A new method for solving 6D Image-Based Visual Servoing with Virtual Composite camera model

Emmanuel Dean-León and Gordon Cheng

***Abstract*— This paper presents a new formulation to the open problem of 6D Image-Based Visual Servoing (IBVS). The main goal is to control the *pose* of an object using visual information from stereo cameras. In this article we introduce a novel image feature representation based on virtual orthogonal cameras to map 6D Cartesian poses to 6D visual poses defined in a *Virtual Visual space* (Image space). This new model is used to compute a full-rank *Image Jacobian matrix* (J_{img}), which overcomes several common problems exhibited by the classical image Jacobians, e.g., Image space singularities and local minima. This Jacobian is a fundamental key for a Image-Based control design, where a stereo camera system can be used to drive a robot manipulator. The properties of the proposed visual model are validated analytically, in simulation and in a real robot.**

I. Introduction

Visual Servoing Control (VSC) is an approach to control the motion of a robot manipulator using visual feedback from a vision system. This has been one of the most active topics in robotics since the early 1990s [1]. The vision system can be mounted directly on a robot end-effector (eye-in-hand configuration) or fixed in the work space (fixed-camera configuration). Additionally, visual servoing approaches differ in the way in which error functions are defined. In *Image-Based Visual Servoing* (IBVS) the error function is defined directly in terms of image features. In *Position-Based Visual Servoing* (PBVS) the error function, which is specified in the *Task space* (e.g. Cartesian coordinates), is obtained from the visual information [2]. The conclusion drawn in many of the previous works, e.g., [1], is that IBVS is more favorable than the PBVS method, since it has low sensitivity to camera calibration errors.

This work is based on the concepts of image-based visual servoing and attempt to address some of the most common problems affecting conventional approaches by introducing a new visual feature mapping based on the composite camera model [3]. As pointed out in [1], convergence and stability problems may sometimes occur in IBVS. Local minima in the trajectories and singularities in the Image Jacobian (also known as Interaction Matrix) can severely affect the visual servoing task. In image-based control approach, the ideal case is to find a particular visual feature where the interaction matrix has neither local minima nor singularities, and where the exponential decrease of the corresponding error function implies a smooth 3D trajectory for the controlled object (e.g. the robot end-effector). During the last decade, several authors have worked on solving these problems. We will describe some relevant approaches used in IBVS and discuss their properties.

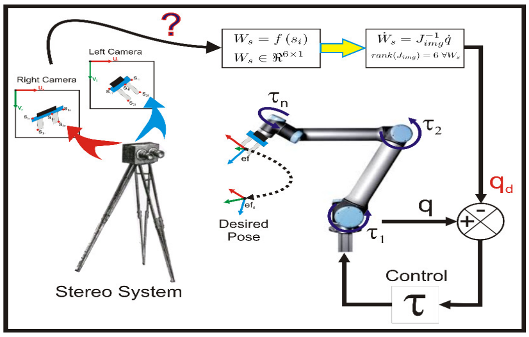

Fig. 1. A classical problem of visual servoing is to generate a full-rank mapping between the visual feature space $s \in \mathbb{R}^{2m\times1}$ and the joint velocity space $\dot{q} \in \mathbb{R}^{n\times1}$ to design dynamic controls for robots $\tau \in \mathbb{R}^{n\times1}[Nm]$.

A. Related work

An IBVS usually employs the *image Jacobian matrix* (J_{img}) to relate velocities of an object or a camera in the Task space (v) to the feature parameter velocities in the feature (image) space $\dot{s}$. A full and comprehensive survey on Visual Servoing and image Jacobian definitions can be found in [1], [2], [4] and more recently in [5]. In general, the classical image Jacobian is defined using a set of image feature measurements, usually denoted by s, and it describes how image features change when the object or camera pose changes, i.e. $\dot{s} = J_{img}v$. This is connected with Dynamic Visual Servoing where the main goal is to control the required robot's dynamical behavior to achieve some desired value of image feature parameters. This implies estimating the image Jacobian to map the image feature velocities $\dot{s}$ into a meaningful state variable required for the control law, usually the generalized joint velocities $\dot{q}$, see Fig. 1.

In general, the image Jacobian can be computed using direct depth information (depth-dependent Jacobian) [6],[7], or by approximation via on-line estimation of depth of the features (depth-estimation Jacobian) [2], [5], [8], or using depth-independent image Jacobian matrix [9],[10]. Additionally, many papers directly estimate on-line the complete image Jacobian in different ways, [11],[12]. However, all these methods use redundant image point coordinates to define (as a general rule) a non-square image Jacobian, which is a differentiable mapping from $SE(3)$ to $s \in \mathbb{R}^{2m}$ (with m as the number of feature points). Then, a generalized inverse of the image Jacobian needs to be computed, which leads to well-known problems such as the Image space singularities

Authors Affiliation: Technischen Universität München, Insitute for Cognitive Systems, Fakultät für Elektrotechnik und Informationstechnik.
Email:{dean,gordon}@tum.de

and local minima. In order to obtain a full-rank Image Jacobian, [13] proposed an approach where the definition of the features is based on a combination of both IBVS and PBVS approaches, attempting to incorporate the advantages of each method. This method requires the knowledge of the 3D model of the object and its performance is guaranteed in the absence of high calibration or model errors. In our case, the structure of the proposed Jacobian do not require information of the object's model and it can be used to design adaptive controllers capable to handle uncertainties in the camera parameters, similar to [14]. The work presented in [15] introduces in a similar fashion a stereo camera model based on a virtual composite camera system. However, that approach is limited to control only 3D positions, and the image Jacobian is only suitable for 3 DOF robots.

In this paper, we extend the visual features to control 6D visual poses in a Virtual Visual space (a 3D space defined in pixels), see Fig. 1. This visual *pose* is composed of 3D visual linear motions and 3D visual angular motions. Therefore, this work proposes a general solution to control the position and orientation of a robot end-effector using visual information from stereo cameras. The key contributions of this paper can be summarized into three aspects: **i)** We specify a new *Virtual Visual space* (pixel), where a 6D visual pose vector can be defined and used for IBVS. **ii)** We obtain a new full-rank *image Jacobian* which maps velocities from the Task space to the 3D Virtual Visual space. **iii)** In order to evaluate the proposed visual Jacobian models, experiments have been conducted in simulation using real camera parameters and in a real robot, where our approach is compared with a standard IBVS method.

