2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Uncalibrated 3D Stereo Image-based Dynamic Visual Servoing for Robot Manipulators

Caixia Cai[†], Emmanuel Dean-León[‡], Dario Mendoza[§], Nikhil Somani[†] and Alois Knoll[†]

Abstract— **This paper introduces a new comprehensive solution for the open problem of uncalibrated 3D image-based stereo visual servoing for robot manipulators. One of the main contributions of this article is a novel 3D stereo camera model to map positions in the task space to positions in a new 3D *Visual Cartesian Space* (a visual feature space where *3D* positions are measured in pixel). This model is used to compute a full-rank *Image Jacobian Matrix* (J_{img}), which solves several common problems presented on the classical image Jacobians, e.g., image space singularities and local minima. This Jacobian is a fundamental key for the image-based control design, where uncalibrated stereo camera systems can be used to drive a robot manipulator. Furthermore, an adaptive second order sliding mode visual servo control is designed to track 3D visual motions using the 3D trajectory errors defined in the *Visual Cartesian Space*. The stability of the control in closed loop with a dynamic robot system is formally analyzed and proved, where exponential convergence of errors in the *Visual Cartesian Space* and *task space* without local minima are demonstrated. The complete control system is evaluated both in simulation and on a real industrial robot. The robustness of the control scheme is evaluated for cases where the extrinsic parameters of the stereo camera system change on-line and the kinematic/dynamic robot parameters are considered as unknown. This approach offers a proper solution for the common problem of visual occlusion, since the stereo system can be moved to obtain a clear view of the task at any time.**

I. Introduction

Visual servoing control (VSC) is an approach to control motion of a robot manipulator using visual feedback signals 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). Aditionally, 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 coordinates, is obtained from the visual information [2]. The conclusion drawn in many of the previous works, e.g.,[1] and [3], is that the IBVS method is more favorable than the PBVS method, since the IBVS has low sensitivity to camera calibration errors.

Two main aspects have a great impact on the behavior of any visual servoing scheme: the selection of the visual features used as input of the control law and the form of the control scheme. This work focuses on the study of IBVS design, where these two fundamental problems are handled: First, the definition of a new *image Jacobian* (J_{img}) for a stereo camera system in fixed-camera configuration (see Fig. 1). Second, the design of a *Dynamic Visual Servoing* control for robot manipulators, which includes *unknown camera and robot parameters* (kinematic and dynamic).

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], [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 robot manipulator pose changes $\dot{s} = J_{img}v$. In Dynamic Visual Servoing we are interested in determining the manipulator's required dynamical behavior to achieve some desired value of image feature parameters. This implies calculating or estimating the image Jacobian and using its inverse 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], 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, e.g. [9]. Additionally, many papers directly estimate on-line the complete image Jacobian in different ways, e.g. [10], [11]. 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 matrix needs to be computed, which leads to well-known problems such as the image space singularities and local minima. In order to obtain a full-rank Image Jacobian, [12] 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. However, this method requires the exact camera calibration, and in the presence of calibration errors the convergence of the error functions (in the image or in the task space) can not be guaranteed.

In the context of control schemes, kinematic-based con-

Authors Affiliation: †Technische Universität München, Fakultät für Informatik. ‡Technischen Universität München, Fakultät für Elektrotechnik und Informationstechnik. §UPIITA-IPN

Address: †Boltzmannstrae 3, 85748 Garching bei München (Germany). ‡Karlstrasse 45/II 80333 München (Germany). §Av. IPN 2508, Mexico-City (Mexico)

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

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

trols cannot yield high performance and guarantee stability because the nonlinear forces in robot dynamics are neglected. This problem known as *Dynamic Visual Servoing* was studied by Hashimoto et al.[13] for the eye-in-hand setup and by Zergeroglu et al. [14] for the fixed-camera configuration. Their methods work well when the camera intrinsic and extrinsic parameters are known. In order to avoid tedious and costly camera calibration, various authors [15], [16], [17], [18] and [19] proposed to employ an adaptive algorithm to estimate the unknown camera parameters on-line. However, these methods are applicable to planar manipulators only. The problem of 3D uncalibrated visual servoing with robot dynamics has been tackled with new adaptive controllers [9],[20]. However, the image jacobians in these approaches are not full-rank and they have the above mentioned problems.

In this paper, we address the problem of controlling a robot manipulator to trace time-varying desired trajectories defined in an uncalibrated image space, taking explicitly into account the robot dynamics in the design and in the passivity proof. The contribution of this paper can be summarized into three aspects: 1) The introduction of a new stereo camera model based on a virtual composite camera system. This model provides a 3D visual feature for the IBVS method. This feature is a representation of task space positions in a 3D visual space. 2) Using this stereo camera model, we design a new *image Jacobian* which maps velocities from the task space to the 3D visual space. The dimension of the image feature vector, obtained from the stereo camera model, allows a full-rank Jacobian, whose singularities depend exclusively on parameters defined by the user. 3) An adaptive second order sliding mode visual servo control using the proposed *image Jacobian* has been implemented for 3D visual motion tracking. The control law is designed to cope with uncertainties in the robot dynamic/kinematic and camera parameters. Moreover, we extend the implementation of this adaptive image-based control law to include environment constraints, such as: robot singularities avoidance and (self-/obstacle-) collision avoidance. By means of Lyapunov stability analysis, we demonstrate the convergence of errors in both the 3D Cartesian visual space and task space without local minima.

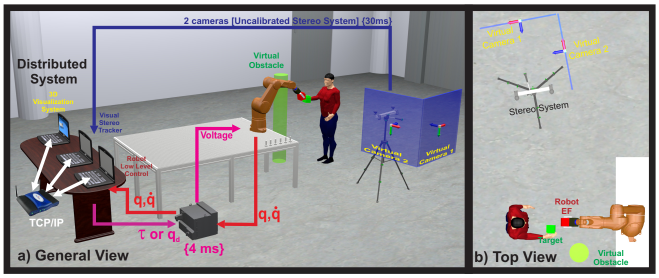

Fig. 1. Description of robotic experimental setup.

