Orthogonal Image Features for Visual Servoing of a 6 DOF Manipulator with Uncalibrated Stereo Cameras

Caixia Cai, Nikhil Somani, Alois Knoll

***Abstract*—We present an approach to control a 6 DOF manipulator using an uncalibrated visual servoing (VS) approach that addresses the challenges of choosing proper image features for target objects and designing a VS controller to enhance the tracking performance. The main contribution of this article is the definition of a new virtual visual space (image space). A novel stereo camera model employing virtual orthogonal cameras is used to map 6D poses from *Cartesian space* to this *virtual visual space*. Each component of the 6D pose vector defined in this virtual visual space is linearly independent, leading to a full-rank 6×6 *image Jacobian matrix* which allows avoiding classical problems, such as, image space singularities and local minima. Furthermore, the control for rotational and translational motion of robot are decoupled due to the diagonal image Jacobian. Finally, simulation results with an eye-to-hand robotic system confirm the improvement in controller stability and motion performance with respect to conventional VS approaches. Experimental results on a 6 DOF industrial robot are provided to illustrate the effectiveness of the proposed method and the feasibility of using this method in practical scenarios.**

***Index Terms*—Orthogonal Image Features, Visual Servoing.**

I. Introduction

Visual Servoing (VS) has been used in a wide spectrum of applications, from fruit picking to robotized surgery, and especially in industrial fields for tasks such as assembling, packaging, drilling and painting [1], [2]. According to the features used as feedback in minimizing the positioning error, visual servoing is classified into three categories [1]: *Position-Based Visual Servoing* (PBVS), *Image-Based Visual Servoing* (IBVS) and *Hybrid Visual Servoing* (HYVS).

In general, a PBVS system has a good 3D trajectory but is sensitive to calibration errors. Compared to PBVS, IBVS is known to be robust to camera model errors [3] and the image feature point trajectories are controlled to move approximately along straight lines[4]. However, one of the main drawbacks of IBVS is that there may exist image singularities and image local minima leading to IBVS failure. The selection of *visual features* is a key point to solve the problem of image singularities. A great deal of effort has been dedicated to determine decoupling visual features that deliver a triangular or diagonal Jacobian matrix [5], [6].

In IBVS, geometric features in the image such as points, segments or straight lines [1] are usually chosen as image features and used as the inputs of controllers. Several novel features such as laser points [7], the polar signatures of an object contour [8], and image moments [6], [9], [10] have been developed to track objects which do not have enough detectable geometric features and to enhance the robustness of visual servoing systems. The image interaction matrix (image Jacobian), can be computed using direct depth information [11], [12], by approximation via on-line estimation of depth of the features[13], [14], [15], [16], or using depth-independent image Jacobian matrix [17], [18]. Additionally, many papers directly estimate on-line the complete image Jacobian in different ways [19], [20], [21], [22]. However, all these methods generally use redundant image point coordinates to define a non-square image Jacobian leading to well-known problems such as image singularities.

It is also possible to combine the advantages of 2D and 3D visual servoing while avoiding their respective drawbacks. This approach is called 2-1/2D visual servoing because the used input is expressed partly in the 3D Cartesian space and partly in the 2D image space [23]. Contrary to PBVS, the 2-1/2D approach does not need any geometric 3D model of the object. In comparison to IBVS, the 2-1/2D approach ensures the convergence of the control law in the whole task space.

In this paper, we propose a new 2-1/2D visual servoing (coined 6DVS) which extracts new orthogonal image features and decouples the translational and the rotational control of a robot under visual feedback from fixed stereo cameras. More precisely, instead of using the classical visual features, we define a new virtual visual space (image space), where a 3D position vector is extracted as a feature. Each principal component of the position vector is linearly independent and orthogonal. We compute the orientation through a rotation matrix with Euler angles representation. We thus obtain a diagonal interaction matrix with very satisfactory decoupling properties. It is interesting to note that this Jacobian matrix has no singularity in the whole task space and the controls for the position and orientation are independent. Simulations and experimental results confirm that this new formulation (6DVS) is more efficient than existing classic VS approaches and the errors in both the virtual visual space and Cartesian space converge without local minima [1]. Moreover, it is less sensitive to image noise than classical 2-1/2D visual servoing.

In Section II we formulate the classical 2-1/2D VS approach, highlight its shortcomings and state the core issues. In Section III we introduce a new *camera model* to construct a *virtual visual space* and define a visual pose vector whose elements are chosen as image features. Using this 6D visual pose, a square full-rank *image Jacobian* is obtained, which is used in Section IV to simulate an adaptive 6D visual servoing controller and evaluate its properties. Section V presents two real-world experiments (Fig. 1) and shows the results obtained in a dynamic environment. Finally, Section VI presents the

Authors Affiliation: Robotics and Embedded Systems Group, Technische Universität München, Germany. Email: {caica,somani,knoll}@in.tum.de

[1]Parts of this work have already been presented at IROS'14 [24]. In this paper, quantitative validations of the approach and comparisons to classical methods in terms of steady state errors, transient systems performance and robustness to uncertainties have been added.

