Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 10768–10773

3D Feature-Based Sampled-Data Visual Tracking ⋆

Marco Costanzo * **Giuseppe De Maria** * **Ciro Natale** * **Antonio Russo** *

* *Dipartimento di Ingegneria, Università degli Studi della Campania "Luigi Vanvitelli", Aversa, 81031 ITALY (e-mail: {marco.costanzo,giuseppe.demaria,ciro.natale,antonio.russo1}@unicampania.it)*

Abstract: 3D feature-based Visual Servoing (VS) on the one hand shows attractive peculiarities, on the other hand it suffers from drawbacks related to the existence of local minima, which may affect the convergence character of the VS control loop. Furthermore, the performance of the visual tracking module may constitute a bottleneck enforcing severe constraints on the workspace and visual task execution speed. In this paper we introduce a novel sampled-data model of the 3D feature-based VS, and, in order to avoid drawbacks due to local minima, we plan the target reference trajectory in the feature space with the aim to constraint the feature error dynamics to remain close to the desired equilibrium point. Then, we propose a novel feature generation based on the homography provided by a template matching algorithm based on the Zero mean Normalized Cross Correlation (ZNCC) and the design of a visual tracking scheme by resorting to the Extended Kalman Filter (EKF) and Lyapunov direct method, which explicitly takes into account the camera velocity limits, while ensuring stability.

Keywords: Robots manipulators, Perception and sensing, Guidance navigation and control, Tracking, Lyapunov methods.

1. INTRODUCTION

The objective of accurately positioning the robot end effector with respect to a target object in dynamic environments is a challenge in robotic vision control. This is due to the scene changing over time, which may generate temporary partial occlusions, variation of object location and lighting conditions. Classical vision controllers (refer to Corke (2017) for an up-to-date textbook on visual servoing), designed to achieve the above objective are based on features extracted from the image acquired by the camera mounted on the end effector (*eye-in-hand* configuration) and can be grouped in two main categories, namely Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS). The former uses the extracted features to estimate the camera pose directly in the 3D space with the aim to minimize the mismatch between the current and the target pose, while the latter aims at minimizing the error between the current image features and the target ones directly in the feature space. The availability of cheap and lightweight RGB-D cameras promotes the use of 3D features, obtained by combining 2D and depth information (Cervera et al., 2003; Costanzo et al., 2022), which allows the design of 3D feature-based VS. Feature-based VS schemes (IBVS and 3D feature-based VS), differently from PBVS, do not require the 3D pose reconstruction, they are robust against camera calibration errors and, since the image features are directly controlled, the feature-based VS controller will likely be able to keep them in the camera field of view (Cervera et al., 2003; Costanzo et al., 2022). The features extracted from the current image are matched with the ones of the target image acquired beforehand, and, then the tracking module provides, frame by frame, the feature vector as input to the visual controller. The tracking module is the core of any vision control scheme and it should be accurate and robust against lighting conditions and object partial occlusions and motion, as well as it should guarantee alignment accuracy of the current image with the target one. Tracking methods can be feature-based, i.e. they track geometrical primitives or object contour, or model-based which use a model of the tracked object. Among the latter methods, we refer to those based on 2D models that are reference template on the target image. The objective is to estimate the warp between the current image and the reference one. The estimation is obtained by a nonlinear optimization process based on a measure of the alignment between the two images. The result of the estimation process can be either linear translation parameters or an homography, which links the current and the reference image. The model-based approaches differ on the measure function used in the estimation process. The standard one is the Sum of Squared Differences (SSD) function (Lucas and Kanade, 1981; Baker and Matthews, 2004; Benhimane and Malis, 2007), but, unluckily, this approach is very sensitive to lighting conditions and occlusions. More robust methods are those using similarity measures invariant to changes in brightness and contrast, such as Mutual

⋆ This work was partially supported by the European Commission within H2020 REMODEL Project (GA no. 870133).

10.1016/j.ifacol.2023.10.742

Information (MI) (Dame and Marchand, 2010) or Zero mean Normalized Cross Correlation (ZNCC) (Di Stefano et al., 2005). Even if MI presents a significant optimum, its convergence basin is very small with respect to ZNCC and this makes it sensitive to the camera velocity. In this paper we use the ZNCC measure to estimate the homography, which links the current image and the reference template designed by the user on the image of a planar target. Then, the homography is used to generate, frame by frame, artificial features on the current image, matched with the ones arbitrarily selected by the user within the reference template, as input to the visual controller.

In order to plan the motion of the camera in terms of displacement, velocity and acceleration and to avoid an anomalous character of the feature trajectories, discussed in Costanzo et al. (2022), due to the existence of local minima, we design a time-varying reference feature trajectory starting from the features selected within the template on the target image and the ones generated by means of the homography estimated in the initial condition. If the visual servoing task starts with a feature error equal to zero, then the feature error dynamics is constrained to stay close to the desired equilibrium point. Arbitrary motion of the target object, (e.g., an object handled by a human during an handover maneuver or a picking operation of an object on a conveyor belt), may affect the performance of the visual tracker. To avoid this drawback, we estimate the object velocity by means of an Extended Kalman Filter (EKF) based on a sampled-data model of the VS introduced in this paper without resorting to the forward Euler approximation. Then, we add it to the camera velocity command to compensate the relative motion, not related to the reference feature, between the camera and the object, thus significantly reducing tracker sensitivity to the object velocity. Finally, we propose a visual controller by resorting to the Lyapunov direct method, which explicitly takes into account the camera velocity limits, while ensuring stability at the same time.

2. 3D FEATURE-BASED VISUAL SERVOING SAMPLED DATA MODEL

The VS consists in the design of a closed-loop control scheme that regulates the current image feature set to the given target of an image acquired beforehand. Since the sampling rate that can be achieved by the visual perception pipeline is much lower then the one of the low-level robot control loop, the sampled-data nature of the visual servoing should be taken into account in order to allow maximum possible performance of the VS task.