B. Organization

This paper is organized as follows, in Section II we highlight the problems in classical image-base visual servoing approaches and state the core issues which we tackle with this work. In Section III we introduce a new *3D Camera Model* and describe how it is used to construct a *Virtual Visual space*. This model is used to define a full-rank *6D Visual Jacobian*, which will be used in the Section V to design a robust dynamic control for robot manipulators. In Section IV we present a comparative analysis in simulation between our method and the standard IBVS. Finally, Section VI draws the conclusions and future work.

II. Problem Formulation

Visual servoing schemes rely on the relationship [2]: $\dot{s} = L_s v_o$, in which s is a set of geometrical features whose time derivative $\dot{s}$ is linearly related to the spatial velocity v_o of an object through the interaction matrix L_s. Using this relationship, control schemes are designed to minimize the error e between the current value of the visual feature s and its desired value $s_d : e = s - s_d$.

A. Classical Image-Based Visual Servoing

Traditional image-based control schemes use the image-plane coordinates of a set of points to define the set s [2]. More precisely, if we have a set of points $p_{0_i} = [x_{0_i}, y_{0_i}, z_{0_i}]^T$, for $i = 1, 2, ..., m$, rigidly attached to an object whose pose is represented by the vector $W_0 = [X_0, \theta_0]^T = [x_0, y_0, z_0, \alpha_{x_0}, \beta_{y_0}, \gamma_{z_0}]^T \in \mathbb{R}^{6\times 1}$, then the visual feature measurements of all p_{0_i} $(m > 3)$ are represented as $s = [x_1, y_1, ..., x_m, y_m]^T \in \mathbb{R}^{2m\times 1}$. The relation between $\dot{s}$ and $\dot{W}_0$ is given by $\dot{s} = J_x \dot{W}_0$, where $J_x = [L_{x_1}, ..., L_{x_m}]^T \in \mathbb{R}^{2m\times 6}$ is known as the image Jacobian, and L_{x_i} is given by[1]:

$$L_{x_i} = \begin{bmatrix} \frac{1}{z_{0_i}} & 0 & -\frac{x_i}{z_{0_i}} & -x_i y_i & (1+x_i^2) & -x_i \\ 0 & \frac{1}{z_{0_i}} & -\frac{y_i}{z_{0_i}} & (1+y_i^2) & x_i y_i & y_i \end{bmatrix} \tag{1}$$

Where $X_i = [x_i, y_i]^T = [\frac{x_{0_i}}{z_{0_i}}, \frac{y_{0_i}}{z_{0_i}}]^T = [\frac{u_i - c_{x_i}}{f_{x_i}}, \frac{v_i - c_{y_i}}{f_{y_i}}]^T$, with $C_i = [c_{x_i}, c_{y_i}]^T$ and $F_i = [f_{x_i}, f_{y_i}]^T$ as the principal point and the scaling factors, respectively.

B. The problem of Classical IBVS

If we consider ΔW_0 as the control variable, then we need to compute the inverse mapping of $\dot{s}$ as

$$\Delta W_0 = J_x^+ \Delta s, \tag{2}$$

where $\Delta *$ is an error function defined in the space $*$, $J_x^+ \in \mathbb{R}^{6\times 2m}$ is chosen as the Moore-Penrose pseudo-inverse of J_x, which leads to the two characteristic problems of the IBVS method: the feature (image) space singularities and local minima. For most IBVS approaches we have $2m > 6$. In this case, the image Jacobian is singular when $rank(J_x) < 6$, while the visual feature local minima is defined as the set of image locations $\Omega_s = \{s | \Delta s \neq 0, \Delta W_0 = 0, \forall s \in \mathbb{R}^{2m\times 1}\}$ when using redundant image features, i.e. $\Delta s \in N(J_x^+)$. Examples of the problems generated by the local minima conditions are illustrated in [2] and [16]. Another drawback of these approaches is the highly-coupled system presented by (2), where the linear velocities and angular velocities can not be controlled independently from each other, leading to complex issues which need to be addressed during control design.

C. Contribution of this Work

In this work, we get a step further towards a general solution for the problem of the IBVS, by introducing a mapping from the classical image features s to a new visual representation defined as $W_s = [X_s, \theta_s]^T = [x_s, y_s, z_s, \alpha_s, \beta_s, \gamma_s]^T \in \mathbb{R}^{6\times 1}$. In this case, W_s is a 6D visual pose vector defined in a 3D Image space (we call this space the *Virtual Visual space*). This visual pose is measured in pixels and is composed of 3D visual positions and 3D visual orientations.

The visual representation W_s is related with the feature points vector s as $W_s = M(s)$, where M is, in general, a nonlinear map[2] $M : \mathbb{R}^{2p\times 1} \longrightarrow \mathbb{R}^{6\times 1}$. In our case, W_s is a minimization of the space s without the need to compute an on-line LSM of the system $|J_x \dot{W}_0 - \dot{s}|$ as discussed in [17].

[1] For the case when the object is moving and the camera is fixed.

[2] This depends on the definition of the visual feature and its relationship with the task space.

The above definition yields a new mapping in the form

$$\dot{W}_s = \underbrace{\frac{\partial M(s)}{\partial s}}_{J_i} \dot{s} = \underbrace{J_i J_x}_{J_{img}} \dot{W}_0 \tag{3}$$

The advantage of this intermediate mapping is that a full-rank *image Jacobian matrix* ($J_{img} \in \mathbb{R}^{6\times 6}$) can be obtained, i.e., if some specific conditions are met, then the Image space singularities and local minima issues can be avoided. In the following section, we will specify these conditions.

Moreover, the aim of all vision-based control schemes is to minimize an error $e(t)$, which is typically defined by $e(t) = s - s_d$. In our case, the error function (ΔW_s) is defined in the *Virtual Visual space* generated from two stereo images.

This is the core design of our approach, and all that remains is to fill in the details. For example, how should W_s be chosen to construct the *Virtual Visual space*? What is the form of J_{img}? How the properties of J_{img} impact in the performance of the control approach? These questions are addressed in the remainder of the article.