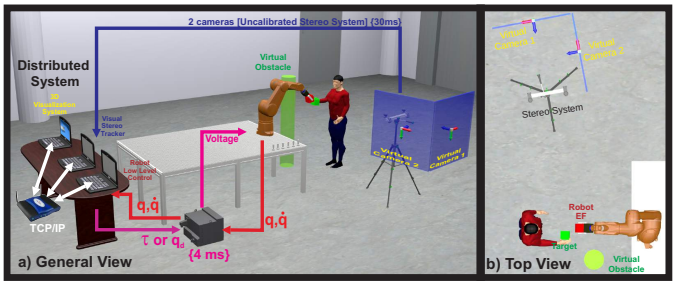

Fig. 1. Description of robotic experimental setup with fixed stereo cameras.

conclusions of our work and directions for future work.

II. Problem Formulation

A. The Problem of Classical Methods

Suppose that the robot end-effector is moving with angular velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$ and translational velocity $v = [v_x, v_y, v_z]^T$ both w.r.t. the camera frame in a fixed camera system (eye-to-hand configuration, see Fig. 1). Let P_c be a point rigidly attached to the end-effector with $X = [x, y, z]^T$.

In classical 2-1/2-D visual servoing, as described by Malis et al.[23], the selected feature vector is $h = [X_h^T, \theta U^T]^T$, where X_h is the position vector, θ and U are the rotation angle and axis of the rotation matrix R. The corresponding image Jacobian is an upper block triangular matrix given by

$$J = \begin{bmatrix} \frac{1}{z} J_v & J_{v\omega} \\ 0_3 & J_\omega \end{bmatrix} \qquad \dot{h} = J \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

The position vector $X_h = [u, v, ln(z)]^T$ is defined in extended image coordinates, where u, v are the image features, 2D data (pixel) and z is the depth of the considered point, 3D data (meter). Moreover, according to image Jacobian (1), the translational velocity of the point P_c is affected by both position and orientation errors.

u and v are two orthogonal principal axes in image coordinates. If we can find a third, normalized z_s component for image coordinates which is orthogonal to u, v and measured in pixels, then all the points in the image coordinates can be decomposed into 3 principal components in such a way that all the elements of the position vector can be controlled in a linearly independent way.

B. Design of proposed VS Features

Motivated by the desire to find a z_s component of the position vector which is also in the image plane (pixel) and decouples the control of the translational and the rotational motion, we define a new virtual visual space (image space), where a 3D pixel position X_s is extracted as a feature. All elements of this 3D position vector are linearly independent and orthogonal to each other. We solve the orientation using a rotation matrix with ZYX Euler angles representation, denoted by θ. Thus, the new feature vector is $W_s = [x_s, y_s, z_s, \alpha, \beta, \gamma]^T$ and the new mapping is given by

$$\dot{W}_s = \begin{bmatrix} \dot{X}_s \\ \dot{\theta} \end{bmatrix} = \overbrace{\begin{bmatrix} J_v & 0_3 \\ 0_3 & J_\omega \end{bmatrix}}^{J_{img}} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

where the new image Jacobian ($J_{img} \in \mathbb{R}^{6\times6}$) is a decoupling diagonal matrix that decouples the translational and rotational control.

III. 6D Adaptive Visual Servoing

Consider the motion of a plane π attached to the end effector of a robot that rotates and translates through space in order to obtain a desired position and orientation of the end-effector. We define four target points on π denoted by P_i, $\forall i = 1, 2, 3, 4$. In this work, we investigate the translational and rotational motion of the end-effector of a robot under visual feedback from a fixed stereo camera system. By assuming knowledge of the camera intrinsic parameters, we obtain the pixel translation motion using triangulation on the center of the four points while utilizing the rotational information of the end-effector through the motion of four tracked points.

The image Jacobian J_{img} has a decoupled structure, which is divided into *position image Jacobian* ($J_v \in \mathbb{R}^{3\times3}$) and *orientation image Jacobian* ($J_\omega \in \mathbb{R}^{3\times3}$).

A. Image Jacobian for 3D Position J_v

Since $X_s \in \mathbb{R}^{3\times1}$ represents the position in the image feature space, the maximum number of independent elements for position is 3. Hence, in this work we construct a virtual visual space using the information generated from the stereo vision system where 3 linearly independent elements can be extracted to get a full-rank image Jacobian (J_v).

$J_v \in \mathbb{R}^{3\times3}$ describes the relationship between the velocities of 3D Cartesian position $\dot{X}_b$ ($meters$) and 3D visual position $\dot{X}_s$ ($pixels$). The key idea of this model is to combine the stereo camera model with a virtual composite camera model to get a full-rank image Jacobian, see Fig. 2.

This new 3D visual model can be computed in two steps:

- The standard stereo vision model [25] is used to analytically recover the 3D *relative* position (X_{C_l}) of an object with respect to the reference frame of the stereo system O_{C_l}.
- The Cartesian position X_{C_l} is projected into two virtual cameras O_{v_1} and O_{v_2}.

*1) **Stereo Vision Model**:* Defining the observed image points in each camera as $p_l = [u_l, v_l]^T$, $p_r = [u_r, v_r]^T$, we can use triangulation [25] to compute the *relative* position $X_{C_l} = [x_c, y_c, z_c]^T$ with respect to the left camera O_{C_l}. Then the position X_{C_l} can be mapped to the world frame X_b through

$$X_{C_l} = R^b_{C_l} X_b + t^b_{C_l} \tag{3}$$

where $T^b_c = [R^b_{C_l}, t^b_{C_l}]$ is the transformation matrix between coordinate frame O_{C_l} and O_b.

