

6D Image-based Visual Servoing for Robot Manipulators with uncalibrated Stereo Cameras

Caixia Cai¹, Emmanuel Dean-León², Nikhil Somani¹, Alois Knoll¹

Abstract—This paper introduces 6 new image features to provide a solution to the open problem of uncalibrated 6D image-based visual servoing for robot manipulators, where the goal is to control the 3D position and orientation of the robot end-effector using visual feedback. One of the main contributions of this article is a novel stereo camera model which employs virtual orthogonal cameras to map 6D Cartesian poses defined in the *Task space* to 6D visual poses defined in a *Virtual Visual space* (Image space). This new model is used to compute a *full-rank square Image Jacobian matrix* (J_{img}), which solves several common problems exhibited by the classical image Jacobians, e.g., Image space singularities and local minima. This Jacobian is a fundamental key for the image-based controller design, where a chattering-free adaptive second order sliding mode is employed to track 6D visual motions for a robot manipulator. Exponential convergence of errors in both spaces without local minima are demonstrated. The complete control system is experimentally evaluated on a real industrial robot. The robustness of the control scheme is evaluated for cases where the extrinsic parameters of the uncalibrated stereo camera system are changed on-line and unknown when the stereo system is manually moved to obtain a clearer view of the task.

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]. We are concerned with *Image-Based Visual Servoing* (IBVS) where the error function is defined directly in terms of image features.

This work aims to build upon the concepts of image-based visual servoing and attempt to address some of the most common problems plaguing conventional approaches by introducing additional features and behaviors. As pointed out in [1] and [2], 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. During the last decade, several authors have worked on solving these problems. In the following section, we will briefly describe the most important approaches used in IBVS and discuss about their properties and limitations.

Authors Affiliation: ¹ Technische Universität München, Fakultät für Informatik. ² Technischen Universität München, Fakultät für Elektrotechnik und Informationstechnik.

Email: {caica,somani,knoll}@in.tum.de

A. Related work

An IBVS usually employs the *image Jacobian matrix* (J_{img}) to relate end-effector velocities in the manipulator's Task space to the feature parameter velocities in the feature (image) space. A full and comprehensive survey on Visual Servoing and image Jacobian definitions can be found in [1], [3], [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 robot manipulator pose changes $\dot{s} = J_{img}v$. In Visual Servoing the image Jacobian needs to be calculated or estimated. Its inverse is used 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}).

In general, the image Jacobian can be computed using direct depth information (depth-dependent Jacobian) [6],[7], by approximation via on-line estimation of depth of the features (depth-estimation Jacobian) [3], [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], [13]. 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 $s \in \mathbb{R}^{2p}$ (with p 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 and local minima.

In our early work [14], we introduced a stereo camera model based on a virtual composite camera system, that can provide 3D visual position vector. This visual position vector generates a 3×3 full-rank Image Jacobian that maps velocities from the Task space to the Virtual Visual space. Using this image Jacobian, a visual servo control is implemented to drive 3 DOF of a real robot to trace time-varying desired trajectories defined in an uncalibrated Image space. However, this 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 (Image space). This visual pose is composed of 3D visual position and 3D visual orientation, which is used to obtain a 6×6 *full-rank square Image Jacobian* to map velocities from the Task space to the 3D Virtual Visual space. Therefore, this work offers a general solution to control the position and orientation of a robot end-effector using uncalibrated visual information with

a new *full-rank square image Jacobian* (J_{img}).

B. Organization

This paper is organized as follows: In Section II we highlight the problems in classical image-based visual servoing approaches and state the core issues which we tackle in this work. In Section III we introduce a new *Camera Model* and describe how it is used to construct a *Virtual Visual space*. This model is used to define the full-rank *6D Visual Jacobian*, which will be used in the Section IV to design an adaptive image-based 6D visual servoing. Section V presents two real-world experiments (as illustrated in Fig. 1) and shows the results obtained in a dynamic environment. Finally, Section VI draws the conclusions and presents directions for future work.

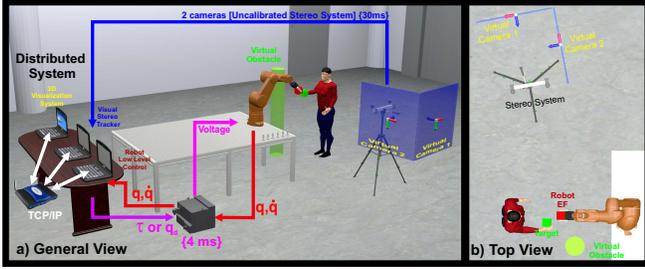


Fig. 1. Description of robotic experimental setup.