In order to evaluate the control approach and the proposed models, experiments have been conducted in both simulation and on a real industrial robot. In simulation, we control the robot without environment constraints to better illustrate the robustness of the system to uncertainties in the robot parameters and changes in the extrinsic parameters of the stereo system. In the experimental setup, we integrate the visual servoing system in a Human-Robot-Interaction (HRI) scenario (see Fig. 1). In this scenario, we reproduce a more realistic environment including obstacles, robot singularities, visual occlusion, collision avoidance and sporadic loss of targets. We prove that the system is always stable and can be used for HRI tasks.

A. Organization

This paper is organized as follows. First, in Section II we will introduce the new *3D Camera Model*. This model defines the complete *3D Visual Jacobian*, which will be used in the Section III to design an adaptive image-based 3D visual servoing. In the same section, the passivity proof is shown and a validation in simulation with remarks about robustness and adaptability of the control is given. In Section IV we present the adaptive image-based torque controller which includes environment constraints. We explain the capabilities of this control and its different properties, such as robot singularity avoidance and collision avoidance. Section V provides an overview of the HRI setup (Fig. 1) and shows the obtained results under different conditions. Finally, Section VI draws the conclusions and future work.

II. 3D Visual Camera Model

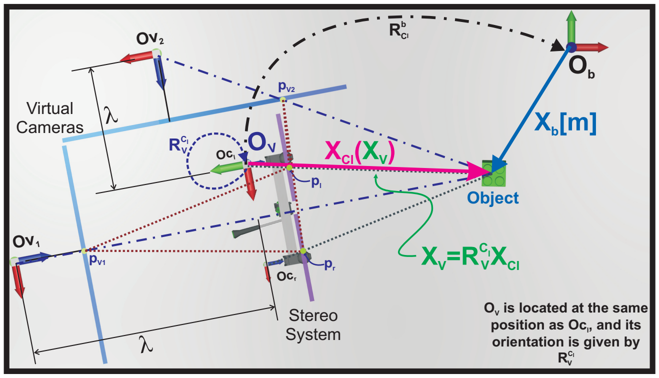

Fig. 2. Image Projections: The figure depicts the different coordinate frames used to obtain a general 3D visual camera model. $X_b \in \mathbb{R}^{3\times 1}$ is the position in meters [m] of an *Object* with respect to the world coordinate frame (wcf) denoted by O_b. O_{C_l} and O_{C_r} are the coordinate frames for the left and right cameras, respectively. $R^b_{C_l} \in SO(3)$ represents the orientation of wcf with respect to the left camera. O_V is a reference coordinate frame for the virtual orthogonal cameras $O_{v_{1,2}}$ where $R^{C_l}_V \in SO(3)$ is its orientation with respect to O_{C_l}. λ is the distance from O_{v_i} to O_V along each optical axis i. 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, $p_{v_i} \in \mathbb{R}^{2\times 1}$ represents the projection of the *Object* in the virtual cameras O_{v_i}.

A. Problem Statement

In the classical IBVS methods, the sensory feedback signals are directly chosen as the image feature measurements s. The vector $X_b \in \mathbb{R}^{3\times 1}$ represents the position vector of a target object in the task space, defined in the world coordinate frame (wcf). The vector $s = [x_1, y_1, ..., x_p, y_p]^T \in \mathbb{R}^{2p\times 1}$

contains the image feature measurements of all p feature points on the target object. Then, the relation between $\dot{s}$ and $\dot{X}_b$ is given by $\dot{s} = J_{img}\dot{X}_b$, where $J_{img} \in \mathbb{R}^{2p\times 3}$ is known as the image Jacobian.

If we consider ΔX_b as the input to a robot controller, then we need to compute the inverse mapping of $\dot{s}$ as $\Delta X_b = J^{+}_{img}\Delta s$, where $\Delta *$ is an error function defined in the space $*$, $J^{+}_{img} \in \mathbb{R}^{3\times 2p}$ is chosen as the Moore-Penrose pseudoinverse of J_{img}, which leads to the two characteristic problems of the IBVS method: the feature (image) space singularities and local minima. In this case, the image Jacobian is singular when $rank(J_{img}) < 3$, while the image local minima are defined as the set of image locations $\Omega_s = \left\{ s | \Delta s = 0, \Delta X_b \neq 0, \forall s \in \mathbb{R}^{2p\times 1} \right\}$ when using redundant image features.

In this work, we define a new type of visual features such that a full-rank *image Jacobian Matrix* ($J_{img} \in \mathbb{R}^{3\times 3}$) can be obtained. The key idea of this model is to combine the stereo camera model with a virtual composite camera model to get a full-rank image Jacobian to map velocities of a target object ($\dot{X}_{C_l}$) to velocities of the image features (in our case, pixel velocities in the 3D visual space, denoted here by $\dot{X}_s$), see Fig. 2.

This new 3D visual model can be computed in 3 main steps.

1) Use the standard stereo vision model [21] to recover the 3D *relative* position of an object $X_{C_l} \in \mathbb{R}^{3\times 1}$ with respect to the frame O_{C_l}[1].
2) The vector X_{C_l} is projected on two virtual cameras O_{v_i}. This projection is a crucial step, since it modifies the dimension of the mapping from two 2D-image feature measurements of all p visual points to a single 3D visual vector X_s in a *3D Visual Cartesian Space*. Since s represents the position $X_{C_l} \in \mathbb{R}^{3\times 1}$ in the image feature space, the maximum dimension of s is 3. Therefore, if $s \in \mathbb{R}^{4p\times 1}$ (as is commonly defined in the classical methods for stereo arrangements) there will be $4p-3$ linearly dependent elements in s. In this work, we propose a virtual projection that reduces the dimension of s and generates 3 linearly independent elements to compute a full-rank image Jacobian (J_{img}).
3) The final step is to transform the visual space velocities ($\dot{X}_s$) to joint velocity space ($\dot{q}$) via the image Jacobian (J_{img}). This is achieved using the robot Jacobian $J(q)$ ($J_s = J_{img}J(q)$).

The following sub-sections are devoted to explain each of these steps in detail.