Before *integrating* the stereo cameras model with the virtual composite model, a re-orientation of the coordinate frame O_{C_l} to a new coordinate frame O_V with the same origin is required. The projection position $X_V = [x_V, y_V, z_V]^T$ with (3) is (Fig. 2)

$$X_V = R^{C_l}_V X_{C_l} = R^{C_l}_V (R^b_{C_l} X_b + t^b_{C_l}) \tag{4}$$

where $R^{C_l}_V$ is the orientation of the reference frame[2] O_V with respect to O_{C_l}.

[2]This reference frame is fixed to the left camera coordinate frame and is defined by the user, therefore $R^{C_l}_V$ is assumed to be known.

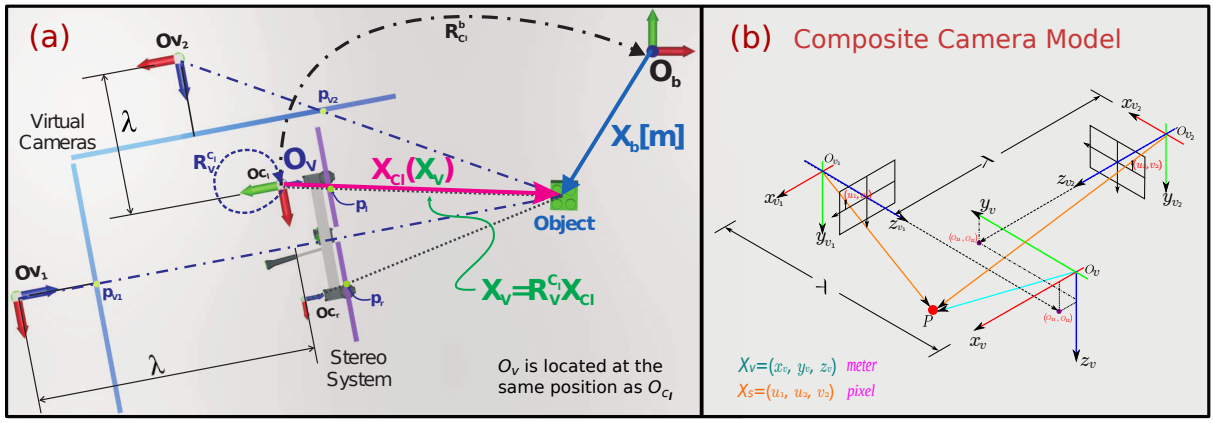

Fig. 2. Image projections: (a) The figure depicts the different coordinate frames used to obtain a general 3D *virtual visual space*. $X_b \in \mathbb{R}^{3\times1}$ is the position in meters [m] of an *Object* with respect to the world coordinate frame (wcf) denoted by O_b. Moreover, O_{C_l} and O_{C_r} 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 $p_l, p_r \in \mathbb{R}^{2\times1}$ are the projections of the point X_b in the left and right cameras. Finally, $p_{v_i} \in \mathbb{R}^{2\times1}$ represents the projection of the *Object* in the virtual cameras O_{v_i}. (b) Placement of the composite camera model with respect to left camera.

*2) **Virtual Composite Camera Model**:* In order to compute the 3D virtual visual space, we define two virtual cameras attached to the stereo camera system using the coordinate frame O_V (Fig. 2 (b)). We use the pinhole camera model [25] 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

$$p_{v_1} = \begin{bmatrix} u_{v_1} \\ v_{v_1} \end{bmatrix} = \frac{1}{-y_V+\lambda} \alpha R(\phi) \begin{bmatrix} x_V - o_{11} \\ z_V - o_{12} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}. \quad (5)$$

where ϕ is the rotation angle of the virtual camera along its optical axis, $O_1 = [o_{11}, o_{12}]^T$ is the projected position of the optical center with respect to the coordinate frame O_V, $C_1 = [c_x, c_y]^T$ is the position of the principal point in 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 $R(\phi)$ are defined as:

$$\alpha = \begin{bmatrix} f\beta & 0 \\ 0 & f\beta \end{bmatrix} \qquad R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}. \quad (6)$$

where f is the focal length of the lens used and β is the magnification factor of the camera.

Since this model represents a user-defined virtual camera, all its parameters [3] are known in the defined configuration of the virtual cameras $\phi = 0$[4] (Fig. 2 (b)).

Similarly, the model for virtual camera 2 is defined as:

$$p_{v_2} = \begin{bmatrix} u_{v_2} \\ v_{v_2} \end{bmatrix} = \frac{1}{x_V+\lambda} \alpha R(\phi) \begin{bmatrix} y_V - o_{21} \\ z_V - o_{22} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}. \quad (7)$$

In order to construct the 3D virtual visual space $X_s \in \mathbb{R}^{3\times1}$, we combine both virtual camera models.

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

[4]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 a specific orientation such as $\phi = 0$.