II. PROBLEM FORMULATION

A. Classical Image-Based Visual Servoing

Suppose that the robot end-effector is moving with angular velocity $\omega_c = [\omega_x, \omega_y, \omega_z]$ and translational velocity $v_c = [v_x, v_y, v_z]$ both w.r.t. the camera frame in a fixed camera system (eye-to-hand configuration). Let $P_c = [x, y, z]^T$ be a point rigidly attached to the end-effector. The velocity of the point P_c , expressed relative to the camera frame, is given by

$$\dot{P}_c = \omega_c \times P_c + v_c \quad (1)$$

where $\dot{P}_c = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$ is the relative velocity of the point in the camera frame.

As described in [1], we can relate image-plane velocity of a point to the relative velocity of the point with respect to the camera as

$$\dot{s} = L_x \dot{P}_c \quad (2)$$

in which $s = [u, v]^T$ is the image feature parameters, and the interaction matrix L_x is defined as

$$L_x = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} & -\frac{uv}{f} & \frac{f^2+u^2}{f} & -v \\ 0 & \frac{f}{z} & -\frac{v}{z} & \frac{uv}{f} & \frac{f^2+v^2}{f} & u \end{bmatrix} \quad (3)$$

where f is the focal length of the camera lens.

To control a 6 DOF robot, at least three points are necessary (i.e., we require $k \geq 6$, where k represents the total number of feature measurements). If we have a vector $W_b = [X_b, \theta_b]^T = [x_b, y_b, z_b, \alpha_b, \beta_b, \gamma_b]^T \in \mathbb{R}^{6 \times 1}$, which is the pose of the end-effector in the robot base frame (in this case we choose Euler angles to represent the orientation of the

end-effector), and a vector $s = [u_1, v_1, \dots, u_p, v_p]^T \in \mathbb{R}^{2p \times 1}$, which contains $p = k/2$ image points. Then, the relation between \dot{s} and \dot{W}_b is given by

$$\underbrace{\dot{s}}_{2p \times 1} = \underbrace{J_x}_{2p \times 6} \underbrace{\dot{W}_b}_{6 \times 1} \quad (4)$$

where $J_x = [L_{x1}, \dots, L_{xp}]^T \cdot M \in \mathbb{R}^{2p \times 6}$ is known as the image Jacobian, $M \in \mathbb{R}^{6 \times 6}$ is the mapping to transform velocities expressed in the camera frame to the robot base frame, and L_{xi} is given by (3).

B. The problems of Classical IBVS

If we consider ΔW_b as the input to a robot controller, then we need to compute the inverse mapping of \dot{s} as

$$\Delta W_b = J_x^+ \Delta s, \quad (5)$$

where Δ^* is an error function defined in the space $*$, $J_x^+ \in \mathbb{R}^{6 \times 2p}$ is chosen as the Moore-Penrose pseudoinverse 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 $2p > 6$. In this case, the image Jacobian is singular when $rank(J_x) < 6$, while the image local minima is defined as the set of image locations $\Omega_s = \{s | \Delta s \neq 0, \dot{W}_b = 0, \forall s \in \mathbb{R}^{2p \times 1}\}$ when using redundant image features. (A local minimum is reached since the velocity of the end-effector is zero while the final end-effector position is far from its desired one). Examples of the problems generated by the local minima conditions are illustrated in [2] and [3].

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 an intermediate 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 it is composed of 3D visual position and 3D visual orientation.

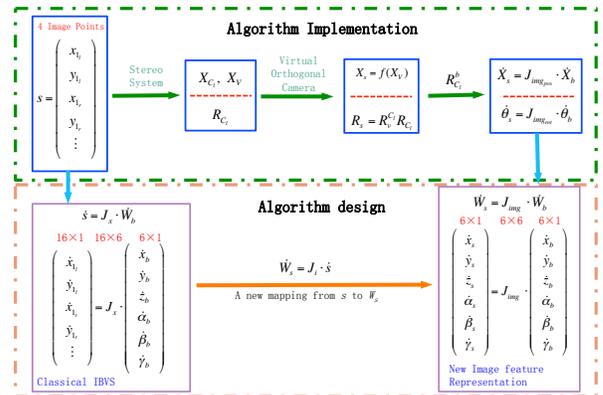


Fig. 2. Algorithm framework: The figure shows the philosophy behind the algorithm proposed in this work (orange box) and its practical implementation (green box).