Let $\boldsymbol{s}_i, \boldsymbol{s}^\star_{f_i}, \boldsymbol{s}^\star_i \in \mathbb{R}^3, i = 1, \dots, n$ be the ith 3D-point feature vectors of the current, target and reference image, respectively, and let $\boldsymbol{s} = \begin{bmatrix}\boldsymbol{s}_1^\top & \dots & \boldsymbol{s}_n^\top\end{bmatrix}^\top \in \mathbb{R}^{3n}$, $\boldsymbol{s}^\star_f = \begin{bmatrix}\boldsymbol{s}^{\star\top}_{f1} & \dots & \boldsymbol{s}^{\star\top}_{fn}\end{bmatrix}^\top \in \mathbb{R}^{3n}$ be the vectors of n matched image features. The reference features $\boldsymbol{s}^\star_i(t)$ are designed starting from the initial features $\boldsymbol{s}_i(0)$ to the target one $\boldsymbol{s}^\star_{f_i}$ to plan the feature trajectory during the visual servoing task.

Denote with $\boldsymbol{v} = \begin{bmatrix}\mathbf{v}^\top & \boldsymbol{\omega}^\top\end{bmatrix}^\top \in \mathbb{R}^6$ the body velocity screw of the camera, where $\mathbf{v}$ is the linear velocity and $\boldsymbol{\omega}$ is the angular velocity. The relationship between the rate of $\boldsymbol{s}(t)$ and the velocity screw $\boldsymbol{v}(t)$ is described as

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{L}(\boldsymbol{s})\boldsymbol{v}(t), \quad \boldsymbol{s}(0) = \boldsymbol{s}_0, \tag{1}$$

where $\boldsymbol{L}(\boldsymbol{s}) \in \mathbb{R}^{3n\times 6}$ is the so-called *interaction matrix* which, due to the particular feature set selected, takes the structure

$$\boldsymbol{L}(\boldsymbol{s}) = \begin{bmatrix} -\boldsymbol{I}_3 & \dots & -\boldsymbol{I}_3 \\ \boldsymbol{S}^\top(\boldsymbol{s}_1) & \dots & \boldsymbol{S}^\top(\boldsymbol{s}_n)\end{bmatrix}^\top, \tag{2}$$

where $\boldsymbol{I}_3$ indicates the 3×3 identity matrix and $\boldsymbol{S}(\boldsymbol{s}_i)$ is the skew symmetric operator applied to the vector $\boldsymbol{s}_i$, such that $\boldsymbol{S}(\boldsymbol{s}_i)\boldsymbol{\omega} = \boldsymbol{s}_i \times \boldsymbol{\omega}$.

By defining the feature error vector $\boldsymbol{e} \in \mathbb{R}^{3n}$ as

$$\boldsymbol{e}(t) = \boldsymbol{s}(t) - \boldsymbol{s}^\star(t), \tag{3}$$

denoting with $\boldsymbol{L}_e = \boldsymbol{L}(\boldsymbol{e}+\boldsymbol{s}^\star)$, the dynamics of the current feature error can be easily derived from (1) and (3) as

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{L}_{\boldsymbol{e}}\boldsymbol{v}(t) - \dot{\boldsymbol{s}}^\star(t), \quad \boldsymbol{e}(0) = \boldsymbol{e}_0. \tag{4}$$

In the following, the dependence from the time will be dropped to simplify the notation.

Taking into account the structure of $\boldsymbol{L}(\boldsymbol{s})$ in (2), the term at the right-hand side of equation (1) can be written as $\boldsymbol{L}(\boldsymbol{s})\boldsymbol{v} = \bar{\boldsymbol{S}}^\top(\boldsymbol{\omega})\boldsymbol{s} - \bar{\mathbf{v}}$ and the feature dynamics can be expressed as

$$\dot{\boldsymbol{s}} = \bar{\boldsymbol{S}}^\top(\boldsymbol{\omega})\boldsymbol{s} - \bar{\mathbf{v}}, \tag{5}$$

where $\bar{\boldsymbol{S}}(\boldsymbol{\omega}) \in \mathbb{R}^{3n\times 3n}$ and $\bar{\mathbf{v}} \in \mathbb{R}^{3n}$ have the following expressions

$$\bar{\boldsymbol{S}}(\boldsymbol{\omega}) = \boldsymbol{I}_n \otimes \boldsymbol{S}(\boldsymbol{\omega}), \ \bar{\mathbf{v}} = \mathbf{1}_n \otimes \mathbf{v}. \tag{6}$$

where the symbol $\otimes$ indicates the Kronecker product, $\boldsymbol{I}_n$ is the $n\times n$ identity matrix and $\mathbf{1}_n = [1\ 1\ \dots\ 1]^\top \in \mathbb{R}^n$.

Denoting with T the control sampling time, the camera velocity control is kept constant at the value $\boldsymbol{v}(kT) = \boldsymbol{v}_k = \begin{bmatrix}\mathbf{v}_k^\top & \boldsymbol{\omega}_k^\top\end{bmatrix}^\top$ in the time interval $[kT,\, (k+1)T)$. Therefore, $\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)$ is updated at each step k and is constant within this time interval. The feature dynamics can be computed as the solution of (5) in $[kT,\, (k+1)T)$, i.e.,

$$\boldsymbol{s}(kT+t) = e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)t}\boldsymbol{s}(kT) - \int_{kT}^{kT+t} e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)(kT+t-\tau)}\,\mathrm{d}\tau\bar{\mathbf{v}}_k \tag{7}$$

for $t \in [0,\, T)$. The feature dynamics in (7), in the discrete time k, is similar to that of switched systems. Relying on the results in Liberzon and Morse (1999), for $t = T$ and by a suitable change of variable, the feature dynamics in the sampling instants is governed by

$$\boldsymbol{s}_{k+1} = e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)T}\boldsymbol{s}_k - \int_0^T e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)\sigma}\,\mathrm{d}\sigma\bar{\mathbf{v}}_k, \ \ \forall k \in \mathbb{N}_0, \tag{8}$$

where $\boldsymbol{s}_k$ stands for $\boldsymbol{s}(kT)$.