B. Stereo Vision Model

The relation between right camera and left camera is given by the orientation matrix $R^l_r \in SO(3)$ and the translation vector $t^l_r \in \mathbb{R}^{3\times 1}$. Using these transformations we can define the projection matrices P_{C_l} and $P_{C_r} \in \mathbb{R}^{3\times 4}$ for each camera as $P_{C_l} = K_l \left[\begin{array}{cc} I_{3\times 3} & 0_{3\times 1} \end{array} \right]$ and $P_{C_r} = K_r \left[\begin{array}{cc} R^l_r & t^l_r \end{array} \right]$, where K_l and K_r are the intrinsic camera matrices of the left and right cameras[2].

Then, defining the observed image points in each camera as $p_l = [x_1 \quad y_1]^T$, $p_r = [x_2 \quad y_2]^T$, we can use triangulation [21] to compute the *relative* position $X_{C_l} = [x_c \quad y_c \quad z_c]^T$ of the point X_b with respect to the left camera O_{C_l}. This can be done by solving the system $A \left[\begin{array}{c} X_{C_l} \\ 1 \end{array} \right] = 0$, where

$$A = \left[\begin{array}{cccc} x_1 p^l_{31} - p^l_{11} & x_1 p^l_{32} - p^l_{12} & x_1 p^l_{33} - p^l_{13} & x_1 p^l_{34} - p^l_{14} \\ y_1 p^l_{31} - p^l_{21} & y_1 p^l_{32} - p^l_{22} & y_1 p^l_{33} - p^l_{23} & y_1 p^l_{34} - p^l_{24} \\ x_2 p^r_{31} - p^r_{11} & x_2 p^r_{32} - p^r_{12} & x_2 p^r_{33} - p^r_{13} & x_2 p^r_{34} - p^r_{14} \\ y_2 p^r_{31} - p^r_{21} & y_2 p^r_{32} - p^r_{22} & y_2 p^r_{33} - p^r_{23} & y_2 p^r_{34} - p^r_{14} \end{array} \right] \quad (1)$$

where, p^l_{ij} denotes the element (i,j) in P_{C_l}, and p^r_{ij} denotes the element (i,j) in P_{C_r}.

Before integrating the stereo camera model with the virtual composite model, a re-orientation of the coordinate frame O_{C_l} is required. This is achieved by defining a new coordinate frame O_V with the same origin as O_{C_l}. The projection $X_V = [x_V \quad y_V \quad z_V]^T$ of X_{C_l} in O_V is defined as (see Fig. 2)

$$X_V = R^{C_l}_V X_{C_l}. \quad (2)$$

where $R^{C_l}_V$ is the rotation matrix from O_{C_l} to the frame[3] O_V.

C. Virtual Composite Camera Model

In order to get the 3D visual space, we define two virtual cameras attached to the stereo camera system using the coordinate frame O_V (see Fig. 2). We use the pinhole camera model [21] to project the relative position X_V to each of the virtual cameras O_{v_1} and O_{v_2}.

The model for the virtual camera 1 is given by

$$p_{v_1} = \left[\begin{array}{c} x_{v_1} \\ y_{v_1} \end{array} \right] = \frac{1}{-y_V + \lambda} \alpha R(\theta) \left[\begin{array}{c} x_V - o_{11} \\ z_V - o_{12} \end{array} \right] + \left[\begin{array}{c} c_x \\ c_y \end{array} \right] \quad (3)$$

where θ is the rotation angle of the virtual camera along its optical axis, $O_1 = [o_{11} \quad o_{12}]^T$ is the position of the optical center with respect to the coordinate frame O_V, $C_1 = [c_x \quad c_y]^T$ is the position of the principal point in the image plane, λ is the distance from the virtual camera coordinate frame O_{v_1} to the reference frame O_V along its optical axis, α and the rotation matrix $R(\theta)$ are defined as:

$$\alpha = \left[\begin{array}{cc} f\beta & 0 \\ 0 & f\beta \end{array} \right] \qquad R(\theta) = \left[\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array} \right] \quad (4)$$

Since this model represents a user-defined virtual camera, all its parameters (extrinsic and intrinsic[4]) are known, in fact, in the defined configuration of the virtual cameras $\theta = 0$[5] (see Fig. 2).

[1] This step could be considered as an intermediate mapping between the stereo image space and the virtual composite image space.

[2] These parameters can be computed off-line.

[3] This reference frame is fixed and defined by the user and hence assumed to be known.

[4] Since the virtual cameras are user-defined, we can set the same intrinsic parameters and λ values for both cameras.

[5] The reason to introduce the auxiliary coordinate frame O_V is to simplify the composite camera model by rotating the coordinate frame O_{C_l} in an specific orientation, such as $\theta = 0$.

Similarly, the virtual camera 2 is defined as:

$$p_{v_2} = \begin{bmatrix} x_{v_2} \\ y_{v_2} \end{bmatrix} = \frac{1}{x_V + \lambda} \alpha R(\theta) \begin{bmatrix} y_V - o_{21} \\ z_V - o_{22} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (5)$$

In order to construct the 3D visual *Cartesian* space $X_s \in \mathbb{R}^{3\times 1}$, we combine both virtual camera models as follows.

Using the properties of the rotation matrix $R(\theta)$ and the fact that α is a diagonal matrix, from (3), x_{v_1} can be written in the form

$$x_{v_1} = \gamma_1 \frac{x_V - o_{11}}{-y_V + \lambda} - \gamma_2 y_{v_1} + \gamma_3, \quad (6)$$

where the constant parameters γ_1, γ_2, $\gamma_3 \in \mathbb{R}$ are explicitly defined as

$$\gamma_1 = \frac{f\beta}{\cos\theta}, \quad \gamma_2 = \tan(\theta), \text{ and } \gamma_3 = c_x + c_y\gamma_2. \quad (7)$$

Based on (5) and (6), we define the *3D Visual Camera Model* representation $X_s = [x_s \quad y_s \quad z_s]^T$ using the orthogonal elements $[x_{v_1} \quad x_{v_2} \quad y_{v_2}]^T$

$$X_s = \begin{bmatrix} x_{v_1} \\ x_{v_2} \\ y_{v_2} \end{bmatrix} = \overbrace{\begin{bmatrix} \gamma_1 & 0_{1\times 2} \\ 0_{2\times 1} & \alpha R(\theta) \end{bmatrix}}^{R_\alpha} \begin{bmatrix} \frac{x_V - 0_{11}}{-y_V + \lambda} \\ \frac{y_V - 0_{21}}{x_V + \lambda} \\ \frac{z_V - 0_{22}}{x_V + \lambda} \end{bmatrix} + \rho \quad (8)$$

where $\rho = [\gamma_3 - \gamma_2 y_{v_1} \quad c_x \quad c_y]^T$.