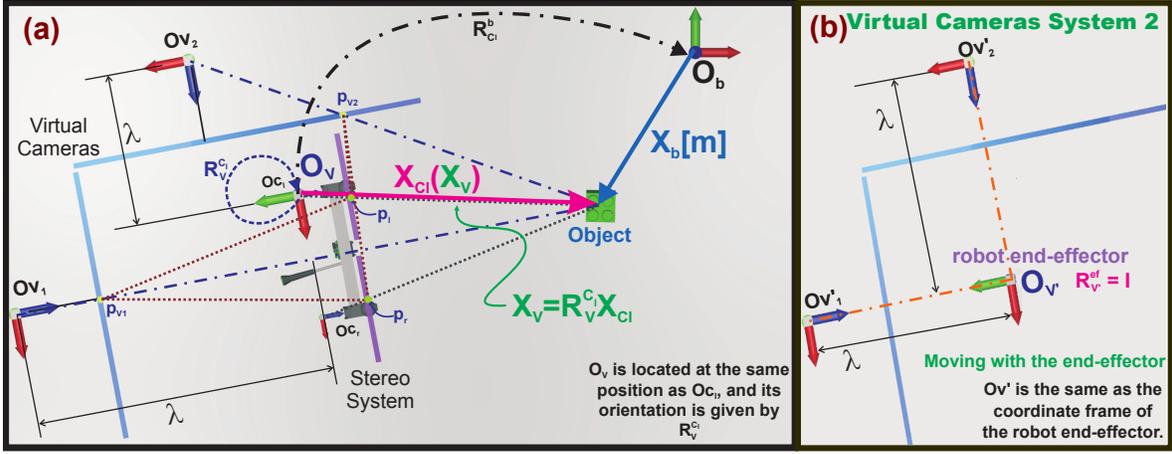


Fig. 3. Image Projections: a) The figure depicts the different coordinate frames used to obtain a general 3D visual camera model. $X_b \in \mathbb{R}^{3 \times 1}$ is the position in meters [m] of an *Object* with respect to the world coordinate frame (wcf) denoted by O_b . $R_{C_1}^b \in SO(3)$ represents the orientation of wcf w.r.t the left camera. O_V is a reference frame for the virtual orthogonal cameras $O_{v_{1,2}}$ where $R_{C_1}^V \in SO(3)$ is the orientation of frame O_{C_1} with respect to O_V . The vectors $p_l, p_r \in \mathbb{R}^{2 \times 1}$ are the projections of the point X_b in the left and right cameras. Finally, $p_v \in \mathbb{R}^{2 \times 1}$ represents the projection of the *Object* in the virtual cameras O_{v_i} . b) Second virtual Camera System, whose reference coordinate frame $O_{V'}$ is fixed on the robot end-effector and has the same rotation as the frame O_{ef} . And its orientation with respect to the first virtual camera system O_V is $R_{V'}^c \in SO(3)$.

The visual representation W_s is related with the feature points vector s as

$$\underbrace{\dot{W}_s}_{6 \times 1} = \underbrace{J_i}_{6 \times 2p} \underbrace{\dot{s}}_{2p \times 1} \quad (6)$$

In other words $W_s \in \mathbb{R}^{6 \times 1}$ can be seen as the decomposition of the vector $s \in \mathbb{R}^{2p \times 1}$ in its principal components, in such a way that all the elements of W_s are independent and orthogonal to each other.

Substituting (4) in to (6) yields a new mapping as

$$\dot{W}_s = (J_i J_x) \dot{W}_b \quad (7)$$

$$\begin{bmatrix} \dot{X}_s \\ \dot{\theta}_s \end{bmatrix}_{6 \times 1} = \underbrace{J_{img}}_{6 \times 6} \begin{bmatrix} \dot{X}_b \\ \dot{\theta}_b \end{bmatrix}_{6 \times 1} \quad (8)$$

The advantage of this intermediate mapping is that, a full-rank square *image Jacobian matrix* ($J_{img} \in \mathbb{R}^{6 \times 6}$) can be obtained, which is nonsingular and without local minima. This is the core design of our algorithm, and all that remains is to define the form of J_{img} . Fig. 2 depicts our algorithm framework.

III. 6D IMAGE JACOBIAN

This section shows how we construct the *Virtual Visual space* using information generated by the stereo vision system. It also explains in detail how to implement the algorithm to obtain the full-rank *image Jacobian matrix* ($J_{img} \in \mathbb{R}^{6 \times 6}$).

In this work, the key idea of the 3D visual camera model is to combine the stereo camera model with a virtual composite camera model. Fig. 3 (a) denotes our new visual camera model and the image projections.

A. Image Jacobian for 3D position $J_{img_{pos}}$

We first define the mapping for 3D positions ($J_{img_{pos}} \in \mathbb{R}^{3 \times 3}$), that shows the relationship between 3D Cartesian

position (*meters*) and 3D visual position (*pixels*). The key idea of this model is to combine the stereo camera model with a virtual composite camera model to get a full-rank square image Jacobian to map velocities of a target object (\dot{X}_b) to velocities of the image features (in our case, pixel velocities in the 3D visual space, denoted here by \dot{X}_s), see Fig. 3.

As presented in our earlier paper [14], the relationship between the pixel velocities (\dot{X}_s) and the object position velocities (\dot{X}_b) in the robot base frame can be rewritten as

$$\dot{X}_s = (J_\alpha R_V^C R_{C_1}^b) \dot{X}_b = J_{img_{pos}} \dot{X}_b \quad (9)$$

where we define the Jacobian $J_{img_{pos}} \in \mathbb{R}^{3 \times 3}$ as the *position image Jacobian* and J_α consists of user-defined virtual camera parameters. $R_{C_1}^b \in SO(3)$ represents the orientation of wcf w.r.t the left camera and $R_V^C \in SO(3)$ is the orientation of frame O_{C_1} with respect to O_V , see Fig. 3.