By expanding the exponential matrices, in the previous equation, in power series, it holds that

$$\int_0^T e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)\sigma}\,\mathrm{d}\sigma = T\sum_{i=0}^{\infty}\frac{(\bar{\boldsymbol{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{(i+1)!}, \tag{9}$$

thus, substituting (9) in (8) and rearranging the terms, equation (8) becomes

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \boldsymbol{P}(\boldsymbol{\omega}_k)\boldsymbol{L}(\boldsymbol{s}_k)\boldsymbol{v}_k, \tag{10}$$

with $\boldsymbol{P}(\cdot) : \mathbb{R}^3 \to \mathbb{R}^{3n\times 3n}$ defined as

$$\boldsymbol{P}(\boldsymbol{\omega}_k) = \int_0^T e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)\sigma}\,\mathrm{d}\sigma. \tag{11}$$

Equation (10) represents the sampled-data feature dynamic model, which describes exactly the behaviour of the VS system at the sampling instants.

Now, by denoting with $\boldsymbol{e}_k = \boldsymbol{e}(kT)$, then it is straightforward to derive the feature error dynamics as

$$\boldsymbol{e}_{k+1} = \boldsymbol{e}_k + \boldsymbol{P}(\boldsymbol{\omega}_k)\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{v}_k - \Delta\boldsymbol{s}^\star_k, \tag{12}$$

where $\boldsymbol{L}_{\boldsymbol{e}_k} = \boldsymbol{L}(\boldsymbol{e}_k + \boldsymbol{s}^\star)$ and $\boldsymbol{v}_k = \left[\mathbf{v}_k^\top \ \boldsymbol{\omega}_k^\top\right]^\top \in \mathbb{R}^6$ and $\Delta\boldsymbol{s}^\star_k = \boldsymbol{s}^\star_{k+1} - \boldsymbol{s}^\star_k$ is the forward finite difference of $\boldsymbol{s}^\star_k$.

3. TIME-VARYING REFERENCE FEATURE GENERATION

The time-varying reference $\boldsymbol{s}^\star(t)$ is designed to keep the feature error close to the origin during the whole task as to avoid the undesired behaviours discussed in (Costanzo et al., 2022). Such a reference can be generated by interpolating each feature between the initial value $\boldsymbol{s}(0)$ and the target one $\boldsymbol{s}^\star_f$ in a given time t_f. The interpolation is obtained by applying the time-varying rigid transformation $\boldsymbol{T}^\star(t) = \boldsymbol{T}(\theta^\star(t), \boldsymbol{r}^\star(t), \boldsymbol{p}^\star(t)) \in \mathbb{R}^{4\times4}$ representing a rotation $\theta^\star$ about a rotation axis $\boldsymbol{r}^\star$, and a translation $\boldsymbol{p}^\star$, to each feature. Thus, the reference features are obtained as

$$\tilde{\boldsymbol{s}}^\star_i(t) = \boldsymbol{T}^\star(t)\tilde{\boldsymbol{s}}_i(0), \ \ i = 1, \dots, n, \tag{13}$$

where $\tilde{\boldsymbol{s}}^\star_i$ and $\tilde{\boldsymbol{s}}_i$ indicate the vector of homogeneous coordinates of the feature $\boldsymbol{s}^\star_i$ and $\boldsymbol{s}_i$, respectively.

Then, assuming $\boldsymbol{r}^\star(t) = \boldsymbol{r}^\star_f$ constant, the designed homogeneous transformation $\boldsymbol{T}(\theta^\star, \boldsymbol{r}^\star, \boldsymbol{p}^\star)$ should be such that

$$\boldsymbol{T}(\theta^\star(0), \boldsymbol{r}^\star_f, \boldsymbol{p}^\star(0)) = \boldsymbol{T}^\star(0) = \boldsymbol{I}_4, \tag{14}$$

$$\boldsymbol{T}(\theta^\star(t_f), \boldsymbol{r}^\star_f, \boldsymbol{p}^\star(t_f))\tilde{\boldsymbol{s}}_i(0) = \boldsymbol{T}^\star\tilde{\boldsymbol{s}}_i(0) = \tilde{\boldsymbol{s}}^\star_{f_i}. \tag{15}$$

In the following we design $\theta^\star$ and $\boldsymbol{p}^\star$ so that they evolve in time as a fifth-order polynomial between $\theta^\star(0) = 0$, $\boldsymbol{p}^\star(0) = 0$ and $\theta^\star(t_f)$, $\boldsymbol{p}^\star(t_f)$, respectively.

Recalling (15), the homogeneous transformation matrix $\boldsymbol{T}^\star(t_f)$ can be found by solving the following problem

$$\min_{\boldsymbol{T}} \frac{1}{2}\sum_1^n{}_i \left\|\boldsymbol{T}\tilde{\boldsymbol{s}}_i(0) - \tilde{\boldsymbol{s}}^\star_{f_i}\right\|^2. \tag{16}$$

To ensure that $\boldsymbol{T}$ is an homogeneous transformation matrix, we parametrize it with a position vector and a unit quaternion.

4. HOMOGRAPHY-BASED 3D FEATURE GENERATION

Measurement of the 3D features can be accomplished by tracking 2D features on the camera image in pixel coordinates and then transforming them in the 3D space via the RGB-D camera intrinsic parameters.We assume that the object to be tracked has a planar textured face on which the features are located (the generalization to the non-planar case is straightforward (Benhimane and Malis, 2007)). Let $[u_{ik} \ v_{ik}]^\top$, $[u^\star_{fi} \ v^\star_{fi}]^\top$ be the projection of the current 3D feature $\boldsymbol{s}_{ik}$ and target $\boldsymbol{s}^\star_{f_i}$ on the image plane in pixel coordinates, respectively. Since the 3D features are located on a planar face, the relation between the features in the image plane is given by the homography matrix $\boldsymbol{H}_k \in \mathbb{R}^{3\times3}$ (Benhimane and Malis, 2007), i.e,

$$[\mu_{ik}u_{ik} \ \mu_{ik}v_{ik} \ \mu_{ik}]^\top = \boldsymbol{H}_k \left[u^\star_{fi} \ v^\star_{fi} \ 1\right]^\top \tag{17}$$

where μ_{ik} is an auxiliary variable.