D. Visual Jacobian

In the previous section, we defined that the position of a point X_V projected in the *3D Visual Space* is given by X_s[6] as (8). The *Optical Flow* can be obtained with the time derivative of (8) as follows[7]:

$$\dot{X}_s = \overbrace{R_\alpha J_v}^{J_\alpha} \dot{X}_V = J_\alpha \dot{X}_V \quad (9)$$

where the image Jacobian Matrix $J_v \in \mathbb{R}^{3\times 3}$ is defined as

$$J_v = \begin{bmatrix} \frac{1}{-y_V+\lambda} & \frac{x_V - o_{11}}{(-y_V+\lambda)^2} & 0 \\ -\frac{y_V - o_{21}}{(x_V+\lambda)^2} & \frac{1}{x_V+\lambda} & 0 \\ -\frac{z_V - o_{22}}{(x_V+\lambda)^2} & 0 & \frac{1}{x_V+\lambda} \end{bmatrix} \quad (10)$$

This image Jacobian Matrix J_v represents the mapping from velocities defined in the reference frame O_V to velocities (pixels/s) in the 3D visual space. In order to complete the 3D visual mapping we need to include the transformations from O_V to O_b. This transformation is given by the following equation (see Fig. 2)

$$X_V = R_V^{C_l}(R_{C_l}^b X_b + t_{C_l}^b) \quad (11)$$

where, $R_{C_l}^b$ and $t_{C_l}^b$ are the rotation matrix and the translation vector between frame O_{C_l} and O_b.

Taking the time derivative of (11) and substituting the robot *Differential Kinematics* $\dot{X}_b = J(q)\dot{q}$, equation (9) can be rewritten in the form

$$\dot{X}_s = J_\alpha R_V^{C_l} R_{C_l}^b \dot{X}_b = J_{img}\dot{X}_b \quad (12)$$

$$= J_{img} J(q)\dot{q} = J_s\dot{q} \quad (13)$$

where $J(q) \in \mathbb{R}^{3\times 3}$ is the analytic Jacobian matrix of the robot manipulator, and the image-based Jacobian matrix $J_s \in \mathbb{R}^{3\times 3}$ is defined as the *Visual Jacobian*.

Then the inverse differential kinematics that relates generalized joint velocities $\dot{q}$ and 3D visual velocities $\dot{X}_s$ is given by

$$\dot{q} = J_s{}^{-1}\dot{X}_s = J(q)^{-1} J_{img}{}^{-1}\dot{X}_s \quad (14)$$

Remark 1: Singularity-free J_{img}. From (9) and (12), we can see that $J_{img}{}^{-1} = R_{C_l}^{b}{}^{-1} R_V^{C_l}{}^{-1} J_v{}^{-1} R_\alpha^{-1}$. The matrices $R_{C_l}^b, R_V^{C_l} \in SO(3)$ are non-singular. $R_\alpha = diag(f\beta) \in \mathbb{R}^{3\times 3}$ consists of user-defined virtual camera parameters. Then, $det(J_v) = 0 \rightarrow det(J_{img}) = 0$. This condition is present only when: 1) $O_{11} + \lambda = 0$ and $O_{21} - \lambda = 0$ or 2) $x_V = -\lambda$ and $y_V = O_{21}$ or 3) $x_V = O_{11}$ and $y_V = -\lambda$. However, O_{11}, O_{21} and λ are also defined by the user. Then, a non-singular J_{img} can be obtained using the condition $O_{11} = O_{21} > \lambda > \max(x_{V_{\max}}, y_{V_{\max}})$, where $x_{V_{\max}}$ and $y_{V_{\max}}$ are delimited by the robot workspace defined wrt O_V.

Therefore, the singularities of J_s are defined only by the singularities of $J(q)$. Hence, to guarantee a non-singular 3D visual mapping an approach to avoid robot singularities must be implemented. In Section IV, we discuss this issue and propose a solution.

Remark 2: Sensitivity to Camera Orientation $R_{C_l}^b$. The orientation matrix $R_{C_l}^b$ requires a special attention because can cause system instability. Instead of demanding an exact off-line calibration of this parameter, the problem is tackled in two parts: a) A coarse on-line estimation of the orientation matrix is computed using the real-time information generated by the robot (see Section II-E) and b) Estimation errors for the complete Jacobian J_s are taken into account in the control design. Thus, a robust control approach is designed to cope with these errors (including $R_{C_l}^b$ and the stereo-rig intrinsic parameters), see Section III-C.

E. On-line Orientation Matrix Estimation

We define this system as uncalibrated because we not only assume that the calibration of the stereo vision system (left camera O_{C_l}) with respect to the wcf (O_b) is unknown, but also consider the possibility of *on-line modification* of the parameters that define this relationship (e.g. $R_{C_l}^b$). The stereo-rig is assumed to be known, which is not a strong constraint because the entire stereo system is referenced with respect to the left camera (see Fig. 2), and this can be done off-line. Nevertheless, an exact calibration of the stereo system is also not required because errors in estimation of these visual parameters are handled in the control design (see Section III-C).

In order to compute the visual Jacobian, in this work we use an on-line orientation matrix estimator, where two sets of

[6] In this work, we use X_s instead of the classical notation s because X_s is more than a image feature measurement, in fact, it defines a position vector in the 3D visual space.

[7] Given that $\theta = 0$, then $\gamma_1 = f\beta$, $\gamma_2 = 0$, $\rho = [c_x \quad c_x \quad c_y]^T$ and $R_\alpha = diag(f\beta) \in \mathbb{R}^{3\times 3}$.

position points defined in each coordinate frame O_b and O_{C_l} are used. These sets are generated while the robot is moving. The estimation approach can be summarized as follows.

