is an omnidirectional mobile platform having 3 dof (2 linear, and 1 angular). Our observer scheme was implemented resorting to Robot Operating System (ROS) [32] for sending controls to the platform, and receiving its odometry readings. Lines are tracked with the moving-edges tracker [33], available in *ViSP* [34]. The camera used was a Pointgrey Flea3 USB3 [35] mounted on top of the robot, as shown in Fig. 2(a).

1) *Single line estimation*: In a first step, we replicate the experimental setup in Sec. IV-A, in order to show that the performance is kept in a real setup. The goal is to estimate a line defined in a chessboard. The board is used only to obtain the initial coordinates of the line, by computing the initial pose of the camera w.r.t. the board. For this purpose,

we used four points (marked in the Fig. 2(b)), and then applied the *POSIT* algorithm [36] available in *ViSP*. The camera's velocity is retrieved from the robot's odometry, after applying the transformation from the robot's base to the camera reference frame (computed a priori). Fig. 3 presents the experimental results for this case, with the following gains $k_1, k_2 = 1$, $\alpha = 2000$, and $\sigma_{des}^2 = [0.1, 0.2]^T$. Fig. 3(a) presents the estimated state evolution, where we can observe that the objective $\hat{\mathbf{h}} \simeq \mathbf{0}$ is not entirely achieved. This is due to the reduced number of dof. Fig. 3(b) presents the state error over time. We can see that convergence is achieved in about 3 seconds. Notice that it takes some time for the robot to start moving, this is a limitation on the robot's drivers. Fig. 3(c) presents the velocities, where we note that the velocity readings are noisy, since they are provided by the robot's odometry. Finally Fig. 3(d) presents the eigenvalues, which given the noisy velocity readings did not reached the desired values. The total error is $\|\mathcal{L} - \mathcal{L}_{est}\| = 0.0067$.

2) *Estimation of three lines*: This section presents the results for three different straight lines. In this case $m = 9$ and $p = 6$, meaning that we have nine measurements and six unknowns, then $\Omega\Omega^T \in \mathbb{R}^{6 \times 6}$ (see the definition of the state in the beginning of Sec. II-B). We considered the variables to optimize to be the mean of the eigenvalues associated with each line, thus

$$\sigma_{3lines}^2 = \frac{1}{3} \sum_{i=1}^3 \sigma_i^2, \quad (26)$$

where $\sigma_i^2 = [\sigma_{i,1}^2, \sigma_{i,2}^2]^T$. The Jacobian matrix, which yields the relation between the new eigenvalues and the camera's linear velocity is then

$$\mathbf{J}_{\nu_c, 3lines} = \frac{1}{3} \sum_{i=1}^3 \mathbf{J}_{\nu_c, i}. \quad (27)$$

The experimental setup is similar to the one in Sec. IV-B.1. A chessboard is used to obtain the initial *Plücker* coordinates of the 3D straight lines. Throughout the experiment lines are tracked with the moving edges tracker of *ViSP*, the tracking of the lines in a camera picture is presented in Fig. 2(b). Fig. 4 presents the experimental results for this case, with the following gains $k_1, k_2 = 1$, $\alpha = 1000$, and $\sigma_{des}^2 = [0.1, 0.2]^T$. Figs. 4(a), 4(b), and 4(c) present the state error estimation for the first, second, and third line respectively. Fig. 4(d) presents the camera's velocities, which are noisy, since they are retrieved from the robot's odometry. Finally, Fig. 4(e) presents a 3D plot with the position of each line (real and estimated coordinates). Besides it presents the final position of the camera, and its path. The total error of the *Plücker* coordinates are $\|\mathcal{L}_1 - \mathcal{L}_{est,1}\| = 0.0388$, $\|\mathcal{L}_2 - \mathcal{L}_{est,2}\| = 0.0109$, and $\|\mathcal{L}_3 - \mathcal{L}_{est,3}\| = 0.0217$.

V. CONCLUSIONS

This paper has proposed a strategy for *Active Structure-from-Motion* for 3D straight lines. Contrarily to previous approaches, *binormalized Plücker coordinates* were used to represent lines, which allow for an explicit representation. A