III. Virtual Visual Space

This section shows how we construct the *Virtual Visual space* using a stereo vision system. It also explains in detail how to obtain a full-rank *image Jacobian matrix* ($J_{img} \in \mathbb{R}^{6\times 6}$). In the remainder of this paper, we will use the notation C^i_j to represent a mapping from frame i to frame j and p_{h_k} to define a point k in the coordinate frame h. The key idea of the 3D visual camera model is to combine the stereo camera model with a virtual composite camera model. Figure 2 a) depicts our new visual camera model and the corresponding image projections.

A. Image Jacobian for 3D visual linear velocities

We first generate a full-rank matrix ($J_{img_v} \in \mathbb{R}^{3\times 3}$), which maps 3D linear velocities $\dot{X}_0 \in \mathbb{R}^{3\times 1}$ (m/s) to 3D visual linear velocities $\dot{X}_s \in \mathbb{R}^{3\times 1}$ $(pixel/s)$, see Figure 2.

This new 3D visual model can be computed in 2 main steps. First, given a standard stereo system ($C^l_0 - C^r_l$), compute the Homography between the right camera C^r_0 and a *virtual right camera* $C^{r_v}_0$. This *virtual* camera must be oriented such as its image plane is orthogonal to the left camera, see Figure 2. From these two cameras, visual information of a rigid object can be obtained x_{l_i} and $x_{r_{v_i}}$, for $i = 1, 2, ..., m$. Second, use this visual information as a projection to form a Composite Camera Model. This projection is a crucial step, since it modifies the dimension of the mapping from two 2D-visual feature measurements of all m points, $s = [u_{l_1}, v_{l_1}, u_{r_{v_1}}, v_{r_{v_1}}, ..., u_{l_m}, v_{l_m}, u_{r_{vm}}, v_{r_{vm}}]^T \in \mathbb{R}^{4m\times 1}$, to a single 3D visual vector $X_s \in \mathbb{R}^{3\times 1}$ defined in a *Virtual Visual space*. Since s represents the position $t^4_0 \in \mathbb{R}^{3\times 1}$ of the rigid object in the image feature space (see Fig. 2), the maximum number of independent elements of s is 3. Therefore, if $s \in \mathbb{R}^{4m\times 1}$ (as is commonly defined in the classical methods for the stereo arrangements) there will be $(4m-3)$ linearly dependent elements in s. In this work, we propose to use the *homography* between the cameras ($C^l_0 - C^r_l$) to define a virtual projection that reduces the dimension of s and generates 3 linearly independent elements to compute a full-rank image Jacobian (J_{img_v}). The following sub-sections are devoted to explain each of these steps in detail.

1) ***Stereo Vision Model:*** The stereo vision system is composed of two cameras (left C^l_0 and right C^r_0) which are rigidly attached to a common frame 0_w (Fig. 2). This reference frame is defined by the user in any fashion, however, it is assumed that the position of 0_w is in the intersection of the optical-axes of left camera z^l_0 and *virtual* right camera $z^{r_v}_0$, and its orientation is exactly the same as C^l_0 on the beads to simplify further steps. Using this position and orientation, we define the *Projection Matrices* P^0_l and $P^0_{r_v} \in \mathbb{R}^{3\times 4}$ for the left camera and the *virtual* right camera as $P^0_l = K_l \begin{bmatrix} I_{3\times 3} & t^0_l \end{bmatrix}$ and $P^0_{r_v} = K_r \begin{bmatrix} R^0_{r_v} & t^0_{r_v} \end{bmatrix}$. K_l and K_r are the intrinsic camera matrices of the left camera and *virtual* right camera[3], respectively. Furthermore, by design $t^0_l = [0, 0, z_l]^T$, $t^0_{r_v} = [0, 0, z_{r_v}]^T$ and $R^0_{r_v} = \left(R^{r_v}_l\right)^T \left(R^l_0\right)^T = R^l_{r_v}$. These *Projection Matrices* can be used to define the *virtual* composite camera model, but first we need to project the real visual features of the right camera x_{r_i} to a visual features on the *virtual* right camera $x_{r_{v_i}}$. This is achieved exploiting the homography property, similar to stereo rectification, but in this case we will generate an orthogonal arrangement.

a) ***Homography between C^r_0 and $C^{r_v}_0$:*** A homography H is a projection of world points between two image planes. This mapping is only valid when the world points are constrained to a common plane or when the two cameras share the same camera center, which is our case. Under these conditions, $H^r_{r_v}$ can be computed as $H^r_{r_v} = K_l R^r_{r_v} K_r^{-1}$. In this case we need to rotate the right camera such as its image plane is orthogonal to C^l_0. Therefore, the relative orientation of C^r_0 with respect C^l_0 must be considered. Therefore, $R^r_{r_v} = R_{y_l}(\pi/2) R^r_l$, where, $R_{y_l}(*)$ represents a rotation matrix around $y^l_0 - axis$. Then, the visual points in $C^{r_v}_0$ are obtained as $x_{hr_{v_i}} = H^r_{r_{v_i}} x_{hr_i}$, where x_{h*_i} represents the homogeneous vector of the image point $x_{*_i} = [u_{*_i}, v_{*_i}]^T$.

2) ***Virtual Composite Camera Model:*** In this work we propose a Composite Camera Model as a geometrical minimization method for the feature space s. The advantage of the Orthogonal Cameras configuration is that it presents the image position as 3 orthogonal signals. These features can be used in the visual servoing instead of the classical features extracted from the stereo cameras to generate a full-rank image Jacobian. However, this configuration is limited, compared with the stereo camera configuration, because it requires a complex and accurate arrangement between the cameras. In this work we propose to combine the benefits of the two configurations by mapping the projections of the standard stereo configuration into a *virtual composite camera arrangement*. The measurements obtained in this virtual sensors will generate a 3D visual position vector.

a) ***Virtual Composite Camera Model Generation:*** The matrix P^0_l projects a point $p_{0_i} = [x_{0_i}, y_{0_i}, z_{0_i}]^T$ into the left

[3] We assumed that the *virtual* right camera has the same intrinsic parameters as the right camera. These parameters can be computed off-line.