In our approach, we estimate the homography $\boldsymbol{H}_k$ via a template-tracking algorithm based on ZNCC (Di Stefano et al., 2005). The algorithm needs to be initialized by selecting the template on the target image and by determining an estimation of the initial homography $\boldsymbol{H}_0$. To estimate $\boldsymbol{H}_0$, we do as follows. Given the initial and target image, couples of 2D features are matched by means of the matching algorithm provided by the ViSP library (Marchand et al., 2005) (`vpKeyPoint` class). The matched features are used to solve (17) for $\boldsymbol{H}_0$ by using the Ransac algorithm (`vpHomography` class). Then, the adopted tracker (`vpTemplateTrackerZNCCInverseCompositional` class) solves, at each time step, the following optimization problem to find the homography parameters (Dame and Marchand, 2010)

$$\hat{\boldsymbol{h}}_k = \arg\max_{\boldsymbol{h}} f(\boldsymbol{\mathcal{I}}^\star, w(\boldsymbol{\mathcal{I}}_k, \boldsymbol{h})) \tag{18}$$

where $\boldsymbol{h} \in \mathbb{R}^8$ are the free parameters of the homography matrix $\boldsymbol{H}_k$, $\boldsymbol{\mathcal{I}}^\star$ and $\boldsymbol{\mathcal{I}}_k$ are the target image template and the current one, respectively, $f(\cdot,\cdot)$ is the similarity function that checks the similarity of two images via the ZNCC algorithm, and $w(\cdot,\cdot)$ is the warping function that transforms the template by applying the homography to each pixel as in (17).

For the control purpose, we select the target features $[u^\star_{fi} \ v^\star_{fi}]^\top$ arbitrarily within the target template, then we generate the matched current features $[u_{ik} \ v_{ik}]^\top$ via the homography (17).

5. OBJECT VELOCITY ESTIMATION

The object motion generates an additional velocity $\boldsymbol{v}_d$ with respect to the camera frame, i.e., $\boldsymbol{v} = \boldsymbol{v}_c + \boldsymbol{v}_d$, where $\boldsymbol{v}_c$ is the actual camera velocity command. By denoting with $\boldsymbol{v}_c = \left[\mathbf{v}_c^\top \ \boldsymbol{\omega}_c^\top\right]^\top$ and $\boldsymbol{v}_d = \left[\boldsymbol{\eta}^\top \ \boldsymbol{\zeta}^\top\right]^\top$, where $\boldsymbol{\eta} \in \mathbb{R}^3$ and $\boldsymbol{\zeta} \in \mathbb{R}^3$ are the linear and angular components of the object velocity, respectively, $\boldsymbol{v}_d$ can be estimated by resorting to an EKF tailored on the following equations where a constant object velocity model is assumed,

$$\boldsymbol{v}_{d\,k+1} = \boldsymbol{v}_{dk} + \boldsymbol{\nu}_k, \tag{19}$$

$$\begin{aligned}\boldsymbol{s}_{k+1} &= \boldsymbol{s}_k + \boldsymbol{P}(\boldsymbol{\omega}_{ck} + \boldsymbol{\zeta}_k)\boldsymbol{L}(\boldsymbol{s}_k)\,(\boldsymbol{v}_{ck} + \boldsymbol{v}_{dk}) + \boldsymbol{\sigma}_k,\\ &= \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{\eta}_k, \boldsymbol{\zeta}_k) + \boldsymbol{\sigma}_k\end{aligned} \tag{20}$$

$$\boldsymbol{y}_k = \boldsymbol{s}_k + \boldsymbol{\chi}_k, \tag{21}$$

where $\boldsymbol{\nu}_k$, $\boldsymbol{\sigma}_k$, $\boldsymbol{\chi}_k$ are Gaussian process noise with covariance matrices $\boldsymbol{\Gamma} \in \mathbb{R}^{6\times6}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{3n\times3n}$ and $\boldsymbol{X} \in \mathbb{R}^{3n\times3n}$, respectively, and $\boldsymbol{y}_k$ is the system measurable output. Computation of the Jacobian matrix $\boldsymbol{J}$ of system (19)-(20) is not trivial and it is obtained as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 \\ \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{\eta}} & \begin{bmatrix}\dfrac{\partial \boldsymbol{f}}{\partial \zeta_1} & \dfrac{\partial \boldsymbol{f}}{\partial \zeta_2} & \dfrac{\partial \boldsymbol{f}}{\partial \zeta_3}\end{bmatrix} & \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{s}} \end{bmatrix}, \tag{22}$$

where

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\eta}} = -\boldsymbol{P}(\boldsymbol{\omega}_c + \boldsymbol{\zeta})\,(\mathbf{1}_n \otimes \boldsymbol{I}_3)\,, \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{s}} = e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_c + \boldsymbol{\zeta})},$$

$$\frac{\partial \boldsymbol{f}}{\partial \zeta_i} = \frac{\partial \boldsymbol{P}(\boldsymbol{\omega}_c + \boldsymbol{\zeta})}{\partial \zeta_i}\boldsymbol{L}(\boldsymbol{s}) \begin{bmatrix}\mathbf{v}_c + \boldsymbol{\eta} \\ \boldsymbol{\omega}_c + \boldsymbol{\zeta}\end{bmatrix} + \boldsymbol{P}(\boldsymbol{\omega}_c + \boldsymbol{\zeta})\boldsymbol{L}(\boldsymbol{s}) \begin{bmatrix}\mathbf{0}_3 \\ \boldsymbol{l}_i\end{bmatrix}$$

where $\boldsymbol{l}_i$ is the ith canonical orthonormal basis versor.

In order to compute the terms $\partial \boldsymbol{P}/\partial \zeta_i$, recalling that $\boldsymbol{\omega} = \boldsymbol{\omega}_c + \boldsymbol{\zeta}$, by virtue of the Euler-Rodrigues rotation formula

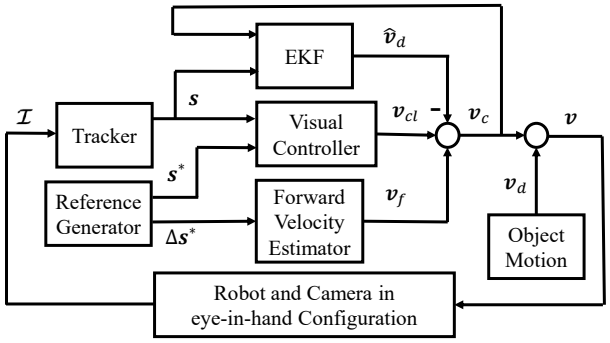

Fig. 1. Visual control system block scheme.

