

3D Image-based Dynamic Visual Servoing with uncalibrated Stereo Cameras

Caixia Cai¹, Emmanuel Dean-León², Nikhil Somani² and Alois Knoll¹

¹Technische Universität München, Fakultät für Informatik, Germany. Email: caica@in.tum.de

²Fortiss - An-Institut der Technischen Universität München, Germany.

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 pixels). 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 control is designed to track 3D visual motions using the 3D trajectory errors defined in the *Visual Cartesian Space*, where a *Torque to Position Model* is designed to allow the implementation of joint torque control techniques on joint position-controlled robots. This approach has been experimentally implemented on a real industrial robot where exponential convergence of errors in the *Visual Cartesian Space* and *Task space* without local minima are demonstrated. This approach offers a proper solution for the common problem of visual occlusion, since the stereo system can be moved manually to obtain a clear view of the task at any time.

Index Terms— 3D Image-based Visual Servoing, Adaptive Dynamic Control, Uncalibrated Stereo Cameras

I. INTRODUCTION

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

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. In this work: 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.

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. The classical image Jacobian is defined using a set of image feature measurements, denoted by s , and it describes how image features change when the robot manipulator pose changes $\dot{s} = J_{img}v$. This explicitly implies

the implementation of the image Jacobian in the control design. A full and comprehensive survey on Visual Servoing and image Jacobian definitions can be found in [1], [2] and more recently in [3].

In general, the image Jacobian can be computed using direct depth information [4],[5], or by approximation via on-line estimation of depth of the features [2], [3], [6], or using depth-independent image Jacobian matrix [7],[8],[9]. Additionally, many papers directly estimate on-line the complete image Jacobian in different ways [10],[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 matrix needs to be computed, which leads to well-known problems such as the image space singularities and local minima.

In the context of control schemes, kinematic-based controls 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 [14],[15]. In order to avoid tedious and costly camera calibration, [16], [17], and [18] 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 [8],[9]. 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. In order to evaluate the control approach and the proposed models, experiments have been conducted on a real industrial robot. We integrate the visual servoing system in a Human-Robot-Interaction (HRI) scenario (see Fig. 1), where we reproduce a more realistic environment including obstacles, robot singularities, visual occlusion, collision avoidance and sporadic loss of targets.

A. Organization

The organization of this article is as follows: First, in Section II we introduce a new *3D Camera Model* to derive a full-rank *3D Visual Jacobian*, which will be used in the Section III to design an adaptive image-based 3D visual