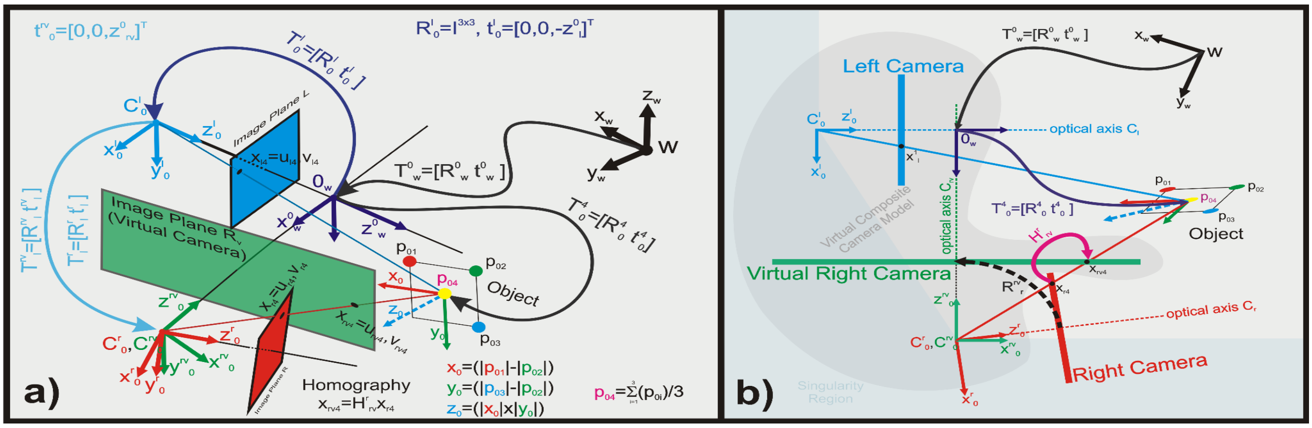

Fig. 2. Image Projections: a) The figure depicts the coordinate frames used to obtain a general 3D *Virtual Visual model*. It shows two physical cameras (*Left* and *Right*) and their coordinate frames (C_0^l and C_0^r, respectively). These two cameras are rigidly linked to a reference frame 0_w, which is defined with respect to a world coordinate frame W. It is also illustrated a *Virtual Camera* denoted by $C_0^{r_v}$, which shares the same position as C_0^r but it is orthogonal to the left camera C_0^l. The pose of a rigid object is defined by the coordinate frame $\overline{p_{w1}p_{w2}p_{w3}p_{w4}}$. Each object point p_{w_i} is referenced to 0_w as p_{0_i} and these new points are projected to the left and right cameras as $x_{l_i}=[u_{l_i},v_{l_i}]^T$ and $x_{r_i}=[u_{r_i},v_{r_i}]^T$, without lost of generality, we assume that p_{0_4} represents the position of the object in 0_w and $\overline{p_{0_1}p_{0_2}p_{0_3}}$ represents its orientation. The projection of these object points in the *virtual camera* $C_0^{r_v}$ is obtained with a homography with C_0^r, i.e. $x_{hr_{v_i}}=H^r_{r_{v_i}}x_{hr_i}$. In b) it is shown the same arrangement from a top view. Here it is possible to observe that the relation between C_0^r and $C_0^{r_v}$ is given by the rotation matrix $R^r_{r_v}$. This rotation must be computed based on the relative orientation of C_0^r with respect to C_0^l. The *Virtual Composite* camera system is highlighted with the light-gray area and is composed of the left camera and the *virtual* right camera, C_0^l and $C_0^{r_v}$ respectively.

camera image plane as

$$x_{l_i}=\left[u_{l_i},v_{l_i}\right]^T=\left[\frac{f_{x_l}x_{0_i}}{(z_{0_i}+z_l)},\frac{f_{y_l}x_{0_i}}{(z_{0_i}+z_l)}\right]^T+\left[c_{x_l},c_{y_l}\right]^T \quad (4)$$

In the same form $P^0_{r_v}$ generates the image projection of p_{0_i} in the *virtual* right camera

$$x_{r_{v_i}}=\left[u_{r_{v_i}},v_{r_{v_i}}\right]^T=\left[\frac{f_{x_{r_v}}z_{0_i}}{(-x_{0_i}+z_{r_v})},\frac{f_{y_{r_v}}y_{0_i}}{(-x_{0_i}+z_{r_v})}\right]^T+\left[c_{x_{r_v}},c_{y_{r_v}}\right]^T \quad (5)$$

These two models can be fused to generate a *Composite Camera Model* which will represent the *3D visual space*. This is possible because by design the component $u_{r_{v_i}}$ measures a quasi-orthogonal signal to the ones captured by x_{l_i}. Then, grouping these terms, we can define the *Virtual Composite Camera Model* as

$$X_{s_i}=[x_{s_i},y_{s_i},z_{s_i}]^T=\left[u_{l_i},v_{l_i},u_{r_{v_i}}\right]^T=FX_{n_i}+C \quad (6)$$

where $F=diag(f_{x_l},f_{y_l},f_{x_r})$, $C=\left[c_{x_l},c_{y_l},c_{x_r}\right]^T$ and

$$X_{n_i}=\left[\frac{x_{0_i}}{(z_{0_i}+z_l)},\frac{y_{0_i}}{(z_{0_i}+z_l)},\frac{z_{0_i}}{(-x_{0_i}+z_r)}\right]^T \quad (7)$$

A velocity relationship can be obtained with the time derivative of (6) as follows:

$$\dot{X}_{s_i}=\overbrace{FJ_{v_i}}^{J^0_{imgv}}\dot{X}_{0_i}=J^0_{img_v}\dot{X}_{0_i} \quad (8)$$

where the Jacobian matrix $J_{v_i}\in\mathbb{R}^{3\times3}$ is defined as

$$J_{v_i}=\begin{bmatrix}\frac{1}{(z_{0_i}+z_l)} & 0 & -\frac{x_{0_i}}{(z_{0_i}+z_l)^2}\\ 0 & \frac{1}{(z_{0_i}+z_l^0)} & -\frac{y_{0_i}}{(z_{0_i}+z_l)^2}\\ \frac{z_{0_i}}{(x_{0_i}-z_r)^2} & 0 & -\frac{1}{(x_{0_i}-z_r)}\end{bmatrix} \quad (9)$$

This *composite image Jacobian* $J^0_{img_v}$ represents the mapping from linear velocities defined in the reference frame 0_w to velocities (pixels/s) in the 3D visual space.