Let datasets A and B be the sets of end-effector positions defined with respect to O_b and O_{C_l}, respectively. The problem is to find the best rotation $R^b_{C_l}$ that will align the points in dataset A with the points in dataset B. This rotation matrix can be obtained by computing the matrix

$$M = \sum_{i=1}^{n} (A_i - C_A)(B_i - C_B)^T \tag{15}$$

where $C_{A,B}$ is the centroid of the data set (A, B). Then a least-squares fit of the rotation matrix can be written as $R^b_{C_l} = U diag\left(1, 1, det\left(UV^T\right)\right) V^T$, where U,V are computed from $SVD(M)$. This rotation matrix is prone to errors of estimation which are considered in the next section.

III. Adaptive Image-based 3D Visual Servoing

In this section we describe the design of an adaptive image-based dynamic control. The proposed second order sliding mode control is chattering-free. This result is obtained using the function $\int sign(*)$ instead of the function $sign(*)$. Furthermore, the function $sign(*)$ is also replaced by $tanh(\mu *)$ to avoid discontinuous signals for the nominal reference, similar to [22].

A. Non Linear Robot Dynamic Model

The dynamics of a serial n-link rigid, non-redundant, fully actuated robot manipulator can be written as follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + B\dot{q} = \tau. \tag{16}$$

where $q \in \mathbb{R}^{n\times 1}$ is the vector of joint positions, $\tau \in \mathbb{R}^{n\times 1}$ stands for the applied joint torques, $M(q) \in \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix, $C(q,\dot{q})\dot{q} \in \mathbb{R}^{n\times n}$ is the vector of centripetal and Coriolis effects, $G(q) \in \mathbb{R}^{n\times 1}$ is the vector of gravitational torques, and finally $B \in \mathbb{R}^{n\times n}$ is a diagonal matrix for the viscous frictions.

The robot model described in (16) can be written in terms of a known state robot regressor $Y = Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n\times m}$ and an unknown robot parameter vector $\Theta \in \mathbb{R}^{m\times 1}$ by using nominal references $\dot{q}_r$ and $\ddot{q}_r$ as follows:

$$M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + B\dot{q}_r = Y_r\Theta \tag{17}$$

Subtracting the linear parameterization equation (17) to (16), produces the open-loop error dynamics

$$M(q)\dot{S}_q + C(q,\dot{q})S_q = \tau - Y_r\Theta \tag{18}$$

with the *joint error surface* S_q defined as $S_q = \dot{q} - \dot{q}_r$, where $\dot{q}_r$ represents the nominal reference of joint velocities. This nominal reference can be used to design a control in the 3D visual space.

B. Joint Velocity Nominal Reference

Considering equation (14), $\dot{q}_r$ can be defined as

$$\dot{q}_r = {J_s}^{-1}\dot{X}_{s_r} \tag{19}$$

where, the 3D visual nominal reference $\dot{X}_{s_r}$ is given by

$$\begin{aligned}\dot{X}_{s_r} = \Bigg(\dot{X}_{s_d} - K_p\Delta X_s + S_{s_d} - K_1\int_{t_0}^{t} S_{s_\delta}(\zeta)\, d\zeta \\ - K_2\int_{t_0}^{t} sign\left(S_{s_\delta}(\zeta)\right) d\zeta\Bigg)\end{aligned} \tag{20}$$

$$S_{s_\delta} = S_s - S_{s_d}, S_s = \left(\Delta\dot{X}_s + K_p\Delta X_s\right), S_{s_d} = S_s(t_0)\, e^{-\kappa t} \tag{21}$$

where $\dot{X}_{s_d}$ is the desired visual velocity; $\Delta X_s = X_s - X_{s_d}$ is the visual position error, $\Delta\dot{X}_s$ is the visual velocity error, $K_p = {K_p}^T \in \mathbb{R}^{3\times 3}_{+}$ and $K_j = {K_j}^T \in \mathbb{R}^{3\times 3}_{+}$ (with $j = 1, 2$) and $S_{s\delta}$ is the *visual error surface*.

Using (19-21) in S_q we obtain:

$$S_q = \dot{q} - \dot{q}_r = {J_s}^{-1}\left(\dot{X}_s - \dot{X}_{s_r}\right) = {J_s}^{-1} S_e \tag{22}$$

with

$$S_e = S_{s_\delta} + K_1\int_{t_0}^{t} S_{s_\delta}(\zeta)\, d\zeta + K_2\int_{t_0}^{t} sign\left(S_{s_\delta}(\zeta)\right) d\zeta \tag{23}$$

where S_e is the *extended visual error manifold.*

C. Uncertainties in J_s

The above definition of $\dot{q}_r$ depends on the exact calibration of J_s. However, this is a very restricted assumption. Hence, the uncertainties in the Visual Jacobian J_s should be taken into account in the control design. To achieve this, the uncalibrated nominal reference is defined by

$$\widehat{\dot{q}_r} = \widehat{J_s}^{-1}\dot{X}_{s_r} \tag{24}$$

where $\widehat{J_s}$ is an estimate of J_s such that $\widehat{J_s}$ is full-rank $\forall q \in \Omega_q$, and $\Omega_q = \left\{q | \det(J(q)) \neq 0, \forall q \in \mathbb{R}^{n\times 1}\right\}$ defines the singularity free workspace. Then, the uncalibrated joint error surface is:

$$\hat{S}_q = \dot{q} - \widehat{\dot{q}_r} = S_q - \Delta J_s \dot{X}_{s_r} \tag{25}$$

with $\Delta J_s = \widehat{J_s}^{-1} - {J_s}^{-1}$ as the estimation errors, which includes both intrinsic and extrinsic parameters.

D. Control Design

Consider a robot manipulator in closed loop with the following second order sliding visual servoing scheme,

$$\tau = -K_d\hat{S}_q + \hat{Y}_r\hat{\Theta} \tag{26}$$

$$\dot{\hat{\Theta}} = -\Gamma{\hat{Y}_r}^T\hat{S}_q \tag{27}$$

where $\hat{\Theta}$ is the on-line estimation of the constant robot parameter vector, $K_d = {K_d}^T \in \mathbb{R}^{n\times n}_{+}$ and $\Gamma \in \mathbb{R}^{m\times m}_{+}$ are constant matrices. This adaptive on-line estimation together with the second order sliding mode in S_{s_δ} handle the uncertainties on the robot dynamic/kinematic and camera parameters.