B. Image Jacobian for 3D Orientation $J_{img_{rot}}$

In order to define orientations in the *Virtual Visual Space*, we need to define 4 different points rigidly attached to the robot end-effector, which can be used to represent the orientation of the end-effector in a base frame. The 4 points expressed in the end-effector frame O_{ef} , are the origin and the canonical basis of a 3D Euclidean space, which means

$$\begin{cases} \text{origin point: } X_{ef} = [0, 0, 0]^T \\ \text{x axis: } X_{e1} = [1, 0, 0]^T \\ \text{y axis: } X_{e2} = [0, 1, 0]^T \\ \text{z axis: } X_{e3} = [0, 0, 1]^T \end{cases} \quad (10)$$

1) *Orientation Definition in Virtual Visual Space*: In order to specify 3 orthogonal vectors in the Virtual Visual space, which can be used to represent visual orientations, we define a second virtual orthogonal camera system, see

Fig. 3 (b). This virtual coordinate frame $O_{V'}$ is fixed to the robot end-effector coordinate frame O_{ef} . All parameters in the second virtual camera system are the same as the first one, with the exception of the optical center offsets $O_1 = O_2 = [0, 0]^T$. Then, the mapping from 3D Cartesian position w.r.t the frame O_v to the 3D image position (check [14] (10)) for the new 3D Virtual Visual space $O_{s'}$ is written as

$$X_{s'} = \text{diag}(f\beta) \begin{bmatrix} \frac{x_{V'}}{\frac{-y_{V'} + \lambda}{x_{V'} + \lambda}} \\ \frac{y_{V'}}{\frac{-y_{V'} + \lambda}{x_{V'} + \lambda}} \\ \frac{z_{V'}}{\frac{-y_{V'} + \lambda}{x_{V'} + \lambda}} \end{bmatrix} + \begin{bmatrix} c_x \\ c_x \\ c_y \end{bmatrix} \quad (11)$$

where $X_{V'} = [x_{V'}, y_{V'}, z_{V'}]^T$ is the 3D position w.r.t. $O_{V'}$.

Therefore, the 4 points defined in (10) can be represented in $O_{s'}$ as

$$\begin{cases} X_{V'_{ef}} = X_{e_{ef}} = [0, 0, 0]^T \\ X_{V'_{1}} = X_{e_1} = [1, 0, 0]^T \\ X_{V'_{2}} = X_{e_2} = [0, 1, 0]^T \\ X_{V'_{3}} = X_{e_3} = [0, 0, 1]^T \end{cases} \Rightarrow \begin{cases} X_{s'_{ef}} = [c_x, c_x, c_y]^T \\ X_{s'_{1}} = [c_x + \frac{f\beta}{\lambda}, c_x, c_y]^T \\ X_{s'_{2}} = [c_x, c_x + \frac{f\beta}{\lambda}, c_y]^T \\ X_{s'_{3}} = [c_x, c_x, c_y + \frac{f\beta}{\lambda}]^T \end{cases} \quad (12)$$

In the same form, using (12), we define a basis of the 3D Virtual Visual space (expressed in pixels): $V_i = X_{s'_i} - X_{s'_{ef}}$

$$V_1 = [\frac{f\beta}{\lambda}, 0, 0]^T, \quad V_2 = [0, \frac{f\beta}{\lambda}, 0]^T, \quad V_3 = [0, 0, \frac{f\beta}{\lambda}]^T \quad (13)$$

From (13), it is evident that the rotation of the robot end-effector in the new 3D Virtual Visual space is represented as:

$$R_{s'}^{ef} = \begin{bmatrix} \frac{V_1}{\|V_1\|} & \frac{V_2}{\|V_2\|} & \frac{V_3}{\|V_3\|} \end{bmatrix} = I.$$

In Section III-A, we mapped the 3D positions to the first virtual orthogonal camera system, which is attached to the stereo vision system. Therefore, the visual orientation also needs to be mapped to the same frame O_s . The rotation of the end-effector with respect to O_s can be computed as

$$R_s^{ef} = R_s^s R_{s'}^{ef} = R_s^s R_{V'}^V R_{V'}^{s'} = R_V^V = R_V^{ef} \quad (14)$$

$$= (R_V^{C_l} R_{C_l}^b) R_b^{ef} \quad (15)$$

where $R_{V'}^s = I$ and $R_s^V = I$.

2) **Orientation Mapping $J_{img_{rot}}$:** Given a rotation matrix R , the angular velocity of the rotating frame can be represented as $S(\omega) = \dot{R}R^T$ in which $S(\omega)$ is a skew symmetric matrix and ω is the angular velocity.