B. Image Jacobian for 3D visual angular velocities

This Jacobian provides a mapping between angular velocities in the task space (e.g. 0_w) and angular velocities in the *virtual* visual space. Specifically, we need to analyze the effects of a velocity ω_0 in the visual space, i.e. ω_s. To this aim, we can describe each of these angular velocities as the induced linear velocity on a point not centered at the axis of rotation in each of the two spaces, e.g. $p_{0_i}, i=1,2,3$. This can be done using the canonical basis of the Cartesian space, namely $e_{x_0}=[1,0,0]^T$, $e_{y_0}=[0,1,0]^T$, $e_{z_0}=[0,0,1]^T$ and its origin $e_{o_0}=[0,0,0]^T$. If we project each of this vectors to the *virtual* visual space we obtain (see (6)), $e_{x_s}=[p_{x_l}+f_{x_l}/z_l,p_{y_l},p_{x_r}]^T$, $e_{y_s}=[p_{x_l},p_{y_l}+f_{y_l}/z_l,p_{x_r}]^T$, $e_{z_s}=[p_{x_l},p_{y_l},p_{x_r}+f_{x_r}/z_r]^T$ and $e_{o_s}=[p_{x_l},p_{y_l},p_{x_r}]^T$. We are interested in angular velocities at the origin of each coordinate frame, therefore, we must compute the radius of rotation for each vector. The radius is obtained as $\bar{e}_*=e_*-e_o$ for each axis. Then, for the Cartesian space we have, $\bar{e}_{x_0}=e_{x_0}$, $\bar{e}_{y_0}=e_{y_0}$, $\bar{e}_{z_0}=e_{z_0}$. For the visual space we have $\bar{e}_{x_s}=[f_{x_l}/z_l,0,0]^T$, $\bar{e}_{y_s}=[0,f_{y_l}/z_l,0]^T$, $\bar{e}_{z_s}=[0,0,f_{x_r}/z_r]^T$.

The linear velocity generated by a general angular velocity is obtained with: $v=w\times r=S(w)r$, where $S(*)$ represents the skew-symmetric matrix of the vector $(*)$. From (8) we can generate a relation between each linear velocity represented in 0_w and its corresponding velocity in the visual space as[4]

$$V_{\{x,y,z\}_s}=F\;J_v|_{\bar{e}_{\{x,y,z\}_0}}\;V_{\{x,y,z\}_0} \quad (10)$$

[4] The expression $J_v|_{\bar{e}_{\{x,y,z\}_0}}$ means J_v evaluated at each vector $\bar{e}_{x_0}$, $\bar{e}_{y_0}$, $\bar{e}_{z_0}$.

Substituting the linear velocity in each space in (10), we obtain a correlation between angular velocities ω_0 and ω_s

$$S\left(\bar{e}_{\{x,y,z\}_s}\right)^T \omega_s = F\, J_v|_{\bar{e}_{\{x,y,z\}_0}} S\left(\bar{e}_{\{x,y,z\}_0}\right)^T \omega_0 \qquad (11)$$

The expression in (11) generates a set of three equations that need to be solved to define the angular velocity mapping. It is clear that the solution will not be unique, however, a solution that fits the real system constraints can be defined. In this case, a system constraint will be that real physical cameras only produce positive values for the visual position $x^i_{l,r} \in \mathbb{R}^{2\times 1}$. Therefore, we can define the following relation

$$\omega_s = F_s^{-1} F_0 \omega_0 = J^0_{img_\omega} \omega_0 \qquad (12)$$

where, $F_0 = diag\left(\frac{f_{x_r}}{z_r}, \frac{f_{x_r}}{z_r}, \frac{f_{y_l}}{z_l}\right)$ and $F_s = diag\left(\frac{f_{y_l}}{(z_l+1)}, \frac{f_{x_l}}{(z_r+1)}, \frac{f_{x_l}}{z_l}\right)$.

The geometrical meaning of (12) is that the angular velocities exhibit the same direction in both spaces, but they are scaled by a non-homogeneous factor. This result can be used to design control approaches, because, despite the non-homogeneous scaling, this mapping is continuous.

C. Visual Jacobian

In the previous sections, we have defined the mappings for the 3D visual velocities (linear and angular) separately as *linear velocity image Jacobian J_{img_v}* and *angular velocity image Jacobian J_{img_ω}*. Combining equation (8) and (12) we have the full expression that can be used for control design.

$$\dot{W}_s = \begin{bmatrix} \dot{X}_s \\ \omega_s \end{bmatrix} = \begin{bmatrix} J^0_{img_v} & 0 \\ 0 & J^0_{img_\omega} \end{bmatrix} \begin{bmatrix} \dot{X}_0 \\ \omega_0 \end{bmatrix} = J^0_{img} \dot{W}_0 \qquad (13)$$

where the Jacobian $J^0_{img} \in \mathbb{R}^{6\times 6}$ is defined as the ***Visual Jacobian***. It is important to notice that the linear and angular motions are decoupled, and this is precisely the behaviour of the system that we want to model. If we observe Fig. 2 we can notice that the position (Cartesian and visual) of point p_{0_4} is not modified if we rotate the points p_{0_i}, $i = 1,2,3$ around p_{0_4}. We can say that equation (13) represents the pose of an object whose position is defined by p_{0_4} and orientation defined by the unit vectors $x_0 = |p_{0_1} - p_{0_2}|, y_0 = |p_{0_3} - p_{0_2}|, z_0 = x_0 \times y_0$. This decoupling between linear and angular velocities is an important feature of our model.