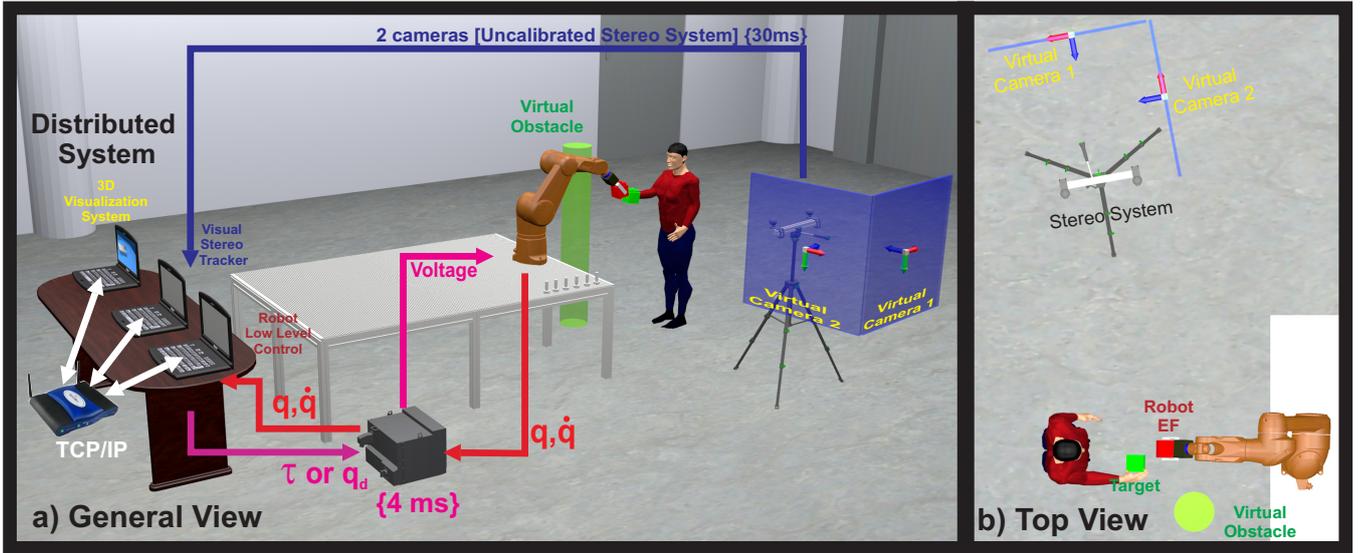


Fig. 1. Description of robotic experimental setup.

servoing. In Section IV we present the adaptive image-based torque controller which includes environment constraints. 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 CAMERA MODEL

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 $s = [x_1, y_1, \dots, x_p, y_p]^T \in \mathbb{R}^{2p \times 1}$ contains the image feature measurements of all p feature points on the target object. Then, the relation between 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 s as $\Delta X_b = J_{img}^+ \Delta s$, where Δ^* 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 $\Omega_s = \{s | \Delta s = 0, \Delta X_b \neq 0, \forall s \in \mathbb{R}^{2p \times 1}\}$ 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. The following sub-sections are devoted to explain 3 main steps to get this new 3D visual model in detail.

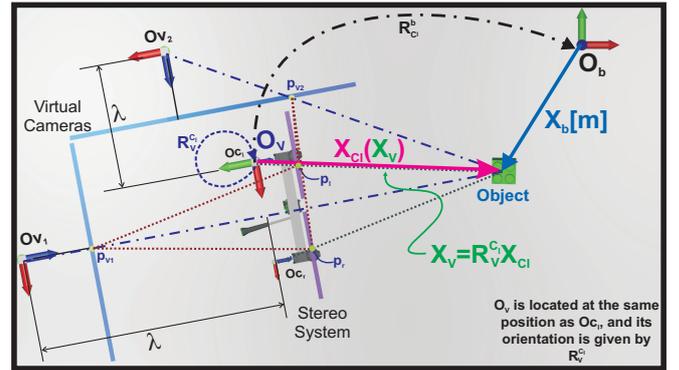


Fig. 2. Image Projections: The figure depicts the different coordinate frames used to obtain a general 3D visual camera model. $X_b \in \mathbb{R}^{3 \times 1}$ is the position in meters [m] of an *Object* with respect to the world coordinate frame (wcf) denoted by O_b . O_{C_l} and O_{C_r} are the coordinate frames for the left and right cameras, respectively. $R_{C_l}^b \in SO(3)$ 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 $R_V^C_l \in SO(3)$ 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} .

B. Stereo Vision Model

The stereo system is composed of 2 USB web cameras fixed on a camera tripod (Fig. 1). Given the observed image points in each camera, $p_l = [x_1, y_1]^T$, $p_r = [x_2, y_2]^T$, we can perform triangulation [19] using the relation between the right and left cameras (given by the orientation matrix R_l^r and the translation vector t_l^r) to compute the *relative* position $X_{C_l} = [x_c, y_c, z_c]^T$ of the point X_b with respect to the left camera O_{C_l} .

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

$$X_V = R_V^{C_l} X_{C_l}, \quad (1)$$

where $R_V^{C_l}$ is the orientation of the reference frame O_V ¹ w.r.t. O_{C_l} .

C. Virtual Composite Camera Model

To construct 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 [19] to project the relative position X_V to each of the virtual cameras O_{v_1} and O_{v_2} .