E. Stability Proof

The stability proof is conducted in three parts:

1) Boundedness of the closed loop trajectories: The uncalibrated closed-loop error dynamics between (18) and (26-27) gives

$$M(q)\dot{\hat{S}}_q = \tau - \hat{Y}_r\Theta - C(q,\dot{q})\hat{S}_q \tag{28}$$

$$= -K_d\hat{S}_q + \hat{Y}_r\Delta\Theta - C(q,\dot{q})\hat{S}_q \tag{29}$$

with $\Delta\Theta = \hat{\Theta} - \Theta$.

The uncalibrated error kinematic energy can be used as a Lyapunov function in the following form as:

$$V = \frac{1}{2}\left[\hat{S}_q{}^T M(q)\hat{S}_q + \Delta\Theta^T\Gamma^{-1}\Delta\Theta\right] \tag{30}$$

Considering the time derivative of (30) in closed loop with (26-29), $\dot{V}$ yields

$$\dot{V} = \hat{S}_q{}^T M\dot{\hat{S}}_q + \frac{1}{2}\hat{S}_q{}^T\dot{M}\hat{S}_q + \Delta\Theta^T\Gamma^{-1}\dot{\hat{\Theta}} \tag{31}$$

$$= -\hat{S}_q{}^T K_d\hat{S}_q + \hat{S}_q{}^T\hat{Y}_r\Delta\Theta - \Delta\Theta^T\hat{Y}_r\hat{S}_q = -K_d\left\|\hat{S}_q\right\| \tag{32}$$

where, the property (17) in terms of $\hat{S}_q$ has been used.

Selecting a positive K_d, equation (32) becomes negative definite and this proves the passivity of the robot dynamics (16) in closed loop with (26-27). Then, the following properties of the closed-loop state arises

$$\hat{S}_q \in L_\infty \to S_e \in L_\infty \Longrightarrow \left(S_{s_\delta}, \int_{t_0}^{t} sign\left(S_{s_\delta}(\zeta)\right)d\zeta\right) \in L_\infty \tag{33}$$

Which implies that all the signal states are bounded, specially $(\dot{q}_r, \ddot{q}_r) \in L_\infty$ and $\left(\dot{X}_{s_r}, \ddot{X}_{s_r}\right) \in L_\infty$.

2) Second-order sliding modes: From (22) and (25) we obtain

$$\hat{S}_q = J_s{}^{-1}S_e - \Delta J_s\dot{X}_{s_r} \tag{34}$$

Using (23), (34) can be written as

$$S_{s_\delta} = -K_1\int_{t_0}^{t} S_{s_\delta}(\zeta)\,d\zeta - K_2\int_{t_0}^{t} sign\left(S_{s_\delta}(\zeta)\right)d\zeta + \Psi \tag{35}$$

with $\Psi = J_s\left(\hat{S}_q + \Delta J_s\dot{X}_{s_r}\right)$,

Taking the time derivative of (35) and multiplying it by $S_{s_\delta}^T$, we can prove the sliding mode regimen

$$S_{s_\delta}^T\dot{S}_{s_\delta} \leq -K_1\left\|S_{s_\delta}\right\| - \mu\left|S_{s_\delta}\right| \tag{36}$$

with $\mu = K_2 - \left|\frac{d}{dt}\Psi\right|$. If $K_2 \geq \left|\frac{d}{dt}\Psi\right|$, then a sliding mode at $S_{v_\delta} = 0$ is induced at $t_s = \frac{\left|S_{s_\delta}(t_0)\right|}{\mu}$. Moreover, notice that for any initial condition $S_{s_\delta}(t_0) = 0$ then $t_s = 0$, which implies that the sliding mode is guaranteed for all time.

3) Exponential convergence of visual tracking errors: Since a sliding mode exists at all times at $S_{s_\delta}(t) = 0$, then $S_s = S_{s_d}$, therefore $\Delta\dot{X}_s = -K_p\Delta X_s + S_s(t_0)e^{-\kappa t}$ $\forall t$, which implies that the 3D visual tracking errors converge to zero exponentially fast.

Remark 3: Convergence of ΔX_b without local minima. Given that J_{img} is full-rank $\forall t$, from (12) can be seen that $\Delta X_s = 0 \to \Delta X_b = 0$ without local minima. This is the most important impact of designing a full-rank image Jacobian which, in general, is not obtained with the classical methods.

F. Simulation

The torque level adaptive control is evaluated in simulation to better illustrate the robustness of the system to uncertainties in the robot parameters and changes in the extrinsic parameters of the stereo system.

In this case, we simulate a 6DOF industrial robot where the last three joints are controlled in a static position with a PID-like controller. The real robot dynamic parameters were used to simulate the robot in closed loop with the control approach in (26-27). Also real camera parameters were used to simulate the camera projections.

The task is defined as follows: the robot end-effector is commanded to draw a circle in the world coordinate frame using the trajectory $X_{b_d} = (0.2\sin(\omega t), 0.2\cos(\omega t) - 0.8, 0.5)$, where $\omega = 10 rad/sec$. Simulations are carried out in Matlab 2010a.

Fig. 3 shows the results obtained from the simulation, where the 3D visual position/velocity tracking can be observed. During the simulation, an estimate of robot parameters and a coarse-calibrated camera intrinsic parameters are used in the control law, and at time $t = 4s$, $t = 7s$, the extrinsic camera parameters are altered, see Fig. 3. From the plots it can be observed that even when the parameters change, the controller is capable to cope with these uncertainties and maintain stability of the system. In the case of changes in the orientation matrix $R^b_{C_l}$, the controller can handle the uncertainty to a certain extent (approx. 20% error). Therefore, a suitable technique to generate a rough estimation of $R^b_{C_l}$ is needed to guarantee the stability (see Section II-E).