(Dai (2015)), matrix $\boldsymbol{P}(\boldsymbol{\omega}_k)$ in the system dynamics (20) can be written as

$$\boldsymbol{P}(\boldsymbol{\omega}_k) = T\left[\boldsymbol{I}_{3n} + \frac{1-\cos(\|\boldsymbol{\omega}_k\|T)}{\|\boldsymbol{\omega}_k\|T}\bar{\boldsymbol{S}}^\top\left(\frac{\boldsymbol{\omega}_k}{\|\boldsymbol{\omega}_k\|}\right) + \frac{\|\boldsymbol{\omega}_k\|T-\sin(\|\boldsymbol{\omega}_k\|T)}{\|\boldsymbol{\omega}_k\|T}\bar{\boldsymbol{S}}^2\left(\frac{\boldsymbol{\omega}_k}{\|\boldsymbol{\omega}_k\|}\right)\right]. \quad (23)$$

From (23) it holds that

$$\frac{\partial \boldsymbol{P}}{\partial \zeta_i} = T\left[\phi_1'(\|\boldsymbol{\omega}\|)\omega_i\bar{\boldsymbol{S}}^\top(\boldsymbol{\omega}) + \phi_1(\|\boldsymbol{\omega}\|)\bar{\boldsymbol{S}}^\top(\boldsymbol{l}_i) + \phi_2'(\|\boldsymbol{\omega}\|)\omega_i\bar{\boldsymbol{S}}^2(\boldsymbol{\omega}) + \phi_2(\|\boldsymbol{\omega}\|)(\bar{\boldsymbol{S}}(\boldsymbol{l}_i)\bar{\boldsymbol{S}}(\boldsymbol{\omega}) + \bar{\boldsymbol{S}}(\boldsymbol{\omega})\bar{\boldsymbol{S}}(\boldsymbol{l}_i))\right]$$

where

$$\phi_1(x) = \sum_{i=1}^{\infty}\frac{(-1)^i x^{2i-2}T^{2i-1}}{(2i)!},\quad \phi_2(x) = \sum_{i=1}^{\infty}\frac{(-1)^i x^{2i-2}T^{2i}}{(2i+1)!},$$

with $\phi_j'(x) = \frac{1}{x}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}$, $j = 1, 2$.

The effectiveness of the estimation by the EKF will be evaluated in Section 7.

6. VISUAL TRACKING

Figure 1 shows the control system architecture. The camera velocity command $\boldsymbol{v}_c$ is obtained as sum of three terms

$$\boldsymbol{v}_c = -\hat{\boldsymbol{v}}_d + \boldsymbol{v}_f + \boldsymbol{v}_{cl}. \quad (24)$$

The first term $\hat{\boldsymbol{v}}_d$ is the estimation of the object velocity by the EKF described in Section 5, its purpose is to counteract the object velocity $\boldsymbol{v}_d$ in the block scheme; $\boldsymbol{v}_f$ represents the forward control action which is designed to balance the term $\Delta\boldsymbol{s}_k^\star$ in (12) and can be computed by solving the following implicit equation

$$\boldsymbol{v}_{fk} = \begin{bmatrix}\mathbf{v}_{fk}\\ \boldsymbol{\omega}_{fk}\end{bmatrix} : \boldsymbol{P}(\boldsymbol{\omega}_{fk})\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{v}_{fk} = \Delta\boldsymbol{s}_k^\star. \quad (25)$$

This equation can be solved by reformulating the problem as a minimization one as follows

$$\min_{\boldsymbol{v}_{f_k}} \quad \|\boldsymbol{P}(\boldsymbol{\omega}_{fk})\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{v}_{fk} - \boldsymbol{\Delta}\boldsymbol{s}_k^\star\|^2. \quad (26)$$

The last term in (24) is the closed loop control action $\boldsymbol{v}_{cl}$ and it is designed to ensure the stability of the VS scheme. Such a component will be derived on system (12) assuming that the estimated relative velocity cancels the actual one ($\hat{\boldsymbol{v}}_d = \boldsymbol{v}_d$). Thus (12), in view of (25), reduces to

$$\boldsymbol{e}_{k+1} = \boldsymbol{e}_k + \boldsymbol{P}(\boldsymbol{\omega}_{clk})\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{v}_{clk}. \quad (27)$$

With the aim to design a velocity control law that aims at guaranteeing a decoupled and exponential decrease of the error in the case of pure translational motion (Corke, 2017; Costanzo et al., 2022), we assume for the system in (27) a reference model $\boldsymbol{\epsilon}_{k+1} = (1-\lambda_k T)\boldsymbol{\epsilon}_k$ with $0 < \lambda_k T < 1$, which assures in the case of pure translational motion a decoupled and convergent dynamics without hidden oscillations. The equivalent control $\boldsymbol{v}_{clk}$, which assures a perfect model reference following in pure translation, is $\boldsymbol{v}_{clk} = -\lambda_k\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k$ where λ_k is the control gain. Thus, the closed-loop dynamics becomes

$$\boldsymbol{e}_{k+1} = \left(\boldsymbol{I}_{3n} - \lambda_k\boldsymbol{P}_k\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\right)\boldsymbol{e}_k. \quad (28)$$

The control law proposed in the following proposition requires the definition of two scalar functions, i.e., $\pi(\cdot,\cdot)$, the *cost function*, and $l(\cdot)$, the *landing function*. The former is denoted as

$$\pi(\boldsymbol{e}_k, \lambda_k T) = -\lambda_k T\left(2l(\boldsymbol{e}_k) - \alpha_k\lambda_k T\|\boldsymbol{\omega}_k^1\| - \lambda_k T\right), \quad (29)$$

where $\boldsymbol{\omega}_k^1 = -[\boldsymbol{O}_3\ \boldsymbol{I}_3]\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k$, $\alpha_k = \|\boldsymbol{e}_k\|/\|\boldsymbol{e}_{Ik}\|$ and $\boldsymbol{e}_{Ik} = \boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k$ is the projection of $\boldsymbol{e}_k$ in $\mathrm{Im}(\boldsymbol{L}_{\boldsymbol{e}})$. The function $\pi(\boldsymbol{e}_k, \lambda_k T)$, as a function of $\lambda_k T$, is a concave up parabola passing through the origin and the point $\lambda_k T = 2l(\boldsymbol{e}_k)/(1+\alpha_k\|\boldsymbol{\omega}_k^1\|)$.