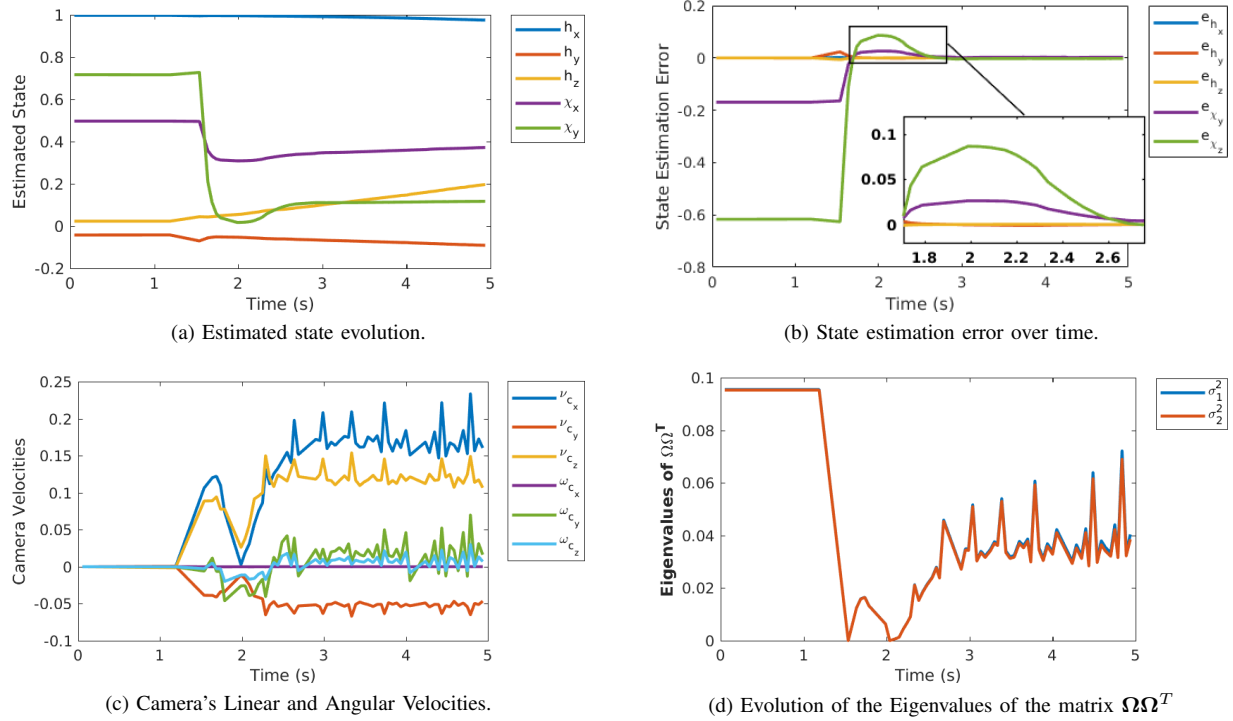


Fig. 3: Experimental results with a real robot and camera for a single line. The estimated state is presented in the top left plot, and the corresponding state estimation error in the top right plot. The velocities of the camera throughout the task are presented in the bottom left plot. Finally, the eigenvalues of the matrix $\Omega\Omega^T$ are presented in the bottom right plot.

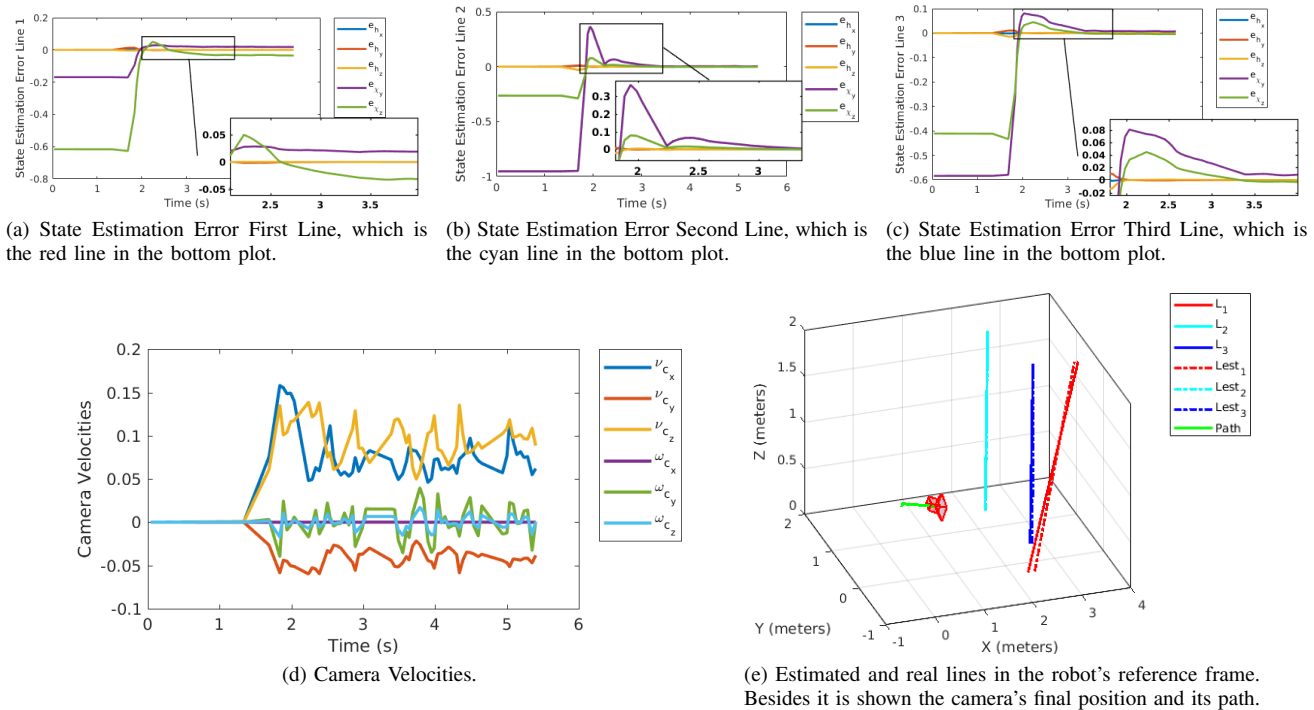


Fig. 4: Experimental results with a real robot and camera for three lines. The estimation state errors are presented for each line (top three plots). The velocities of the camera throughout the task are presented in the bottom left plot. A 3D plot of the estimated and real lines, alongside the camera and its path is presented in the bottom right.