The model for the both virtual cameras are given by

$$p_{v_1} = \begin{bmatrix} x_{v_1} \\ y_{v_1} \end{bmatrix} = \frac{1}{\lambda - y_V} \alpha R(\theta) \begin{bmatrix} x_V - o_{11} \\ z_V - o_{12} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (2)$$

$$p_{v_2} = \begin{bmatrix} x_{v_2} \\ y_{v_2} \end{bmatrix} = \frac{1}{x_V + \lambda} \alpha R(\theta) \begin{bmatrix} y_V - o_{21} \\ z_V - o_{22} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (3)$$

where θ is the rotation angle of the virtual camera along its optical axis, $O_i = [o_{i1}, o_{i2}]^T$ is the position of the optical center with respect to the coordinate frame O_V , $C = [c_x, c_y]^T$ is the position of the principal point in the image plane, λ is the distance from the virtual camera coordinate frame O_{v_i} to the reference frame O_V along its optical axis, α and the rotation matrix $R(\theta)$ are defined as:

$$\alpha = \begin{bmatrix} f\beta & 0 \\ 0 & f\beta \end{bmatrix} \quad R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (4)$$

Since this model represents user-defined virtual cameras, all the parameters (extrinsic and intrinsic²) are known, in fact, in the defined configuration of the virtual cameras, $\theta = 0$ (see Fig. 2).

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 (2), x_{v_1} can be written in the form

$$x_{v_1} = \gamma_1 \frac{x_V - o_{11}}{-y_V + \lambda} - \gamma_2 y_{v_1} + \gamma_3, \quad (5)$$

where the constant parameters $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}$ are explicitly defined as $\gamma_1 = \frac{f\beta}{\cos \theta}$, $\gamma_2 = \tan(\theta)$, and $\gamma_3 = c_x + c_y \gamma_2$. Based on (3) and (5), we define the *3D Visual Camera Model* representation $X_s = [x_s, y_s, z_s]^T$ using the orthogonal elements $[x_{v_1}, x_{v_2}, y_{v_2}]^T$

$$X_s = \begin{bmatrix} x_{v_1} \\ x_{v_2} \\ y_{v_2} \end{bmatrix} = \overbrace{\begin{bmatrix} \gamma_1 & 0_{1 \times 2} \\ 0_{2 \times 1} & \alpha R(\theta) \end{bmatrix}}^{R_\alpha} \begin{bmatrix} \frac{x_V - o_{11}}{-y_V + \lambda} \\ \frac{y_V - o_{21}}{x_V + \lambda} \\ \frac{z_V - o_{22}}{x_V + \lambda} \end{bmatrix} + \rho \quad (6)$$

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

¹The 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 $\theta = 0$.

²We can set the same intrinsic parameters and λ values for both cameras.

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 ³ as (6). The *Optical Flow* can be obtained with the time derivative of (6) as follows⁴:

$$\dot{X}_s = R_\alpha J_v \dot{X}_V = J_\alpha \dot{X}_V \quad (7)$$

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

$$J_v = \begin{bmatrix} \frac{1}{-y_V + \lambda} & \frac{x_V - o_{11}}{(-y_V + \lambda)^2} & 0 \\ -\frac{y_V - o_{21}}{(x_V + \lambda)^2} & \frac{1}{x_V + \lambda} & 0 \\ -\frac{z_V - o_{22}}{(x_V + \lambda)^2} & 0 & \frac{1}{x_V + \lambda} \end{bmatrix} \quad (8)$$

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 (see Fig. 2) $X_V = R_V^{C_l} (R_{C_l}^b X_b + t_{C_l}^b)$ 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 X_V and substituting the robot *Differential Kinematics* $\dot{X}_b = J(q)\dot{q}$, equation (7) can be rewritten in the form

$$\dot{X}_s = J_\alpha R_V^{C_l} R_{C_l}^b \dot{X}_b = J_{img} \dot{X}_b = J_{img} J(q) \dot{q} = J_s \dot{q} \quad (9)$$

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

$$\dot{q} = J_s^{-1} \dot{X}_s = J(q)^{-1} J_{img}^{-1} \dot{X}_s \quad (10)$$

Remark 1: Singularity-free J_{img} .