Using properties of the rotation matrix $R(\phi)$ and the fact that α is a diagonal matrix, from (5), u_{v_1} can be written in the form

$$u_{v_1} = \gamma_1 \frac{x_V - o_{11}}{-y_V+\lambda} - \gamma_2 v_{v_1} + \gamma_3 \quad (8)$$

where the constant parameters $\gamma_1,\ \gamma_2,\ \gamma_3 \in \mathbb{R}$ are explicitly defined as

$$\gamma_1 = \frac{f\beta}{\cos\phi}, \quad \gamma_2 = \tan(\phi), \quad \text{and } \gamma_3 = c_x + c_y\gamma_2. \quad (9)$$

Based on (7) and (8), we define a *visual camera model* (O_s) representation $X_s = [x_s, y_s, z_s]^T$ using the orthogonal elements $[u_{v_1}, u_{v_2}, v_{v_2}]^T$ as

$$X_s = \begin{bmatrix} u_{v_1} \\ u_{v_2} \\ v_{v_2} \end{bmatrix} = \overbrace{\begin{bmatrix} \gamma_1 & 0_{1\times2} \\ 0_{2\times1} & \alpha R(\phi) \end{bmatrix}}^{R\alpha} \begin{bmatrix} \frac{x_V - o_{11}}{-y_V+\lambda} \\ \frac{y_V - o_{21}}{x_V+\lambda} \\ \frac{z_V - o_{22}}{x_V+\lambda} \end{bmatrix} + \rho \quad (10)$$

where $\rho = [\gamma_3 - \gamma_2 v_{v_1}, c_x, c_y]^T$. The pixel position X_s constructs the 3D *virtual visual space*.

Given that $\phi = 0$, then $\gamma_1 = f\beta$, $\gamma_2 = 0$, $\gamma_3 = c_x$, implies that $\rho = [c_x, c_x, c_y]^T$ and $R_\alpha = diag(f\beta) \in \mathbb{R}^{3\times3}$. Therefore, the mapping in (10) can be simplified as

$$X_s = diag(f\beta) \begin{bmatrix} \frac{x_V - o_{11}}{-y_V+\lambda} \\ \frac{y_V - o_{21}}{x_V+\lambda} \\ \frac{z_V - o_{22}}{x_V+\lambda} \end{bmatrix} + \begin{bmatrix} c_x \\ c_x \\ c_y \end{bmatrix}. \quad (11)$$

The velocity mapping can be obtained with the time derivative of (11) as follows:

$$\dot{X}_s = R_\alpha J_o \dot{X}_V = J_\alpha \dot{X}_V \quad (12)$$

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

$$J_o = \begin{bmatrix} \frac{1}{-y_V+\lambda} & \frac{x_V - o_{11}}{(-y_V+\lambda)^2} & 0 \\ -\frac{y_V - o_{21}}{(x_V+\lambda)^2} & \frac{1}{x_V+\lambda} & 0 \\ -\frac{z_V - o_{22}}{(x_V+\lambda)^2} & 0 & \frac{1}{x_V+\lambda} \end{bmatrix}. \quad (13)$$

Taking the time derivative of (4), (12) can be rewritten as

$$\dot{X}_s = J_\alpha (R_V^{C_l} R_{C_l}^b) \dot{X}_b = J_v \dot{X}_b \tag{14}$$

where we define $J_v \in \mathbb{R}^{3\times 3}$ as the *position image Jacobian*.

Remark 1: Virtual Cameras. The two virtual cameras are selected in such a way that their optical axes intersect at 90 degrees. Since the cameras are virtual they have infinite field of view and pixel positions X_s can be either negative or positive.

B. Image Jacobian for 3D Orientation J_ω

Let $\theta = [\alpha, \beta, \gamma]^T$ be a vector of *ZYX* 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 [26]

$$\omega = T(\theta)\dot{\theta}. \tag{15}$$

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

$$T(\theta) = \begin{bmatrix} \cos(\gamma)\cos(\beta) & -\sin(\gamma) & 0 \\ \sin(\gamma)\cos(\beta) & \cos(\gamma) & 0 \\ -sin(\beta) & 0 & 1 \end{bmatrix} \tag{16}$$

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

Therefore,

$$\dot{\theta} = T^{-1}(\theta)\omega = J_\omega \cdot \omega. \tag{17}$$

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

Combining (14) and (17) we have the full expression

$$\dot{W}_s = \begin{bmatrix} \dot{X}_s \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} J_v & 0 \\ 0 & J_\omega \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{18}$$

$$= J_{img} \cdot V \tag{19}$$

where the matrix $J_{img} \in \mathbb{R}^{6\times 6}$ is defined as the new ***image Jacobian***, which is a block diagonal Jacobian matrix.

C. Control Scheme

Substituting the robot *differential kinematics* $V = J(q)\dot{q}$, equation (19) can be rewritten in the form

$$\dot{W}_s = J_{img} J(q) \cdot \dot{q} = J_s \cdot \dot{q} \tag{20}$$

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

According to (20) the corresponding control law is

$$\dot{q_r} = {J_s}^{-1} \dot{W}_{s_r} \tag{21}$$

where $\dot{q_r}$ is the joint velocity nominal reference and is used in an adaptive second order sliding mode controller, which is described in detail in the paper [27].