The *landing function* $l(\boldsymbol{e}_k)$, which allows a smooth approach to the reference trajectory, is defined as follows. Denote with $e_{Mk} = \|\boldsymbol{e}_k\|/\sqrt{n}$ the Root Mean Square Error (RMSE), let e_H and e_L be two scalar values such that $e_L < e_H$ and $\beta \in (0,\,1]$, then

$$l(\boldsymbol{e}_k) = \begin{cases}\beta & \text{if } e_{Mk} \le e_L\\ 1 & \text{if } e_{Mk} \ge e_H\\ \beta + (1-\beta)q\left(\frac{e_{Mk}-e_L}{e_H-e_L}\right) & \text{otherwise}\end{cases}, \quad (30)$$

where $q(\cdot)$ is a fifth-order polynomial whose coefficients are determined such that the first and second order derivatives are zero at the extremes of the interval $[0,\,1]$ and $q(0) = 0$, $q(1) = 1$. Note that $0 < l(\boldsymbol{e}_k) \le 1$ and $\beta = 1$ implies $l(\boldsymbol{e}_k) = 1$.

Now denote with ω_k^M and v_k^M the maximum rotational and linear camera velocities, respectively, depending on the selected joint velocity limits and the configuration of the robot performing the task. Moreover, let $\mathcal{D} \subset \mathbb{R}^{3n}$ be an open neighborhood of the equilibrium point $\boldsymbol{e} = \boldsymbol{0}$, which does not contain any other point satisfying the condition $\boldsymbol{L}_{\boldsymbol{e}}^\top\boldsymbol{e} = \boldsymbol{0}$ (see (Costanzo et al., 2022) for details). The following proposition holds.

Proposition 1. If the control gain λ_k is determined for each sampling interval by solving the following problem

$$\min_{\lambda_k T} \quad \pi(\boldsymbol{e}_k, \lambda_k T) \quad (31a)$$

$$\text{s.t.} \quad 0 < \lambda_k T < \frac{2l(\boldsymbol{e}_k)}{1+\alpha_k\|\boldsymbol{\omega}_k^1\|} \quad (31b)$$

$$\|\boldsymbol{\omega}_k\| \le \omega^M, \quad \|\mathbf{v}_k\| \le \mathrm{v}^M, \quad (31c)$$

then the equilibrium point $\boldsymbol{e} = \boldsymbol{0}$ is asymptotically stable in $\mathcal{D}$ and the feature error dynamics does not exhibit hidden oscillations over the continuous time t.

Proof. By selecting as Lyapunov function candidate $V(\boldsymbol{e}_k) = \boldsymbol{e}_k^\top\boldsymbol{e}_k$, its first order variation $\Delta V_k = \boldsymbol{e}_{k+1}^\top\boldsymbol{e}_{k+1} - \boldsymbol{e}_k^\top\boldsymbol{e}_k$ along the error dynamics (28) can be written, by denoting $\boldsymbol{B}_k = \boldsymbol{P}_k - T\boldsymbol{I}_{3n}$ with $\boldsymbol{P}_k = \boldsymbol{P}(\boldsymbol{\omega}_k)$, as

$$\Delta V_k = -2\lambda_k T\boldsymbol{e}_k^\top\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k - 2\lambda_k\boldsymbol{e}_k^\top\boldsymbol{B}_k\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k + \lambda_k^2\boldsymbol{e}_k^\top\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{P}_k^\top\boldsymbol{P}_k\boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\boldsymbol{e}_k. \quad (32)$$

By neglecting high-order terms in the series expansions of functions in (23), the matrix $\boldsymbol{P}_k$ can be approximated as

$$\boldsymbol{P}(\boldsymbol{\omega}_k) \approx T\boldsymbol{I}_{3n} + \frac{1}{2}T^2\bar{\boldsymbol{S}}^\top(\boldsymbol{\omega}_k) \tag{33}$$

thus it results $\left\|\boldsymbol{P}_k^\top \boldsymbol{P}_k\right\| \leq T^2$ and $\|\boldsymbol{B}_k\| \leq \|\boldsymbol{\omega}_k\|T^2/2$. Since α_k is finite in $\mathcal{D}$ and it is lowerbounded by 1 (Costanzo et al., 2022), it is easy to show that ΔV_k can be overbounded as

$$\Delta V_k \leq -\lambda_k T\left(2 - \lambda_k T\alpha_k\left\|\boldsymbol{\omega}_k^1\right\| - \lambda_k T\right)\|\boldsymbol{e}_{Ik}\|^2 \tag{34}$$

implying, in view of (29) and the fact that $l(\boldsymbol{e}_k) \in (0, 1]$

$$\Delta V_k \leq \pi(\boldsymbol{e}_k, \lambda_k T)\|\boldsymbol{e}_{Ik}\|^2. \tag{35}$$

Then asymptotic stability in $\mathcal{D}$ is achieved for the desired equilibrium point if, for each k, λ_k satisfies (31b).
Using (32) and (33) it is easy to show that $\Delta V_k/\left\|\boldsymbol{e}_{Ik}^2\right\|$ as function of $\lambda_k T$ is a concave upward parabola since the last term in (32) is positive definite in $\mathcal{D}$. For each k, the function $\pi(\boldsymbol{e}_k, \lambda_k T)$ has a minimum for $\lambda_k T = \lambda_k^* T$, which is lower than the one for which the function $\Delta V_k/\|\boldsymbol{e}_{Ik}\|^2$ exhibits its minimum value. Then, for $\lambda_k T \leq \lambda_k^* T$, it is very easy to prove that $\|\boldsymbol{e}(kT + t)\|$ for $t \in [0\ T)$ is monotonically decreasing and the feature error dynamics does not exhibits hidden oscillations. If $\lambda_k = \lambda_k^*$ is such that the constraint (31c) is not satisfied the minimization algorithm selects $\lambda_k < \lambda_k^*$ as the camera velocity depends linearly on λ_k. □