From (9), $J_{img}^{-1} = R_{C_l}^{b-1} R_V^{C_l-1} J_v^{-1} R_\alpha^{-1}$. The matrices $R_{C_l}^b, R_V^{C_l} \in SO(3)$ are non-singular. $R_\alpha \in \mathbb{R}^{3 \times 3}$ and J_v consist of user-defined virtual camera parameters. Then, a non-singular J_{img} can be obtained by delimiting by the robot workspace.

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

Remark 2: Sensitivity to Camera Orientation $R_{C_l}^b$.

The orientation $R_{C_l}^b$ 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 V-C) 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, see Section III-C.

³In 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.

⁴Given that $\theta = 0$, then $\gamma_1 = f\beta$, $\gamma_2 = 0 \implies \rho = [c_x, c_x, c_y]^T$ and $R_\alpha = \text{diag}(f\beta) \in \mathbb{R}^{3 \times 3}$.

III. 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.

A. Non Linear Robot Dynamic Model

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

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + B\dot{q} = \tau. \quad (11)$$

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, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ 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.

Subtracting the linear parameterization equation to (11), produces the open-loop error dynamics

$$M(q)\dot{S}_q + C(q, \dot{q})S_q = \tau - Y_r \Theta \quad (12)$$

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 (10), \dot{q}_r can be defined as

$$\dot{q}_r = J_s^{-1} \dot{X}_{s_r} \quad (13)$$

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

$$\begin{aligned} \dot{X}_{s_r} = & \left(\dot{X}_{s_d} - K_p \Delta X_s + S_{s_d} - K_1 \int_{t_0}^t S_{s_\delta}(\zeta) d\zeta \right. \\ & \left. - K_2 \int_{t_0}^t \text{sign}(S_{s_\delta}(\zeta)) d\zeta \right) \end{aligned} \quad (14)$$

$$S_{s_\delta} = S_s - S_{s_d}, S_s = (\Delta \dot{X}_s + K_p \Delta X_s), S_{s_d} = S_s(t_0) e^{-\kappa t} \quad (15)$$

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_δ} is the *visual error surface*.

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

$$\hat{S}_q = \dot{q} - \hat{q}_r = S_q - \Delta J_s \dot{X}_{s_r} \quad (16)$$

with $\Delta J_s = \hat{J}_s^{-1} - J_s^{-1}$ 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,

$$\tau = -K_d \hat{S}_q + \hat{Y}_r \hat{\Theta}, \quad \dot{\hat{\Theta}} = -\Gamma \hat{Y}_r^T \hat{S}_q \quad (17)$$

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.

Exponential convergence of visual tracking errors:

Substituting (17) in the dynamic (12) using 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 (9) can be seen that $\Delta X_s = 0 \rightarrow \Delta X_b = 0$ 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.

The robot motion may lead to unsafe/unstable situations, e.g. the motion can take the robot to a singularity positions. This is critical specially in the case of Human-Robot-Interaction, where the safety of the user is the primary concern. To avoid this problem a dynamic trajectory planner is introduced in the following section.

IV. 3D VISUAL SERVOING WITH ENVIRONMENT CONSTRAINTS IN HRI SCENARIO

In the real experiment, we integrate the visual servoing system in a HRI 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. In this work, we use the *Impedance Control* schemes to connect different dynamic phenomena together. This means, the complete environment can be modeled as a single control input. External factors, such as: singularity avoidance, (self- and object-) collision avoidance and specially Human collision avoidance can be integrated in a control strategy (e.g. visual servoing). Fig. 3 shows the integration of the control with the environment constraints.