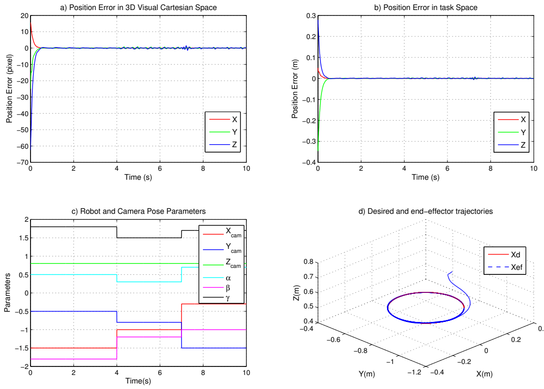

Fig. 3. Simulation results: The exponential convergence of the tracking errors is depicted. It can be seen that the trajectories are stable, even when the parameters change $t = 4s$ and $t = 7s$. a),b) The position errors in both task space (meter) and 3D visual *Cartesian* space (pixel) are shown. c) The parameters of the stereo camera pose are illustrated, notice how they change in time. d) is the tracking trajectory in task space.

IV. 3D Visual Servoing with Environment Constraints

In the real experiment, we integrate the visual servoing system in a HRI scenario, where enviroment constraints, such as: robot singularites avoidance, (self-/obstacle) collision avoidance must be included to generate a safe and

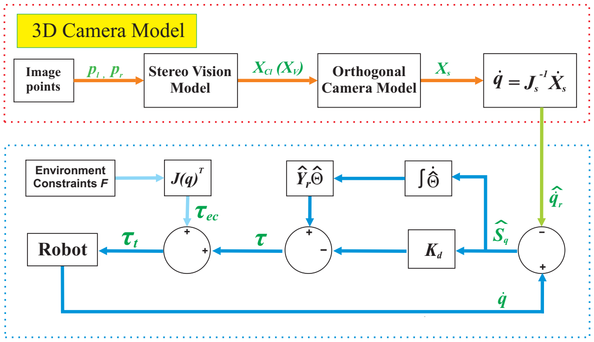

Fig. 4. 3D adaptive visual servoing including enviroment constraints

singularity-free trajectory for the robot. In this work, we modeled 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. 4 shows the integration of the control with the environment constraints.

1) Singularity Force: An example of how to compute this force is given by $F_r = -K_r P_r D_r - B_r \dot{X}_{ef}$, where $\dot{X}_{ef}$ is the velocity of the end-effector, $K_r, B_r \in \mathbb{R}^{3\times 3}$ are constant matrices, and we define Δq as the absolute value of the difference between q_i and $q_{singularity}$, D_r is the direction of gradient for the maximum manipulability factor μ. Then, $P_r = e^{\alpha_r \Delta q} - 1$, where α_r is a constant to control the stiffness of the applied force.

2) Collision Avoidance Forces: We use the artificial potential field approach to compute the forces for collision avoidance problem, where the obstacles are repulsive surfaces for the manipulator. This force is given by $F_o = -K_o P_o D_o - B_o \dot{X}_{ef}$, with $P_o = e^{\alpha_o(Dist_o - h_o)} - 1$, where h_s is the minimum distance between the obstacle and robot arm on the xoy plane, and D_o is the direction of the force, obtained between the center of the obstacle and the end-effector.

V. Experiments

A. System Review

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.

1) Visual Stereo Tracker: The stereo system is composed of 2 USB cameras fixed on a camera tripod. The stereo rig is uncalibrated with respect to the wcf and can be moved. The parameters of the virtual cameras (see Section II) are selected such that J_α is always non-singular. In order to compute τ and avoid multiple-sampling system, an extended Kalman filter (EKF) is used to estimate the visual position (sampling period $1ms$), where the reference is updated each $30ms$ with the real visual data of both cameras.

2) Robot Control System: The robot system comprises of a StaübliTX90 industrial robot arm (6DOF), a CS8C control unit and a Workstation running on GNU/Linux OS with real-time patch, see Fig. 1. The data communication between the PC and the control unit is in a local network based on TCP/IP. In this article, the robot is controlled in torque mode using a Low Level Interface (LLI) library.

3) 3D Visualization System: This module performs OpenGL based real-time rendering of the workspace in 3D. It uses Qt and Coin3D as its backbone. The system updates the configuration of the robot arm and the positions of the target in real-time. This is achieved by means of TCP/IP communication.

B. Experiment Results

In this case we use a simple color-based visual tracker to identify the target (green) and the robot end-effector (red). The target is held by a human, and the control goal is to follow the object in the human hand with the end-effector. The stereo vision tracking system provides the positions of both red and green cubes with respect to the stereo coordinate frame. This information is mapped to the 3D visual Cartesian space to compute the errors. Using the adaptive control a dynamic collision/singularity free trajectory can be obtained.

During the experimental validation, several behaviors are evaluated. These behaviors are depicted in Fig. 5 and Fig. 6.

Fig. 5 a) shows how the robot handles self-collisions, the system generates a collision-free trajectory (red line) instead of moving directly to the target. Fig. 5 b) demonstrates the obstacle avoidance. The blue line is the target motion and the red line is the robot end-effector motion. In the center, there is a static obstacle. The robot can avoid it while continuing to track the target.

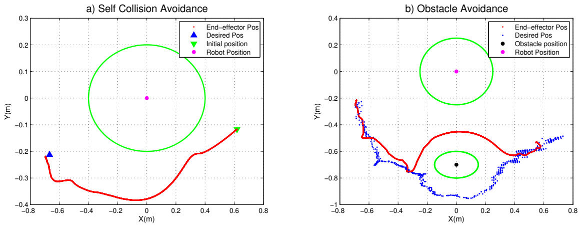

Fig. 5. Experiment results: a) Self-collision avoidance. b) Obstacles avoidance.

Fig. 6 demonstrates the visual servoing control in combination with different forces. Fig. 6 f) and a) show visual tracking with and without obstacle avoidance, respectively. Fig. 6 c) illustrates the result of singularity avoidance, the robot does not reach the singular condition ($q_3 = 0$), even when the user tries to force it. Fig. 6 e) shows self collision avoidance. Fig. 6 d) depicts table avoidance where the motion of the robot in the $z-axis$ is constrained by the height of the table (the end-effector is not allow to go under the table).

The primary advantage of our system is when the target object is occluded, the stereo system can be moved to maintain the targets in the field of view, see Fig. 6 h). After moving the stereo system, our approach can use real data to estimate the orientation matrix on-line. The control absorbs the perturbations and maintains the overall stability.