Remark 1. If the EKF estimation is not perfect, by denoting with $\tilde{\boldsymbol{v}}_{dk} = [\tilde{\boldsymbol{\eta}}_k^\top\ \tilde{\boldsymbol{\zeta}}_k^\top]^\top = \boldsymbol{v}_{dk} - \hat{\boldsymbol{v}}_{dk}$ and folding it into the feature dynamics (10), the error dynamics in (27) can be rewritten as

$$\boldsymbol{e}_{k+1} = \left(\boldsymbol{I}_{3n} - \lambda_k \boldsymbol{P}_k \boldsymbol{L}_{\boldsymbol{e}_k}\boldsymbol{L}_{\boldsymbol{e}_k}^\dagger\right)\boldsymbol{e}_k + \boldsymbol{\delta}_k, \tag{36}$$

where $\boldsymbol{\delta}_k = \boldsymbol{\delta}(\boldsymbol{e}_k, \tilde{\boldsymbol{v}}_{dk})$ is the cumulative function of all the terms depending on $\tilde{\boldsymbol{v}}_{dk}$, that is limited if the EKF converges. Due to the estimation error, the feature error does not converge to zero but it remains bounded in a neighborhood of $\boldsymbol{e} = \boldsymbol{0}$. To prove the existence of such a bound, we derive ΔV_k by resorting to (33), then an upperbound of ΔV_k can be obtained as follows

$$\begin{aligned}\Delta V_k \leq &-\lambda_k T\left(2 - \lambda_k T\alpha_k\left\|\boldsymbol{\omega}_k^1\right\| - \lambda_k T\right)\|\boldsymbol{e}_{Ik}\|^2 \\ &+ 2(1 + 2\lambda_k T)\|\boldsymbol{\gamma}_k\|\|\boldsymbol{e}_{Ik}\| + \|\boldsymbol{\gamma}_k\|^2,\end{aligned} \tag{37}$$

where terms of order higher than T^2 have been neglected without affecting the sign of the right-hand side, and $\boldsymbol{\gamma}_k = \boldsymbol{\gamma}(\tilde{\boldsymbol{v}}_{dk})$ is a function depending linearly on $\tilde{\boldsymbol{v}}_{dk}$ only (space limitations do not allow us to provide all the details of ΔV_k derivation). This upperbound of ΔV_k, as function of $\|\boldsymbol{e}_{Ik}\|$ with λ_k satisfying constraint (31b), is a concave downward parabola which has only one non-negative root e_{Ik}^*. The greater $\|\boldsymbol{\gamma}_k\|$, the greater e_{Ik}^*. Thus, for $\|\boldsymbol{e}_{Ik}\| > e_{Ik}^*$, ΔV_k is negative definite. The better the EKF estimation, the lower the bound e_{Ik}^*.

7. EXPERIMENTS

The proposed controller has been experimentally tested by using a Kuka LBR iiwa 7 robot equipped with an Intel RealSense D435i RGB-D camera in an *eye-in-hand* configuration. The target image is the one of a plastic bottle acquired beforehand (Figure 2-right). The target features have been selected by the user on the bottle surface (right image), while the current features (left image) are obtained by applying the homography (17) to the target ones.

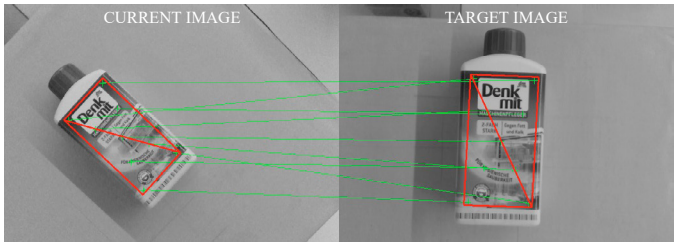

Fig. 2. Target features (right) and matched current ones (left). The area inside the red triangles represents the tracked template.

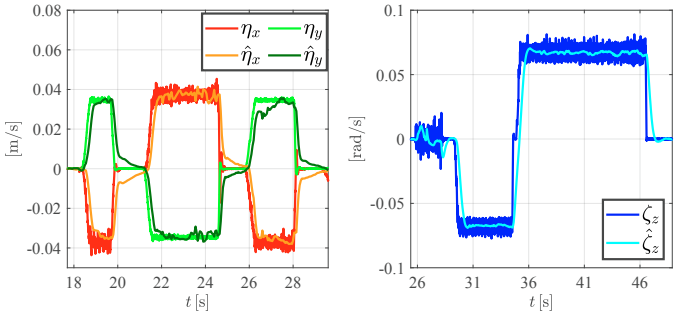

Fig. 3. Exogenous velocity estimated by the EKF: translational (left) and rotational (right).

The first experiment is devoted to test the accuracy of the object velocity estimation by the EKF. To this aim, we open the loop at the output of the first sum node in Fig. 1 and then move the robot using the teach pendant while keeping the object fixed. This way, the velocity $\boldsymbol{v}_d$ is the measured robot Cartesian velocity. The noise covariance matrices are selected as $\boldsymbol{\Gamma} = \text{diag}(5\boldsymbol{I}_3, 20\boldsymbol{I}_3) \times 10^{-7}$, $\boldsymbol{\Sigma} = 10^{-9}\boldsymbol{I}_{3n}$ and $\boldsymbol{X} = 10^{-7}\boldsymbol{I}_{3n}$. The results are shown in Fig. 3, the EKF is able to accurately estimate the camera/object relative velocity, in particular in the time intervals where the relative velocity is constant, the estimated one is a filtered version of the actual velocity measured as robot Cartesian velocity.