***Remark 1: Singularity-free** J_{img}.* From (13), we can see that $det(J_{img}) = det(J_{img_v}) det(J_{img_\omega})$, therefore the set of singular configurations of J_{img} is given by the equations $det(J_{img_v}) = 0$ and $det(J_{img_\omega}) = 0$. From (8), we can observe that singularities of J_{img_v} are defined by $det(J_{v_i}) = 0$. This equation is satisfied when i) $z_{0_i} = -z_l$, ii) $z_{r_v} = 0$, iii) $x_{0_i} = z_{0_i} + z_l$ and vi) $x_{0_i} = z_{r_v}$. The geometric meaning of each equation can be used to demonstrate the singularity-free visual Jacobian: i) the tracked object is located at the center of the left camera, ii) the center of the virtual right camera is located at the left camera's optical-axis, iii) the object is at the stereo system base line and finally, iv) the object position lies at the virtual right camera plane. The last constraint implies that the usable work space of the right camera can be reduced, depending on the configuration of the stereo arrangement. This singularity can be avoided if the stereo system is located such as the workspace is mainly covered by the left camera. On the other hand, from (12) the singularities are i) $z_l = 0$, ii) $z_{r_v} = 0$, iii) $z_l = -1$ and iv) $z_{r_v} = -1$. In the same form, any of these situations are feasible in a real physical scenario, i) and ii) by design 0_w is located neither at C^l_0 nor at $C^{r_v}_0$, iii) and iv) mean that 0_w is either behind C^l_0 or C^r_0, which is not possible. In Figure 2b) is shown the singularity area of J_{img} (light-blue area).

IV. Validation on Simulation

A. Convergence without local minima

In order to numerically validate the properties of the *Visual Jacobian* we simulate a simple kinematic control of an object using a) standard IBVS approach [2] and b) the new approach presented in this work. The task is to drive a square from a initial pose ($position_w = [0.2, 0.5, 1.0]^T$[m], *euler Angles* $= [0,0,0]^T$[deg])[5] to a desired pose ($position_w = [0.6, 0.7, 1.5]^T$[m], *euler Angles* $= [-140, -30, -20]^T$[deg]). The 4 corners were used as a visual feature for both approaches, i.e. p_{0_1}, p_{0_2}, p_{0_3}, p_{0_4}. Where p_{0_4} (center) was used as the object's position. The projection of these points produce two sets of data $s = [u_{l_1}, v_{l_1}, u_{r_1}, v_{r_1}, ..., v_{r_4}]^T \in \mathbb{R}^{16\times 1}$ used in the standard approach, and $W_s = [X_s, \omega_s] \in \mathbb{R}^{6\times 1}$, used in our approach.

1) Standard IBVS: A kinematic P-control can be implemented as $u = -\lambda \widehat{L_e^+} \Delta s$ [2], where $\lambda_+ = \lambda_+^T \in \mathbb{R}^{6\times 6}$, $\Delta s = s - sd \in \mathbb{R}^{16\times 1}$ and $\widehat{L_e^+} = \frac{L_x + L_{x_d}}{2} \in \mathbb{R}^{6\times 16}$ (see (1)).

2) Virtual Composite Cameras IBVS: In this case we implement also a simple kinematic P-control based on (13), i.e. $u = -\lambda (J^0_{img})^{-1} \Delta W_s$, with $\Delta W_s = B[\Delta X_s, \Delta\alpha_s]^T$. B is the *ZYZ-Euler Angle Transformation* and is defined as

$$B = \begin{bmatrix} I & 0 \\ 0 & B(s) \end{bmatrix}, B(s) = \begin{bmatrix} \cos(\varphi_s)\sin(\theta_s) & -\sin(\varphi_s) & 0 \\ \sin(\varphi_s)\sin(\theta_s) & \cos(\varphi_s) & 0 \\ \cos(\theta_s) & 0 & 1 \end{bmatrix} \qquad (14)$$

a) Visual Position Errors: The position error is defined as $\Delta X_s = X_{s_4} - X_{s_{d_4}}$, where X_{s_4} and $X_{s_{d_4}}$ are computed using (6) for p_{0_4} and $p_{0_{d_4}}$.

b) Visual Orientation Errors: The visual orientation is represented as visual Euler angles α_s, which are computed in the following form:

Compute Visual Rotation: Using the projection X_{s_k} of points p_{0_k}, with $k = 1,2,3$, we compute a rotation matrix R_s in the *Virtual Visual* space.

$$R_s = \left[\|\Delta X_{s_1}\|, -\left\|\Delta X_{s_3}\right\|, \|\Delta X_{s_1}\| \times \left(-\left\|\Delta X_{s_3}\right\|\right)\right] \qquad (15)$$

$$\Delta X_{s_j} = X_{s_j} - X_{s_2} \qquad (16)$$

Compute Visual Euler angles: The rotation matrix R_s is represented as Euler angles to generate α_s and α_{s_d}.

Figure 3 shows the results using the standard method (see IV-A.1), and Figure 4 illustrates the results obtained with our approach (see IV-A.2). From the results, it is clear that

[5] We use the Euler Angles (ZYZ) to represent the orientation of the box.

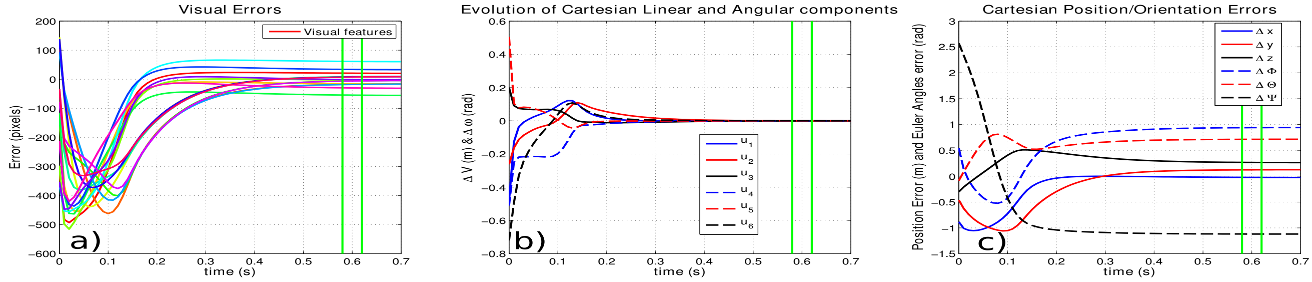

Fig. 3. This figure shows the behavior of a box controlled using the standard IBVS method. In a) it is shown the visual errors Δs between the current visual position of the box and its desired visual position. It can be noticed that the visual errors converge to a local minima, in b) the control input $u = -\lambda \widehat{L_e^+} \Delta s$ is depicted and in c) the Cartesian pose errors are shown. It can be seen that in the time frame $t = [0.58, 0.62]s$ (green-vertical lines) the box is attracted to a local minima, since $u \approxeq 0^{6\times 1}$ (**b**), even when $\Delta s \neq 0^{16\times 1}$ (**c**). This produces a steady state error in the Cartesian space, i.e. local-minima (**f**).