Therefore, the angular velocity of the end-effector frame with respect to the robot base frame (ω_b) is given by

$$S(\omega_b) = \dot{R}_b^{ef} (R_b^{ef})^T \quad (16)$$

and using (15)¹ the angular velocity of the end-effector frame with respect to O_s (ω_s) is given by

$$S(\omega_s) = \dot{R}_s^{ef} (R_s^{ef})^T = (R_V^{C_l} R_{C_l}^b) \dot{R}_b^{ef} (R_b^{ef})^T (R_V^{C_l} R_{C_l}^b)^T \quad (17)$$

$$= (R_V^{C_l} R_{C_l}^b) S(\omega_b) (R_V^{C_l} R_{C_l}^b)^T \quad (18)$$

$$= S((R_V^{C_l} R_{C_l}^b) \omega_b) \quad (19)$$

¹Properties of Skew Symmetric matrices show that: $RS(\alpha)R^T = S(R\alpha)$ with $R \in SO(3)$ and $\alpha \in \mathbb{R}^3$.

From (19) we can obtain the visual angular velocity

$$\omega_s = (R_V^{C_l} R_{C_l}^b) \omega_b. \quad (20)$$

Now, let $\theta = [\alpha, \beta, \gamma]^T$ be a vector of Euler angles, which denotes a minimal representation for the orientation of the end-effector frame relative to the robot base frame. Then, the definition of the angular velocity ω is given by [15]

$$\omega = T(\theta) \dot{\theta}. \quad (21)$$

If $R_{ef} = R_{z,\gamma} R_{y,\beta} R_{x,\alpha}$ is the Euler angle transformation, then

$$T(\theta) = \begin{bmatrix} \cos(\gamma) \cos(\beta) & -\sin(\gamma) & 0 \\ \sin(\gamma) \cos(\beta) & \cos(\gamma) & 0 \\ -\sin(\beta) & 0 & 1 \end{bmatrix} \quad (22)$$

Singularities of the matrix $T(\theta)$ are called *representational singularities*. It can easily be shown that $T(\theta)$ is invertible provided $\cos(\beta) \neq 0$.

Substituting (21) into (20) we obtain

$$T(\theta_s) \dot{\theta}_s = (R_V^{C_l} R_{C_l}^b) T(\theta_b) \dot{\theta}_b \quad (23)$$

Furthermore, this expression can be written as

$$\dot{\theta}_s = \begin{bmatrix} \dot{\alpha}_s \\ \dot{\beta}_s \\ \dot{\gamma}_s \end{bmatrix} = \overbrace{T(\theta_s)^{-1} (R_V^{C_l} R_{C_l}^b) T(\theta_b)}^{J_{img_{rot}}} \dot{\theta}_b = J_{img_{rot}} \dot{\theta}_b \quad (24)$$

where the Jacobian $J_{img_{rot}} \in \mathbb{R}^{3 \times 3}$ is defined as the *orientation image Jacobian*.

C. Visual Jacobian

In the previous sections, we have defined the mappings for the 3D visual position and orientation separately as *position image Jacobian $J_{img_{pos}}$* and *orientation image Jacobian $J_{img_{rot}}$* . Combining (9) and (24) we have the full expression

$$\dot{W}_s = \begin{bmatrix} \dot{X}_s \\ \dot{\theta}_s \end{bmatrix} = \begin{bmatrix} J_{img_{pos}} & 0 \\ 0 & J_{img_{rot}} \end{bmatrix} \begin{bmatrix} \dot{X}_b \\ \dot{\theta}_b \end{bmatrix} = J_{img} \dot{W}_b \quad (25)$$

where the Jacobian $J_{img} \in \mathbb{R}^{6 \times 6}$ is defined as the **Image Jacobian**.

Substituting the robot *Differential Kinematics* $\dot{W}_b = J_a(q) \dot{q}$, (25) can be rewritten in the form

$$\dot{W}_s = J_{img} J_a(q) \dot{q} = J_s \dot{q} \quad (26)$$

where $J_a(q) \in \mathbb{R}^{6 \times 6}$ is the analytical Jacobian matrix of the robot manipulator and the Jacobian matrix $J_s \in \mathbb{R}^{6 \times 6}$ is defined as the **Visual Jacobian**.

Then the inverse differential kinematics that relates generalized joint velocities \dot{q} and 6D visual velocities \dot{W}_s is given by

$$\dot{q} = J_s^{-1} \dot{W}_s = J_a(q)^{-1} J_{img}^{-1} \dot{W}_s \quad (27)$$

Remark 1: Singularity-free J_{img} . From (25), we can see that $\det(J_{img}) = \det(J_{img_{pos}}) \det(J_{img_{rot}})$, therefore the set of singular configurations of J_{img} is the union of the set of position configurations satisfying $\det(J_{img_{pos}}) = 0$ and the set of orientation configurations satisfying $\det(J_{img_{rot}}) = 0$.

From [14], we know that a non-singular $J_{img_{pos}}$ can be obtained, and $J_{img_{rot}}$ is non-singular provided that $T(\theta)$ is invertible.

Therefore, the singularities of J_s in (27) are defined only by the singularities of $J_a(q)$. Hence, to guarantee a non-singular visual mapping, an approach to avoid robot singularities must be implemented. In Section V, we discuss this issue and propose a solution.