A. Environment Constraints

1) **Singularity Force:** An example of how to compute this force is given by

$$F_r = -K_r P_r D_r - B_r \dot{X}_{ef} \quad (18)$$

where \dot{X}_{ef} is the linear 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

$$F_o = -K_o P_o D_o - B_o \dot{X}_{ef} \quad (19)$$

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 obstacle and the end-effector.

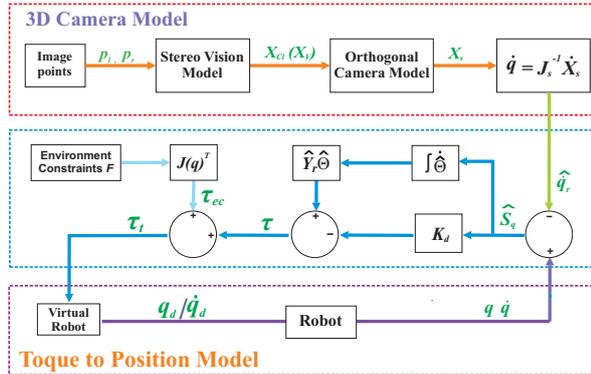


Fig. 3. 3D adaptive visual servoing including environment constraints

B. Torque to Position Model

This Impedance Control generates a virtual force that can be used to define a desired dynamic behavior for a robot manipulator. The changes in the robot dynamic behavior are reflected in the robot trajectory which is generated on-line and should take into account the target task and information of the environment (external factors). One way of generating this trajectory is to apply the forces to a virtual robot, whose dynamic behavior should be similar to the real robot. This virtual robot will generate the desired joint positions/velocities (q_d/\dot{q}_d), which will be used as the input of a position/velocity controller of the real robot, this is named as **Torque to Position Model** (see Fig. 3). The output joint position/velocity processed by the open architecture control of an industrial robot is certified and secure. This also allows the implementation of different control strategies in standard industrial robots.

This **Torque to Position Model** is designed to allow the implementation of joint torque control techniques on joint position-controlled robots. Robot torque control is essentially accomplished by converting desired joint torques into instantaneous increments of joint position inputs. This model can be implemented applied to any conventional position-controlled robot so that torque command to the robot becomes available.

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.

B. Experimental Results

We use a simple color-based visual tracker to identify the target (green, held by the human) and the robot end-effector (red). The control goal is to follow this object with the end-effector. The tracked positions of the red and green cubes are mapped from the stereo coordinate space to the 3D visual 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. Fig. 4 demonstrates the visual servoing control in combination with different forces. Fig. 4 a)- c) demonstrates how the robot handles self-collisions and obstacle avoidance while continuing to track the target, where the system generates a collision-free trajectory (robot motion-red line) instead of directly following the target (target motion-blue line). The green circles are the robot and a static obstacle. Fig. 4 e) 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. 4 f) 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 allowed to go under the table).

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. 4 d) and g). The visual tracking is resumed as soon as the target is visible again.

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.

C. 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, where two sets of position points defined in each coordinate frame O_b and O_{C_i} are used by using $SVD(M)$ (Singular Value Decomposition). The estimation errors for the complete Jacobian J_s can be handled in the controller to some extent.

Object occlusion occurs in Fig. 4 h), where the stereo system can then be moved to maintain the 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. We compare the actual end-effector position with the recovered end-effector position that is obtained by using the estimated rotation. Fig. 5 (a) 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. 5 (b).