In the second experiment, the camera starts from the configuration depicted in Fig. 2. The desired feature trajectory $\boldsymbol{s}^\star$ is generated with the procedure explained in Section 3 with $t_f = 6\,\text{s}$. The forward action velocity $\boldsymbol{v}_f$ is computed by solving problem (26) and the closed loop control $\boldsymbol{v}_{cl}$ is obtained as described in Section 6. When the approach to the target is completed (i.e., when $\boldsymbol{s}^\star = \boldsymbol{s}_f^\star$) a human operator perturbs the object pose by applying both translational and rotational velocities. The experiment is repeated two times, in the first time the EKF estimation $\hat{\boldsymbol{v}}_d$ is not used in the control action (24), while in the second time the estimated velocity is used to counteract the relative velocity $\boldsymbol{v}_d$. Figure 4 shows a comparison between the two task executions. The top plot shows the root mean square errors. The robot moves from $\boldsymbol{s}_0 = \boldsymbol{s}_0^\star$ by following the reference trajectory up to $t = 6\,\text{s}$, then the object is moved by the human operator. In both executions the error is bounded during the object motion, but, when the EKF estimation is not used (blue line), the bound is 2.5 times higher. The bottom plot shows the control gain

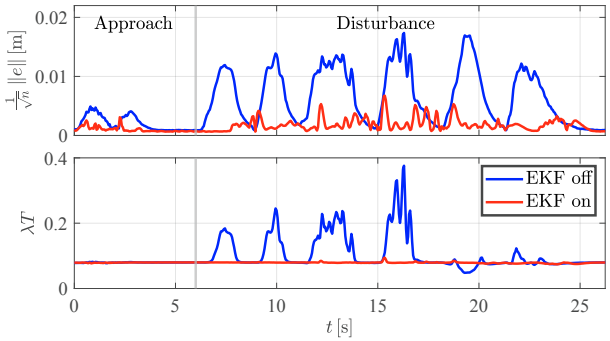

Fig. 4. Second experiment. RMSE (top); control gain λT (bottom).

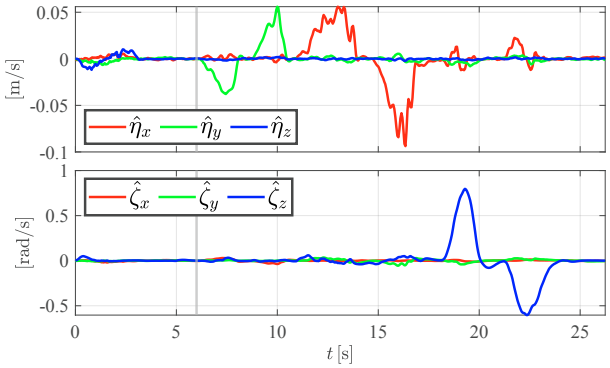

Fig. 5. Second experiment: Estimated velocity in the first task execution.

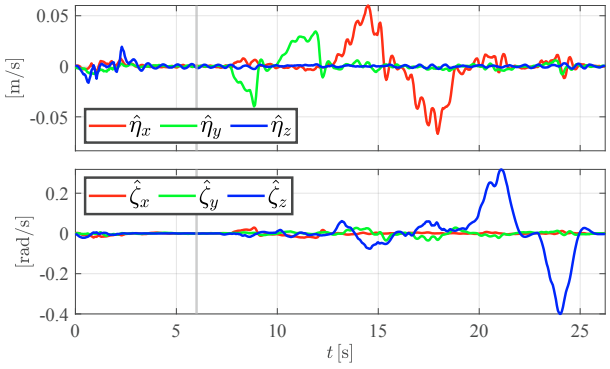

Fig. 6. Second experiment: Estimated velocity in the second task execution.

λ, when the EKF estimation is used (red line) the gain of the closed loop component $\boldsymbol{v}_{cl}$ is very low since the error is low being $\hat{\boldsymbol{v}}_d$ an accurate estimation of $\boldsymbol{v}_d$. On the other hand, when the EKF estimation is not used (blue line) the control gain increases during the object motion and the camera is controlled exclusively by the closed loop component $\boldsymbol{v}_{cl}$ and the forward action $\boldsymbol{v}_f$. Figure 5 and Fig. 6 show the relative velocity estimated by the EKF during the two task executions. Note that during the approach phase in both executions, the EKF estimates a small relative velocity even if the object is fixed. This is due to the fact that the robot is not a perfect velocity generator and its tracking error can be seen as a disturbance equivalent to a relative object/camera velocity. Also in this phase the error is lower when the EKF estimation is used in the control action.

The video of the experiments is available at `https://youtu.be/beNiXAdvrnU`.

8. CONCLUSION

In this paper we have proposed a 3D feature-based sampled-data visual control scheme which takes into account the sampling rate of the visual perception pipeline. To avoid anomalous behaviours due to local minima, we have suitably designed a reference feature trajectory to the control system in order to constraint the feature dynamics close to the desired equilibrium point. To improve the performance of the tracking module in terms of robustness against lighting conditions and partial occlusions, we adopted a ZNCC-based template matching algorithm. The current features are generated by the homography, provided by the visual tracker, applied to the features selected on the target image by the user. As the target object moves, e.g., during an handover maneuver, and this can affect the performance of the visual tracking module, the object motion has been estimated by means of an EKF. Such an estimation has been added to the camera velocity command so that the camera/object relative motion is determined by the reference feature trajectory only. Stability of the closed loop system is ensured by the visual controller, designed by resorting to the Lyapunov direct method, and explicitly taking into account the robot joint velocity limits.

REFERENCES

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3), 221–255.

Benhimane, S. and Malis, E. (2007). Homography-based 2d visual tracking and servoing. *The International Journal of Robotics Research*, 26(7), 661–676.

Cervera, E., del Pobil, A.P., Berry, F., and Martinet, P. (2003). Improving image-based visual servoing with three-dimensional features. *The International Journal of Robotics Research*, 22(10-11), 821–839.

Corke, P. (2017). *Robotics, Vision and Control*. Springer Nature, Cham, CH.

Costanzo, M., De Maria, G., Natale, C., and Russo, A. (2022). Stability and convergence analysis of 3d feature-based visual servoing. *IEEE Robotics and Automation Letters*, 7(4), 12022–12029.

Dai, J.S. (2015). Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92, 144–152.

Dame, A. and Marchand, E. (2010). Accurate real-time tracking using mutual information. In *2010 IEEE Int. Symp. on Mixed and Augmented Reality*, 47–56.

Di Stefano, L., Mattoccia, S., and Tombari, F. (2005). Zncc-based template matching using bounded partial correlation. *Pattern Recognition Letters*, 26(14), 2129–2134.

Liberzon, D. and Morse, A. (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5), 59–70.

Lucas, B.D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 674–679. San Francisco, CA, USA.

Marchand, E., Spindler, F., and Chaumette, F. (2005). ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics Automation Magazine*, 12(4), 40–52.