Remark 2: Singularity-free J_{img}.
From (18), we can see that $det(J_{img}) = det(J_v) det(J_\omega)$. Hence, the set of singular configurations of J_{img} is the union of the set of position configurations satisfying $det(J_v) = 0$ and the set of orientation configurations satisfying $det(J_\omega) = 0$.

From (14), we can see that $J_v^{-1} = {R_{C_l}^b}^{-1} {R_V^{C_l}}^{-1} {J_o}^{-1} R_\alpha^{-1}$. The matrices $R_{C_l}^b, R_V^{C_l} \in SO(3)$ and $R_\alpha = diag(f\beta) \in \mathbb{R}^{3\times 3}$ are non-singular. Then, $det(J_o) = 0 \rightarrow det(J_v) = 0$. 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) $y_V = \lambda$ and $x_V = O_{11}$. However, O_{11}, O_{21} and λ are all defined by the user. Hence, a non-singular J_v can be obtained by enforcing 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 with respect to O_V. Therefore, $det(J_v) \neq 0$ and can not become infinite.

Provided $det(T(\theta)) \neq 0$, J_ω^{-1} always exists. Therefore, the singularities of J_s are defined only by the singularities of $J(q)$.

IV. Simulation

We simulate a 6DOF industrial robot with real robot parameters in closed loop with the control approach. Real camera parameters are used to simulate the camera projections. Our simulation platform is identical to the real experiments, except that we simulate the 6D desired pose. The robot motions are visualized in a 3D visualization system (Section V-A3).

Simulation tests have been carried out with four feature points, which give us a 16×6 interaction matrix in the classical IBVS with stereo vision system and a 6×6 Jacobian matrix in our algorithm (6DVS). In proposed method 6DVS, the image features s are mapped to the virtual visual space to get X_s, which is used to design the error function for the control scheme. For classical stereo IBVS, the image features s are directly used to design the error function and the real z obtained from the stereo vision system is used to compute the interaction matrix. The 3D Cartesian position from the stereo vision system is used to perform PBVS. Classical 2-1/2D (2.5DVS) with Euler angle representation is also used in the comparisons.

TABLE I
Initial(I) and Desired(D) location of feature points in image plane (Pixel) of left camera

		Point 1		Point 2		Point 3		Point 4	
		(u	v)	(u	v)	(u	v)	(u	v)
Test 1	I	(207	254)	(194	212)	(213	200)	(225	238)
	D	(422	379)	(406	307)	(471	290)	(486	360)
Test 2	I	(207	254)	(194	212)	(213	200)	(225	238)
	D	(422	379)	(406	307)	(471	290)	(486	360)
Test 3	I	(318	224)	(304	191)	(312	177)	(325	208)
	D	(226	228)	(200	184)	(244	159)	(269	203)
Test 4	I	(207	254)	(194	212)	(213	200)	(225	238)
	D	(291	444)	(304	382)	(367	376)	(354	434)

The initial (I) and desired (D) configurations of the image features ($s = [u_1, v_1, u_2, v_2, u_3, v_3, u_4, v_4]^T$)[5] in left camera for each of the tests are shown in Table I.

1) Test 1: In this test, we examine the convergence of each image feature point and pose errors when the desired location is far away from the initial one. The Cartesian and image trajectories of the proposed 6D visual servoing (6DVS) and the conventional approaches (IBVS, PBVS and 2.5DVS) are compared in Fig. 3.

[5]In simulation figures, only image features in the left camera are illustrated, since the results in the right camera are similar.