IV. CONTROL LAW

In this section, we describe the design of an adaptive image-based dynamic control (second order sliding mode control) which includes the robot dynamics model in its passivity proof. The proposed second order sliding mode control is chattering free.

The robot dynamic model and joint velocity nominal reference are the same as described in our earlier paper [14]. Here the 6D visual nominal reference $\dot{W}_{sr} \in \mathbb{R}^{6 \times 1}$ is used instead of the $\dot{X}_{sr} \in \mathbb{R}^{3 \times 1}$ as

$$\dot{q}_r = J_s^{-1} \dot{W}_{sr}. \quad (28)$$

Details about the control law are shown in the paper [14].

Remark 2: Convergence of ΔW_b without local minima. Given J_{img} is square and full-rank, from (25) it can be seen that $\Delta W_s = 0 \rightarrow \Delta W_b = 0$ without local minima. This is the most important impact of designing a full-rank image Jacobian which, in general, is not obtained with classical methods.

V. EXPERIMENTS

Two experiments were performed to validate and evaluate this work on a standard industrial robot. The first experiment focuses on the 6D visual servoing algorithm as an application of a teaching interface, where the user defines the pose of the end-effector using a visual marker and this information is later used to define the desired visual trajectory. The second experiment uses the 6D visual servoing algorithm for real-time tracking of a moving target in a Human-Robot interaction scenario. In order to make this a practically useful application, several other features such as singularity avoidance, self-collision avoidance and obstacle detection and avoidance are implemented, which ensure safety of the robot and human. This experiment also shows how the orientation matrix is estimated on line enabling an uncalibrated visual servoing system, where occlusions due to camera placement can be handled in a natural and intuitive way by simply moving manually the camera to a better position.

This system consists of 3 sub-systems: a) the Visual Stereo Tracker, b) the Robot Control System and c) the 3D visualization System, see Fig. 1.

A. Experiment 1: 6D Visual Tracking

This scheme has been implemented on a robotic platform with six degrees of freedom and an eye-to-hand configuration. 2D visual (image) features are extracted from a stereo vision system with AR markers. We use the ArUco library² (based on OpenCV) to detect markers. Every marker

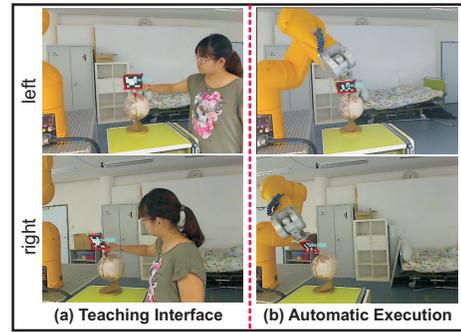


Fig. 4. Snapshot of the 6D visual tracking.

provides 2D image features for 4 corner points. 3D position and rotation of the robot end-effector with respect to the camera frame are obtained from the image features of these 4 points. This experiment consists of two phases: teaching and execution.

1) **Teaching Interface:** In this phase, we provide a teaching interface for the user, see Fig. 4 (a), where the user is holding an AR marker, which is detected by the stereo camera system that provides 2D image features. A red square and a marker ID (displayed in cyan) on the image indicates the detection. The user moves the marker, creating some visual trajectories, such as two orthogonal straight lines on the table and two smooth curves on the surface of the Globe. These trajectories include both translation and rotation motions. During the movement, the 2D features for 4 points of the target in the camera frames are recorded and saved. At some points, when the marker is lost or can not be detected, the last available data is saved, which guarantees that the desired pose can be reached and is safe for robot execution.

2) **Automatic Execution:** After the teaching phase, the robot can automatically execute the recorded visual trajectories. Another AR marker with the same size is attached to the robot end-effector to detect the pose of the robot during execution. In this case, inputs to visual servoing are the 2D image features which were recorded in the teaching phase. From these features we extract our new 6D visual pose vector W_s in the Virtual Visual space, which is used to create the error function, see Section III.

Experimental results are depicted in Fig. 5. The plots in the first row (a) show the 3D linear and 3D angular visual tracking in our *Virtual Visual space* while the second row (b) shows the target trajectory tracking in the *3D Task space*. The red lines in the plots are the target trajectories, which exhibit some noise and chattering due to the unsteady movement of the user. However, the blue lines which illustrate the trajectories of the robot end-effector, are smooth and chattering free. This experiment shows that when the errors in the Virtual Visual space converge to zero, the errors in the Task space converge to zero without local minima. Therefore during execution, the AR marker on the robot end-effector shows identical linear and angular motions as instructed in the teaching part.