A video where these behaviors are illustrated can be seen under: <http://youtu.be/UMjnFCQwThc>

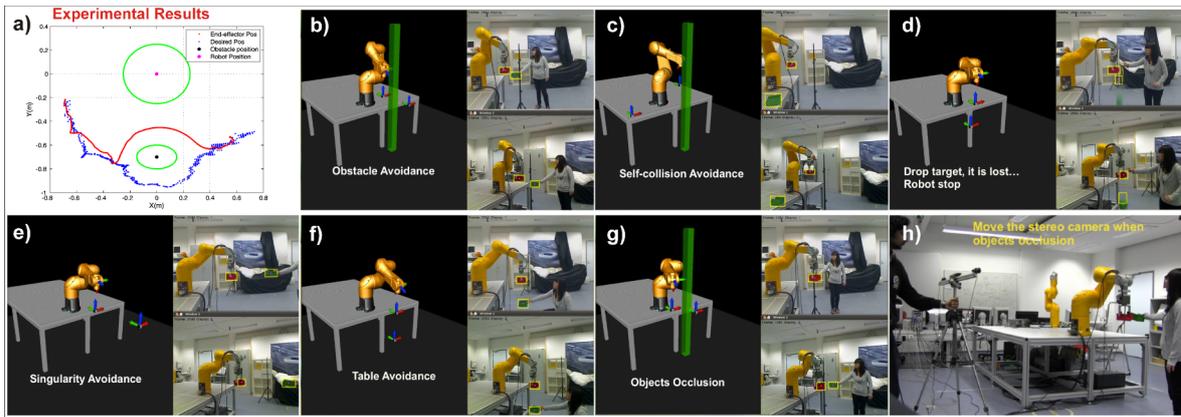


Fig. 4. Experiment results: a) A collision-free trajectory. b) Obstacles avoidance. c) Self-collision avoidance. d) The robots stops when the target is lost until is visible again. e) Singularity avoidance. f) Table avoidance. g) When the object is occluded, then robot changes its behavior. h) The camera can be moved when the target is occluded. The orientation matrix is estimated on-line.

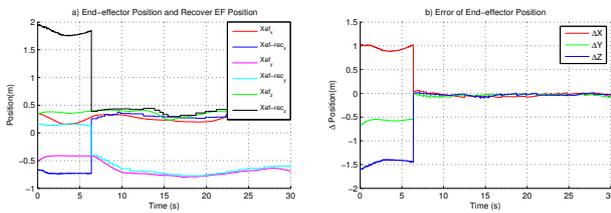


Fig. 5. Results of the on-line orientation matrix estimation: (a) shows the 3D position of the end-effector in the base frame and the recovered position for the end-effector using the estimated rotation, (b) shows the errors.

VI. CONCLUSIONS

We proposed a novel image-based controller for 3D image-based visual servoing using uncalibrated stereo vision system, which was evaluated on a real industrial robot. Results show the stability of the control even with parametric uncertainties. Furthermore, this work extends the adaptive image-based control law to include environment 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 improve the object tracking accuracy. Also, an 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 Transactions 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. 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] D. Kim, A. Rizzi, G. Hager, and D. Zoditschek, "A 'robust' convergent visual servoing system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Aug. 1995, pp. 348–353.
- [8] 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.
- [9] H. Wang, Y.-H. Liu, and W. Chen, "Uncalibrated Visual Tracking Control Without Visual Velocity," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1359–1370, Nov. 2010.
- [10] B. Yoshimi and P. Allen, "Alignment using an uncalibrated camera system," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 516–521, Aug. 1995.
- [11] 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.
- [12] 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.
- [13] S. Azad, Farahmand, Amir-Massoud, and M. Jagersand, "Robust jacobian estimation for uncalibrated visual servoing," in *Robotics and Automation (ICRA)*, May 2010, pp. 5564–5569.
- [14] K. Hashimoto, T. Ebine, and H. Kimura, "Dynamic Visual Feedback Control For A Hand-eye Manipulator," in *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, July 1992, pp. 1863–1868.
- [15] 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.
- [16] A. Astolfi, L. Hsu, M. Netto, and R. Ortega, "Two solutions to the adaptive visual servoing problem," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 387–392, June 2002.
- [17] 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.
- [18] 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.
- [19] A. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (2. ed.)*. Cambridge University Press, 2006.