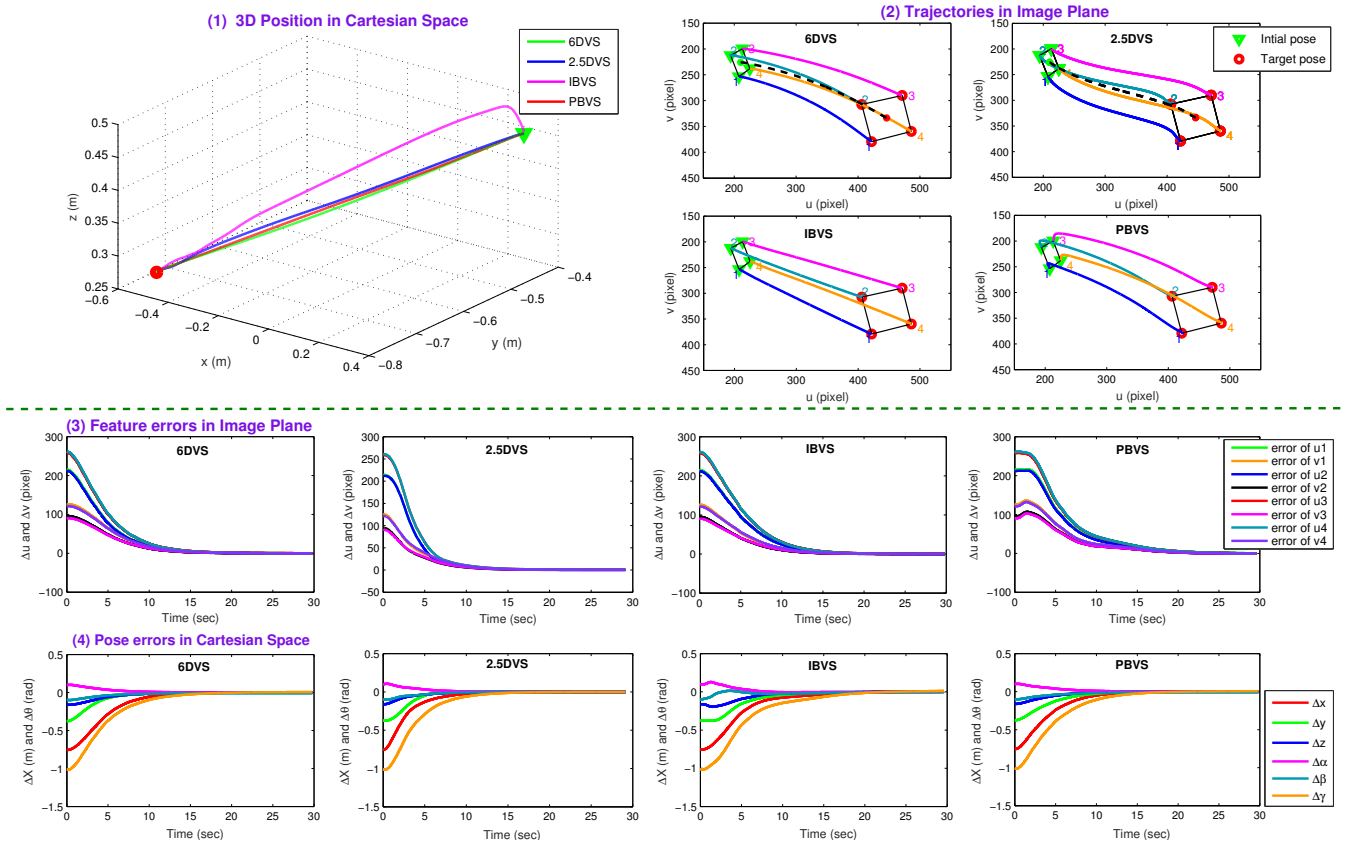

Fig. 3. Simulation Test 1: Large translational and rotational motion. (1) 3D robot end-effector trajectory in Cartesian (X_b); (2) image plane trajectories of 6DVS, 2.5DVS, IBVS and PBVS (s); (3) feature errors in the image plane, (4) robot end-effector pose errors in 3D Cartesian space.

As shown in Fig. 3 (1), PBVS results in a straight line end-effector Cartesian trajectory. Since there is no control of the image features, the image trajectory, (as shown in Fig. 3 (2)), is unpredictable and may leave the camera field of view. In IBVS, straight-line image trajectory is observed while the end-effector Cartesian trajectory is not controlled. For 2.5DVS, the trajectory of the reference point in the image is a straight line and the Cartesian trajectory is also well behaved. However, other points in image plane have curved trajectories.

Contrary to the previous approaches, the proposed 6DVS has a straight-line Cartesian trajectory (similar to that of PBVS) and all the image features are "indirectly" controlled to move approximately along straight-line trajectories like IBVS, see Fig. 3 (1), (2). Moreover, both the features errors and the Cartesian pose errors converge to zero very smoothly without any overshooting (Fig. 3 (3), (4)). Although 2.5DVS has a similar trade-off between these properties, the proposed 6DVS is more efficient and has better performance than 2.5DVS. Hence, 6DVS combines the advantages of PBVS in terms of controlling straight trajectories in Cartesian Space, and the advantages of IBVS in terms of controlling image trajectories.

2) Test 2: For this test we compare the robustness of the proposed 6DVS, the classical IBVS and PBVS to camera errors. These errors are formulated as:

- Camera intrinsic parameter errors $\hat{f} = 1.1f$.
- Camera extrinsic parameter errors $\hat{T}_c^b = 1.1T_c^b$.

The results from this test show that despite the camera errors, the controllers for each of the evaluated approaches don't become unstable. The resulting Cartesian and image trajectories have some notable differences (see Fig. 4), but are still well behaved.

Due to camera errors, the end-effector Cartesian trajectory of PBVS deviates from the original straight-line trajectory to a circular motion, and slight effects on the image trajectories can also be observed (see Fig. 4 (c)). Slight differences in the end-effector Cartesian trajectory of IBVS are shown in Fig. 4 (b). As expected, the image trajectories for IBVS are robust to camera errors. Fig. 4 (a) illustrates that the effects on both Cartesian and image trajectories in the proposed 6DVS are minor. Hence, 6DVS is as robust to camera calibration errors as IBVS.

3) Test 3: In this test we evaluate a common problem in classical IBVS: local minima. By definition, local minima are cases where $V = 0$ and $s \neq s_d$. So, a local minimum is reached when the point velocity on the robot end-effector is zero while its final position is far away from the desired position. At that position, the errors $s - s_d$ in image plane do not completely vanish (residual error is approximately two pixels on each u and v coordinate). Introducing noise in the image measurement leads to the same results.