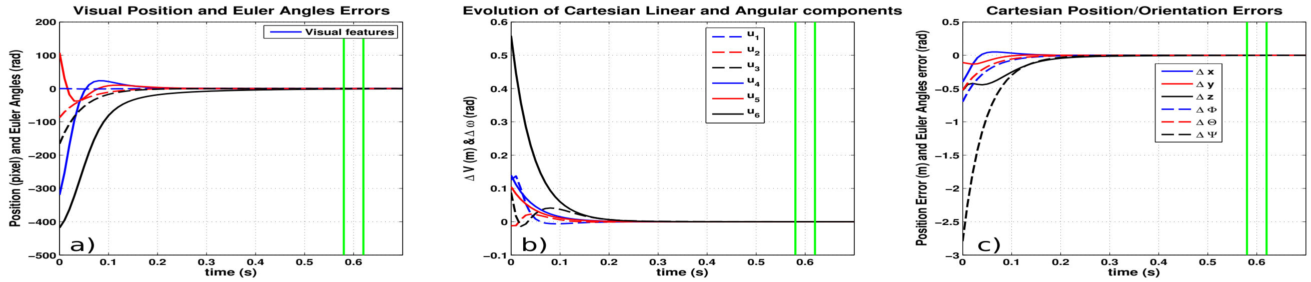

Fig. 4. The figure shows the behavior of the box using an IBVS based on our *Virtual Composite Model*. In: a) the visual errors ΔW_s b) the control input $u = -\lambda (J_{img}^0)^{-1} \Delta W_s$ and c) the Cartesian pose errors. In contrast with Figure 3, all the error and control signals converge to zero. In other words, the system reaches the desired state without local-minima.

the standard method suffers from local-minima, while our approach allows the convergence of errors in both spaces, *virtual visual* and Cartesian. This can be easily proved by analyzing the *Null* space for both methods.

V. Experimental Results

In order to validate the image Jacobian presented in this work, we used our robot TOM (Tactile Omni-directional Mobile Manipulator) depicted in Fig. 5. The robot is composed of 2 industrial robot arms (UR-5) mounted on a Omni-directional platform developed at our institute. An AR marker has been mounted on the right arm of the robot. The markers are tracked using the AruCo library which is based on OpenCV. Every marker provides 2D image features for 4 corner points. This visual points are used to compute the Visual Jacobian presented in this work. Both the visual stereo-tracking, the virtual composite model and the control have been implemented in ROS. The task is to track the pose of a target marker which is controlled by a user. We have added to the desired position an offset of $[118, 20, -100]$ $(pixels)$, which represent an offset of $[0, 0.17, 0]^T$ (m) with respect to the robot base, to avoid occlusions. The target orientation is used without offsets. The control approach is an extension of [14] for the 6D case and is defined in the next section.

A. Adaptive Image-based 6D Visual Servoing

In this section, we describe the design of an adaptive image-based dynamic control (second order sliding mode control). The proposed second order sliding mode control is chattering free. The control approach is defined as:

$$\tau = -K_d S_q + Y_r \Theta, \dot{\Theta} = -\Gamma {Y_r}^T S_q \qquad (17)$$

where $Y_r\Theta$ is the on-line estimation of the robot regressor, $K_d = {K_d}^T \in \mathbb{R}_+^{n\times n}$ and $\Gamma \in \mathbb{R}_+^{m\times m}$ are constant matrices and S_q is the *Joint Error Velocity* surface defined as:

$$S_q = \dot{q} - \dot{q}_r \qquad (18)$$

where the *Joint Velocity Nominal Reference* $\dot{q}_r = {J_s}^{-1} \dot{W}_{s_r}$ has been defined using (13) with respect to the robot base. This produces the new Jacobian $J_s = J_{img}^0 R J(q)$, with $R = diag([R_0^b; R_0^b]) \in \mathbb{R}^{6\times 6}$, where $R_0^b = R_0^l R_l^b \in SO(3)$ is the orientation of the robot base with respect to the frame O_w and $J(q)$ is the robot Jacobian. It's easy to prove that the singularities of J_s are the singularities of the visual Jacobian and the standard singularities of the robot. The 6D visual nominal reference $\dot{W}_{s_r}$ is given by

$$\dot{W}_{s_r} = \left(\dot{W}_{s_d} - K_p \Delta W_s + S_{s_d} - K_1 \int_{t_0}^{t} S_{s_\delta}(\zeta)\, d\zeta \; - K_2 \int_{t_0}^{t} sign\left(S_{s_\delta}(\zeta)\right) d\zeta \right) \qquad (19)$$

$$S_{s_\delta} = S_s - S_{s_d}, S_s = \left(\Delta \dot{W}_s + K_p \Delta W_s\right), S_{s_d} = S_s(t_0)\, e^{-\kappa t} \qquad (20)$$

where $\dot{W}_{s_d}$ is the desired visual velocity, $\Delta W_s = W_s - W_{s_d}$ is the visual position error, $\Delta \dot{W}_s$ is the visual velocity error, $K_p = {K_p}^T \in \mathbb{R}_+^{6\times 6}$ and $K_j = {K_j}^T \in \mathbb{R}_+^{6\times 6}$ (with $j = 1, 2$) and $S_{s\delta}$ is the *virtual visual error surface*. The visual position/velocity errors can be described as Euler angles in the same form as explained in Section IV-A.2. This adaptive on-line estimation together with the second order sliding mode in S_{s_δ} handle the uncertainties on the robot parameters. The passivity proof of this control can be found here [14]. The results obtained using the above control law with the robot depicted in Fig. 5 are shown in Figure 6. A video with more details about the experimental results can be found in: http://web.ics.ei.tum.de/~emmanuel/Dean/humanoids14.html