²ArUco: <http://www.uco.es/investigacion/grupos/ava/node/26>

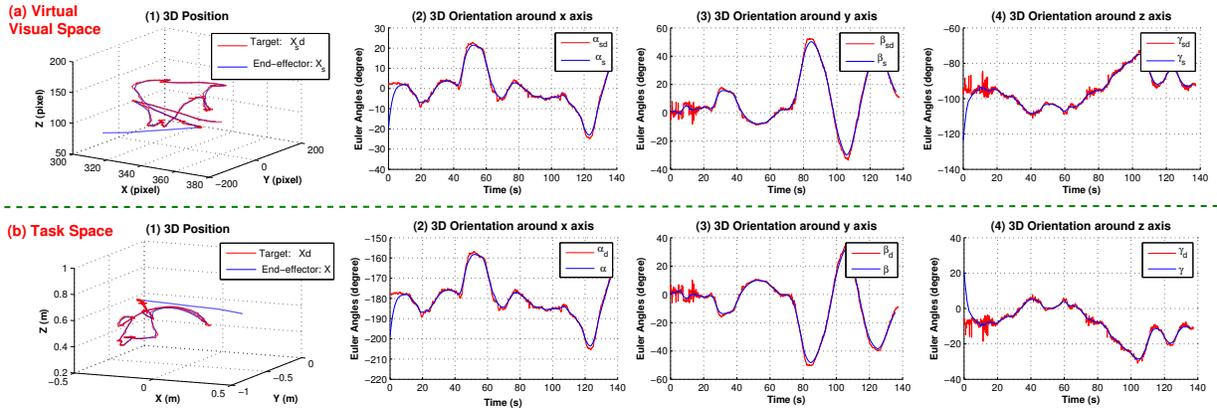


Fig. 5. Experiment results for 6D visual tracking: (a) 3D position and orientation angles in the Virtual Visual Space, (b) 3D position and orientation angles in the Task Space.

B. Experiment 2: 6D Uncalibrated Image-based Visual Servoing in a Human-Robot Interaction Scenario

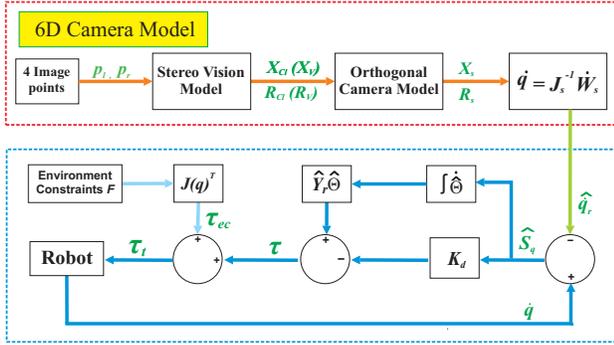


Fig. 6. 6D adaptive visual servoing including environment constraints.

In this experiment, we integrate the visual servoing system in a scenario where environment constraints such as: robot singularities avoidance, (self-/obstacle) collision avoidance must be included to generate a safe and singularity-free trajectory for the robot. We model the environment constraints as a total force F , which includes singularity avoidance force F_r , self-collision avoidance force F_c , obstacle-collision avoidance force F_o , etc. Fig. 6 shows integration of the control with these environment constraints.

In this task, we demonstrate real time tracking using AR markers for identifying the target pose and the current pose of the robot end-effector. The target is carried by a human, and the control goal is to make the robot end-effector follow the target placed in the human's hand.

1) Interaction results: This experiment demonstrates real time tracking of both the 3D visual position and 3D orientation (Fig. 7 (a)). The system proves to be stable and safe even in situations where the target is lost (due to occlusions by the robot or the human), 7 (b). In this case, the robot pauses and the visual tracking is resumed as soon as the target is visible again.

To demonstrate stability, we test our system under several environment constraints. Fig. 7 (c) illustrates the results of

singularity avoidance, where the robot does not reach the singular condition ($q_3 = 0$), even when the user tries to force it. Fig. 7 (d) depicts table avoidance where the motion of the robot is constrained in the z_b - axis by the height of the table (the end-effector is not allow to go under the table) but it can still move in the x_b and y_b axes, and Fig. 7 (e) shows how the robot handles self-collisions. Fig.7 (f) shows obstacle avoidance while continuing to track the target.

One of the key contributions of this paper is the possibility of handling situations where the target object is occluded, and the stereo system can be moved to maintain the target in the field of view. This feature is analyzed in next subsection.

2) On-line Orientation Matrix Estimation: In order to compute the visual Jacobian, in this work a coarse on-line estimation of the orientation matrix is computed using the real-time information generated by the robot. As shown in our previous work [14], SVD (Singular Value Decomposition) on two sets of position points defined in each coordinate frame O_b and O_C is used to estimate the orientation matrix. The estimation errors for the complete Jacobian J_s can be handled in the controller to some extent.

Object occlusion occurs in Fig. 7 (g), where the stereo system can then be moved physically to maintain targets in the field of view. The camera motion is detected by the system and a process for the coarse estimation of the orientation matrix between the stereo system and the robot base frame is started. The robot performs a small motion and a set of points are collected, shown in Fig. 8 (d). We compare the actual end-effector position with the recovered end-effector position using the estimated rotation. Fig. 8 (a) and (c) depicts these two positions trajectories. It can be clearly observed that the error between them is close to zero after on-line rotation matrix estimation, as illustrated in Fig. 8 (b).