Reaching such a local minimum is illustrated in Fig. 5 (b) for IBVS. Each component of the feature errors e has a exponential convergence but is not exactly zero ($s \neq s_d$) while

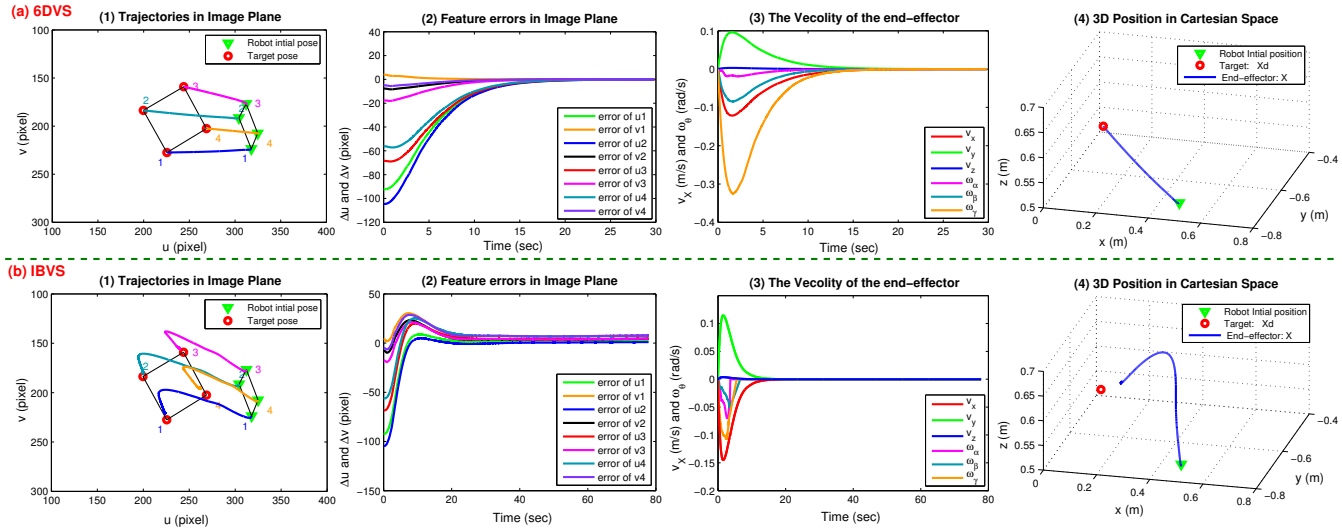

Fig. 5. Simulation Test 3: Reaching (or not) local minima. Row (a) is reaching a global minimum using the 6DVS algorithm: (1) feature trajectories (s) in image plane, (2) feature errors e, (3) the evolution of the six components of the point velocity on the robot end-effector, (4) the trajectory of the robot end-effector in 3D Cartesian Space. Row (b) is reaching a local minima using the classical IBVS.

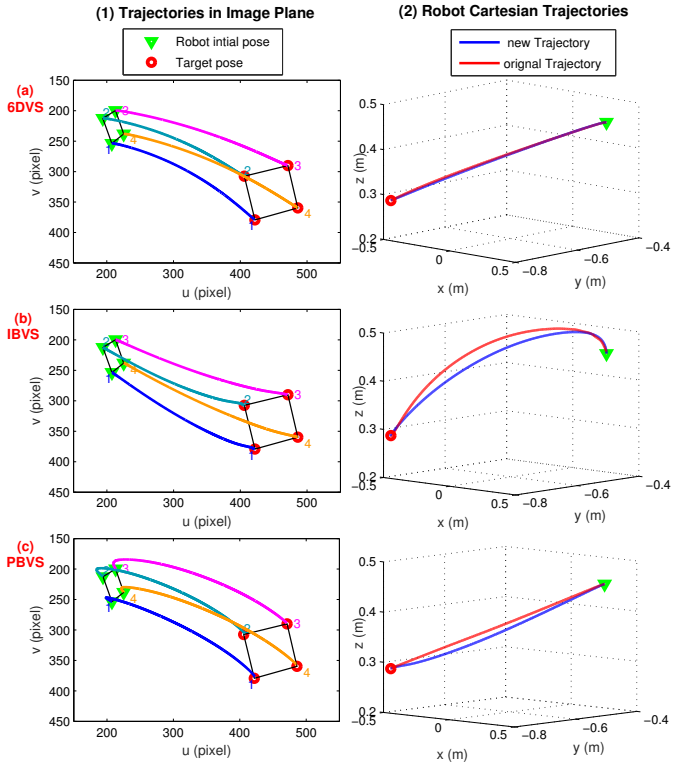

Fig. 4. Simulation Test 2: Comparison of the robustness of the controllers with effects of camera errors. Comparison of 6DVS, IBVS and PBVS in terms of (1) image feature trajectories and (2) 3D Cartesian trajectories.

the robot velocity is close to zero in Fig. 5 (b).(3). It is clear from Fig. 5 (b).(4) that the system has been attracted to a local minimum far away from the desired configuration.

In the proposed scheme (6DVS), the image Jacobian J_{img} has full rank of 6, which implies there are no local minima. The global minimum is correctly reached from the same initial position if the proposed J_{img} is used in the control scheme (Fig. 5 (a)). In this case, the trajectories in image plane are straight and each component of the errors e has a exponential convergence to zero without local minima. Moreover, when the velocity reaches zero, the errors in image plane and Cartesian space are both close to zero ($V \to 0, \Delta s \to 0, \Delta X_b \to 0$).