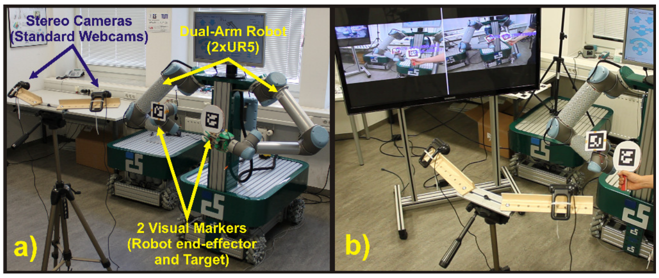

Fig. 5. This figure shows in a) the experimental setup used to validate our approach. It is composed of a dual-arm mobile manipulator, where 2 UR-5 robots are used as arms. The right arm is equipped with a AR marker mounted on its end-effector. The stereo-camera setup comprises two standard webcams logitech pro9000 mounted on a tripod. The system has been calibrated with respect to the body of the robot. In b) is depicted a snapshot of the task, where the target marker is moved by the user. The goal is to visually track the pose of the target marker with the robot arm.

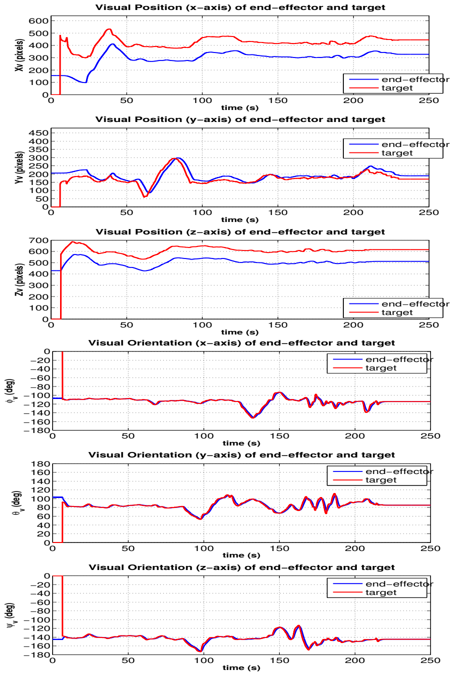

Fig. 6. This figure shows the visual tracking of the target using the robot end-effector. The visual position and the visual Euler angles of the target and the end-effector are shown. It can be noticed that the robot tracks the position of the target with a user-defined offset which has been added to avoid occlusions. The orientations are tracked without adding any offsets.

VI. Conclusions

In this paper, we have proposed the composition of a new *Virtual Visual space* (measured in pixels) to define visual poses (positions and orientations). This composition converts the visual feature space to a minimum set of generalized variables $X_s \in \mathbb{R}^{3\times1}$. Using this visual space, we design a novel full-rank image Jacobian, which avoids the well-known problems in image-based Visual Servoing such as the Image space singularities, local minima and motion coupling. Analytic, simulation and experimental results show that this visual Jacobian surpass the standard visual Jacobians based on the classical *interaction matrix*. We also presented the design of a control approach using the image Jacobian and implemented it in a real robot. We are working on the implementation of this approach to validate its feasibility in dual-arm manipulation tasks.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Rob. and Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

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

[3] T. Sahin and E. Zergeroglu, "Adaptive visual servo control of robot manipulators via composite camera inputs," in *Int. Workshop on Robot Motion and Control*, June 2005, pp. 219–224.

[4] M. Marey and F. Chaumette, "Analysis of classical and new visual servoing control laws," in *IEEE ICRA*, May 2008, pp. 3244–3249.

[5] F. Janabi-Sharifi, L. Deng, and W. Wilson, "Comparison of Basic Visual Servoing Methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, Oct. 2011.

[6] J. Feddema, C. S. G. Lee, and O. Mitchell, "Model-based visual feedback control for a hand-eye coordinated robotic system," *Computer*, vol. 25, no. 8, pp. 21–31, Aug. 1992.

[7] Y. Mezouar and F. Chaumette, "Optimal camera trajectory with image-based control." *The International Journal of Robotics Research*, vol. 22, no. 10, pp. 781–804, 2003.

[8] E. Nematollahi and F. Janabi-Sharifi, "Generalizations to Control Laws of Image-Based Visual Servoing," *International Journal of Optomechatronics*, vol. 3, no. 3, pp. 167–186, 2009.

[9] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

[10] H. Wang, Y.-H. Liu, and D. Zhou, "Dynamic Visual Tracking for Manipulators Using an Uncalibrated Fixed Camera," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 610–617, June 2007.

[11] L. Pari, J. Sebastián, A. Traslosheros, and L. Angel, "Image Based Visual Servoing: Estimated Image Jacobian by Using Fundamental Matrix VS Analytic Jacobian," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, A. Campilho and M. Kamel, Eds. Springer Berlin Heidelberg, 2008, vol. 5112, pp. 706–717.

[12] S. Azad, Farahmand, Amir-Massoud, and M. Jagersand, "Robust Jacobian estimation for uncalibrated visual servoing," in *IEEE ICRA*, May 2010, pp. 5564–5569.

[13] E. Cervera, A. P. D. Pobil, F. Berry, and P. Martinet, "Improving Image-Based Visual Servoing with Three-Dimensional Features." *Int. J. of Robotics Research*, vol. 22, no. 10-11, pp. 821–840, 2003.

[14] E. Dean-Leon, V. Parra-Vega, A. Espinosa-Romero, and J. Fierro, "Dynamical image-based PID uncalibrated visual servoing with fixed camera for tracking of planar robots with a heuristical predictor," in *IEEE INDIN*, June 2004, pp. 339–345.

[15] C. Cai, E. Dean-Leon, D. Mendoza, N. Somani, and A. Knoll, "Uncalibrated 3D Stereo Image-based Dynamic Visual Servoing for Robot Manipulators," in *IEEE/RSJ IROS*, Nov. 2013.

[16] M. Gridseth, C. P. Quintero, R. Tatsambon-Fomena, O. Ramirez, and M. Jagersand, "Bringing Visual Servoing into Real World Applications," in *In Human Robot Collaboration workshop in RSS*, June 2013.

[17] B. Lamiroy, C. Puget, and R. Horaud, "What metric stereo can do for visual servoing," in *IEEE IROS*, vol. 1, 2000, pp. 251–256 vol.1.