A video where more details for all these experimental results are illustrated can be seen in: <http://youtu.be/arNFrBJOLj4>

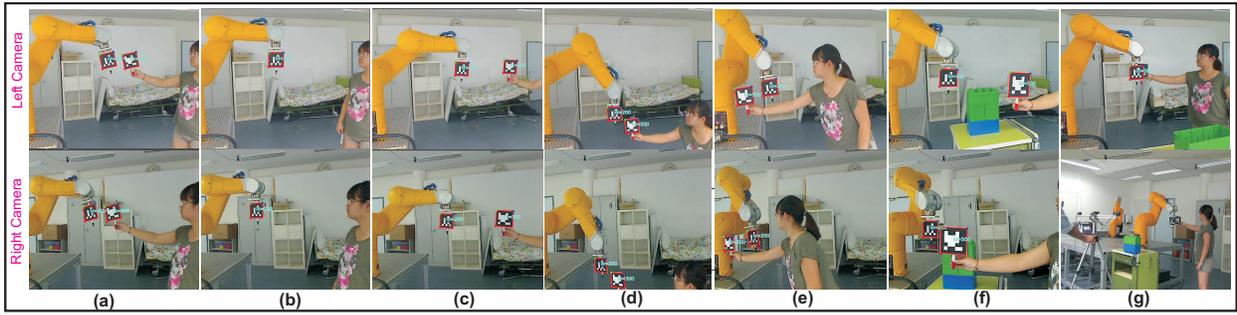


Fig. 7. System behaviors: (a) Position and orientation tracking, (b) Case when the target is lost, (c) Case with singularity avoidance, (d) Case with table collision avoidance, (e) Case with self-collision avoidance and (f) Obstacle avoidance. (g) Target occlusion and the user manually moves the stereo system.

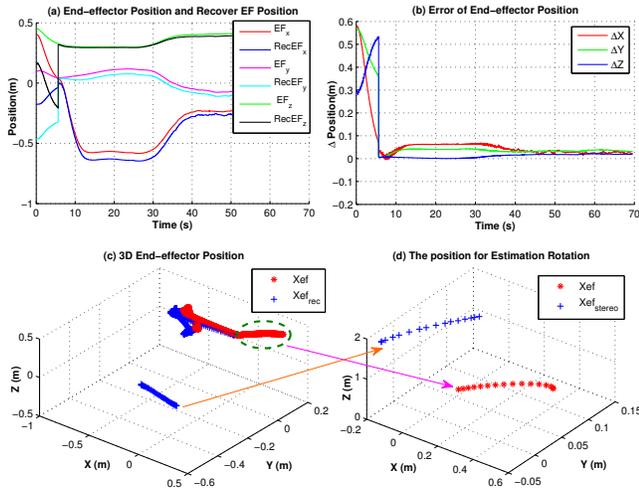


Fig. 8. Results of the on-line orientation matrix estimation: (a) and (c) show the 3D position of the end-effector in the base frame and the recovered position for the end-effector using the estimated rotation, (b) shows the errors, (d) illustrates the selected points for the estimation of the orientation.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new *Virtual Visual space* (measured in pixels) for visual servoing from 4 image points using an uncalibrated stereo vision system with virtual orthogonal cameras, in which we define a 6D visual pose vector. Using this 6D visual pose we obtain a new *full-rank square Image Jacobian*, which can avoid the well-known problems in image-based Visual Servoing such as Image space singularities and local minima. Then, an adaptive second order sliding mode visual servo control is designed to track 6D visual motions using the 6D trajectory errors defined in the *Virtual Visual space*. The control was evaluated on a real industrial robot to experimentally show stability of the control. Furthermore, to create a practically useful and safe scenario, environment constraints were integrated with our 6D visual servoing approach to generate a robot dynamics control system with a trajectory free of collisions and singularities.

Experimental results on a robotics platform show the feasibility of the proposed scheme in a real-world environment. Future work includes the study of the efficiency and

robustness of the proposed scheme when compared to the classic and other image-based visual servoing approaches. We will also try to use an advanced object tracker instead of using the AR markers. We will implement this approach in more real world applications, by fusing this technique with various sensors, e.g. depth-cameras and tactile sensors.

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [2] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*. LNCIS Series, No 237, Springer-Verlag, 1998, pp. 66–78.
- [3] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
- [4] M. Marey and F. Chaumette, "Analysis of classical and new visual servoing control laws," in *IEEE International Conference on Robotics and Automation*, 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] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, vol. 1, Sep. 1994, pp. 186–193.
- [12] J. Piepmeyer, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 143–147, Feb. 2004.
- [13] S. Azad, Farahmand, Amir-Massoud, and M. Jagersand, "Robust jacobian estimation for uncalibrated visual servoing," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 5564–5569.
- [14] 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 International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013.
- [15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control-Second Edition*, 2004.