The system proves to be stable and safe for HRI scenarios, even in situations where the target is lost (due to occlusions by the robot or the human), see Fig. 6 b) and g). In this case, the robot system disables the contribution of τ (visual servoing contribution) and reacts only to collision and singularity forces. The visual tracking is resumed as soon as the target is visible again. A video where these behaviors are illustrated can be downloaded under: http://youtu.be/INwI2pDWYYo

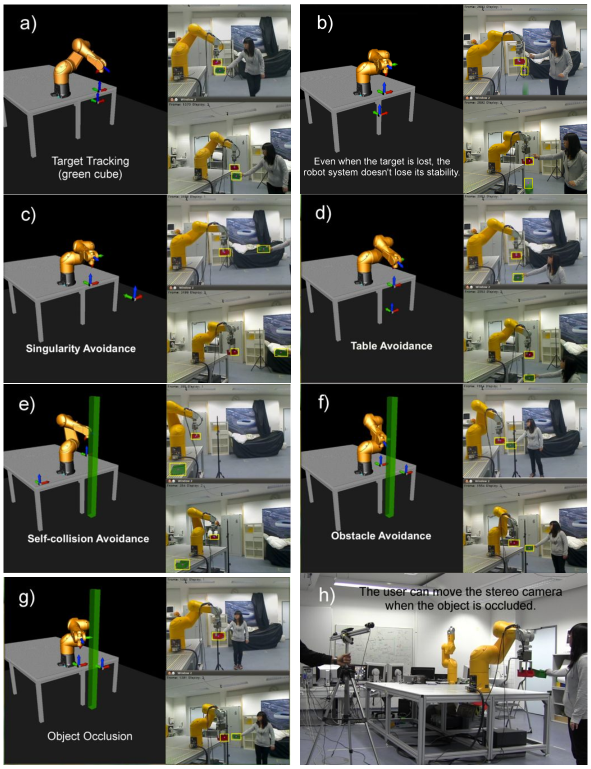
a)
Target Tracking
(green cube)
b)
Even when the target is lost, the robot system doesn't lose its stability.
c)
Singularity Avoidance
d)
Table Avoidance
e)
Self-collision Avoidance
f)
Obstacle Avoidance
g)
Object Occlusion
h)
The user can move the stereo camera when the object is occluded.

Fig. 6. System behaviors: a) The control uses the information of both cameras to track the object. b) The robots stops when the target is lost until is visible again. c) Singularity avoidance. d) Table avoidance. e) Self-collision avoidance. f) Obstacle avoidance. g) When the object is occluded, then robot changes its behavior. h) The camera can be moved when the target is occluded. The orientation matrix is estimated on-line.

VI. Conclusions

We proposed a novel image-based controller for 3D image-based visual servoing using uncalibrated stereo vision system. The control was evaluated both in simulation and on a real industrial robot. The obtained results show the stability of the control even with parametric uncertainties. Furthermore, this work extends the adaptive image-based control law to include enviroment constraints. As a result, information about the environment and the kinematic constraints can be integrated with the 3D visual servoing to generate a robot dynamic system with trajectory free of collisions and singularities. This approach was evaluated in a real HRI scenario. The future work is to include an advanced object tracker to improve the performance, compare the proposed technique with standard approaches to verify its efficiency. Also, the extension to 6D (visually control the position and orientation of the end-effector using the visual Cartesian space) is being analyzed.

References

[1] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, 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] 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.

[4] M. Marey and F. Chaumette, “Analysis of classical and new visual servoing control laws,” in *IEEE Int. Conf. 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 Trans. 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 Int. 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,” *Int. 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 Trans. on Robotics*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

[10] L. Pari, J. Sebastin, 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.

[11] S. Azad, Farahmand, Amir-Massoud, and M. Jagersand, “Robust jacobian estimation for uncalibrated visual servoing,” in *Robotics and Automation (ICRA), 2010 IEEE Int. Conf. on*, May 2010, pp. 5564–5569.

[12] E. Cervera, A. P. D. Pobil, F. Berry, and P. Martinet, “Improving image-based visual servoing with three-dimensional features.” *The Int. Journal of Robotics Research*, vol. 22, no. 10-11, pp. 821–840, 2003.

[13] K. Hashimoto, T. Ebine, and H. Kimura, “Dynamic Visual Feedback Control For A Hand-eye Manipulator,” in *Proceedings of the 1992 lEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, July 1992, pp. 1863–1868.

[14] E. Zergeroglu, D. Dawson, M. de Queiroz, and S. Nagarkatti, “Robust visual-servo control of robot manipulators in the presence of uncertainty,” in *IEEE Conference on Decision and Control*, vol. 4, 1999, pp. 4137–4142.

[15] A. Astolfi, L. Hsu, M. Netto, and R. Ortega, “Two solutions to the adaptive visual servoing problem,” *IEEE Trans. on Robotics and Automation*, vol. 18, no. 3, pp. 387–392, June 2002.

[16] B. Bishop, S. Hutchinson, and M. Spong, “Camera modelling for visual servo control applications,” *Math. Comput. Model.*, vol. 24, no. 5-6, pp. 79–102, Sep. 1996.

[17] L. Hsu and P. Aquino, “Adaptive visual tracking with uncertain manipulator dynamics and uncalibrated camera,” in *IEEE Conference on Decision and Control*, vol. 2, 1999, pp. 1248–1253.

[18] 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 Int. Conf. on Industrial Informatics*, June 2004, pp. 339–345.

[19] R. Kelly, “Robust asymptotically stable visual servoing of planar robots,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 759–766, Oct. 1996.

[20] H. Wang, Y.-H. Liu, and W. Chen, “Uncalibrated Visual Tracking Control Without Visual Velocity,” *IEEE Trans. on Control Systems Technology*, vol. 18, no. 6, pp. 1359–1370, Nov. 2010.

[21] A. Harltey and A. Zisserman, *Multiple View Geometry in Computer Vision (2. ed.).* Cambridge University Press.

[22] E. Dean-León, V. Parra-Vega, A. Espinosa-Romero, and L. García, “Visual servoing for constrained robots: A new complete theoreticalframework and its experimental validation,” in *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, Jul 2005.