variable change of these coordinates has been presented to comply with the requirements of a recently presented framework for *Active Structure-from-Motion*. The new dynamics of the line, and an observer for retrieving the lines' 3D structure were then proposed. A control law was used with the purpose of keeping the control effort relatively low, while achieving a desired convergence rate. This approach was validated in simulation and afterwards in a real robotic platform, for both one and three lines. Future work consists in including both the algebraic constraints on the *Plücker* coordinates in the estimation scheme, and the observer in a Image-Based Visual Servoing control scheme.

REFERENCES

- [1] F. Chaumette and S. Hutchinson, "Visual Servo Control. I. Basic Approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [2] N. Andreff, B. Espiau, and R. Horaud, "Visual Servoing from Lines," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.
- [3] J. Koenderink and A. Van Doorn, "Affine structure from motion," *J. Optical Society of America A (JOSA)*, vol. 8, no. 2, pp. 377–385, 1991.
- [4] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 416–441, 2005.
- [5] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.
- [6] S. Soatto, R. F., and P. Perona, "Motion Estimation via Dynamic Vision," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 393–413, 1996.
- [7] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.
- [8] J. Civera, O. Grasa, A. Davison, and J. Montiel, "1-Point RANSAC for Extended Kalman Filtering: Application to Real-Time Structure from Motion and Visual Odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.
- [9] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature Depth Observation for Image-Based Visual Servoing: Theory and Experiments," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.
- [10] W. Dixon, Y. Fang, D. Dawson, and T. Flynn, "Range Identification for Perspective Vision Systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2232–2238, 2003.
- [11] F. Morbidi, G. Mariottini, and D. Prattichizzo, "Observer design via Immersion and Invariance for vision-based leader-follower formation control," *Automatica*, vol. 46, no. 1, pp. 148–154, 2010.
- [12] P. Corke, "Spherical Image-Based Visual Servo and Structure Estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 5550–5555.
- [13] M. Sassano, D. Carnevale, and A. Astolfi, "Observer design for range and orientation identification," *Automatica*, vol. 46, no. 8, pp. 1369–1375, 2010.
- [14] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
- [15] A. Dani, N. Fischer, and W. Dixon, "Single Camera Structure and Motion," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 238–243, 2012.
- [16] V. Grabe, H. Bühlhoff, D. Scaramuzza, and P. Robuffo Giordano, "Nonlinear Ego-Motion Estimation from Optical Flow for Online Control of a Quadrotor UAV," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.
- [17] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active Vision," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.
- [18] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure From Controlled Motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 492–504, 1996.
- [19] R. Spica and P. Robuffo Giordano, "A Framework for Active Estimation: Application to Structure from Motion," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 7647–7653.
- [20] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere, and Cylinder," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.
- [21] —, "Plane Estimation by Active Vision from Point Features and Image Moments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6003–6010.
- [22] P. Robuffo Giordano, R. Spica, and F. Chaumette, "Learning the Shape of Image Moments for Optimal 3d Structure Estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5990–5996.
- [23] O. Tahri, P. Robuffo Giordano, and Y. Mezouar, "Rotation Free Active Vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3086–3091.
- [24] O. Tahri, D. Boutat, and Y. Mezouar, "Brunovsky's linear form of incremental structure from motion," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1491–1499, 2017.
- [25] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Coupling active depth estimation and visual servoing via a large projection operator," *The International Journal of Robotics Research*, vol. 36, no. 11, pp. 1177–1194, 2017.
- [26] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.
- [27] E. Rosten and T. Drummond, "Fusing Points and Lines for High Performance Tracking," in *IEEE International Conference on Computer Vision, 2005. ICCV 2005*, vol. 2, 2005, pp. 1508–1515.
- [28] J. Messias, R. Ventura, P. Lima, J. Sequeira, P. Alvito, C. Marques, and P. Carriço, "A Robotic Platform for Edutainment Activities in a Pediatric Hospital," in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2014, pp. 193–198.
- [29] H. Pottmann and J. Wallner, *Computational Line Geometry*. Secaucus, NJ, USA: Springer, 2001.
- [30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [31] S. L. V. Campbell, *Singular systems of differential equations II*. Pitman Publishing (UK), 1982.
- [32] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009, p. 5.
- [33] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, 2005.
- [34] 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.
- [35] FLIR Cameras. (2018) Flea3 usb3 vision cameras for industrial, life science, traffic, and security applications. Accessed: 2018-07-27. [Online]. Available: <https://www.ptgrey.com>
- [36] D. Oberkampf, D. F. DeMenthon, and L. Davis, "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding (CVIU)*, vol. 63, no. 3, pp. 495–511, 1996.