4) Test 4: Motion decoupling is compared between the proposed 6DVS and conventional 2.5DVS in this test. Both approaches receive the same desired Cartesian position and orientation. First the desired position is given and both methods perform equally well. As shown in Fig. 6, the trajectories in both Cartesian space and the image plane are smooth for 6DVS and 2.5DVS in the position task.

At around $t = 30s$, the desired orientation is modified. In the standard 2-1/2D method, a triangular interaction matrix is used for motion control, as defined in (1). Hence, the rotational error can affect the translation, as shown in Fig. 6 (b). The visual signals are coupled and both position and orientation in the Cartesian space change when only the desired orientation is updated. For the proposed 6DVS, we introduce the virtual visual space to decouple the control features and get a diagonal image Jacobian. This decoupling of rotational and translational motions allows a better control design. Fig. 6 (a) demonstrates the decoupled performance in the proposed method.

Simulation results of four different tests demonstrate the novel properties and better performance of the proposed 6DVS algorithm over conventional VS approaches. 6DVS has a reliable straight 3D Cartesian trajectory like PBVS, and straight feature trajectories like IBVS. Moreover, 6DVS allows to avoid local minima (unlike IBVS) and is robust to camera calibration errors. Contrary to classical 2-1/2D visual servoing, 6DVS decouples the control of the translational and rotational motion.

V. Experiments

Two experiments were performed to validate and evaluate this work on a standard industrial manipulator in a realistic

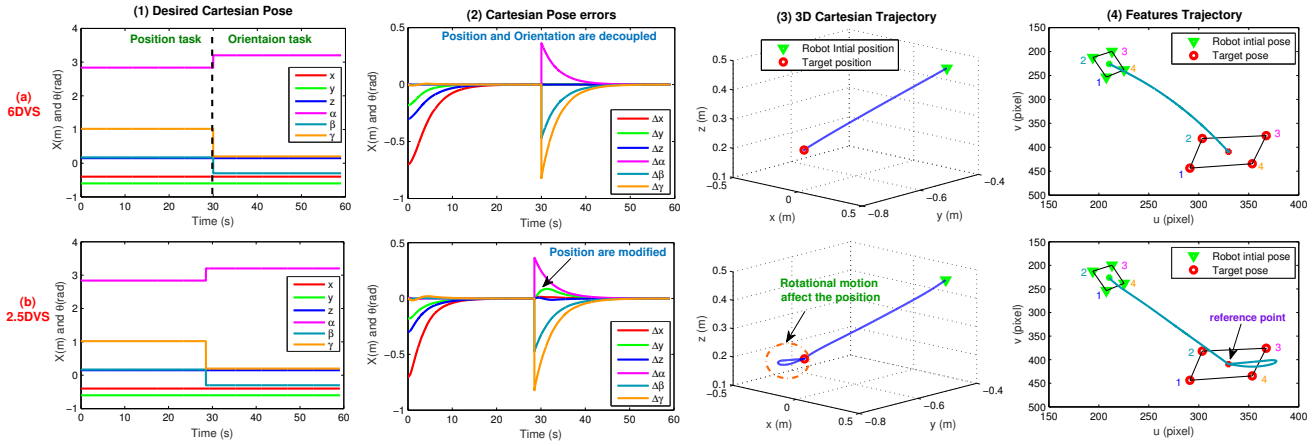

Fig. 6. Simulation Test 4: Decoupling analysis of position and orientation. Row (a) is the decoupled result for 6DVS: (1) the desired Cartesian pose, (2) Cartesian pose errors, (3) 3D Cartesian trajectory of the robot end-effector, (4) image feature trajectory of the reference point. Row (b) is coupled result using the classical 2.5DVS.

human-robot interaction scenario. In the first experiment, we control the robot without environment constraints to better illustrate the stability of the control scheme and the convergence of 6D visual trajectory error. The second experiment uses the 6DVS scheme for tracking a moving target in real-time. It is also an example of how the proposed algorithm is used in a practical human-robot interaction scenario. To this effect, several other features such as singularity avoidance, self-collision avoidance and obstacle detection and avoidance are implemented to ensure safety of the robot and human.

A. System Overview

The experimental setup consists of 3 sub-systems:

1) Visual Stereo Tracker: The stereo system is composed of 2 USB cameras fixed on a tripod, in a eye-to-hand configuration (Fig. 1). The stereo rig is uncalibrated with respect to the robot base frame and can be manually moved. The parameters of the virtual cameras (see Section III) are selected such that J_α is always non-singular. In order to compute torque (τ) and avoid a multiple-sampling system, an extended Kalman filter (EKF) is used to estimate the visual position (sampling period 4*ms*), whereas the reference is updated each 30*ms* with the real visual data of both cameras.

2) Robot Control System: The robot system comprises a StaübliTX90 industrial robot arm (6DOF), a CS8C control unit and a workstation running on GNU/Linux OS with real-time kernel (Fig. 1). 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, using the Robotics Library[6]. The system updates the configuration of the robot arm and the position of the target in real time, which is achieved by means of TCP/IP communication.

B. Experiment 1: 6D Visual Tracking

2D image features are extracted from AR markers using a stereo vision system. We use the ArUco library based on OpenCV to detect markers. Every marker provides 2D image features for 4 corner points. 3D position and rotation with respect to the camera frame are obtained from the image features using the camera intrinsic parameters.

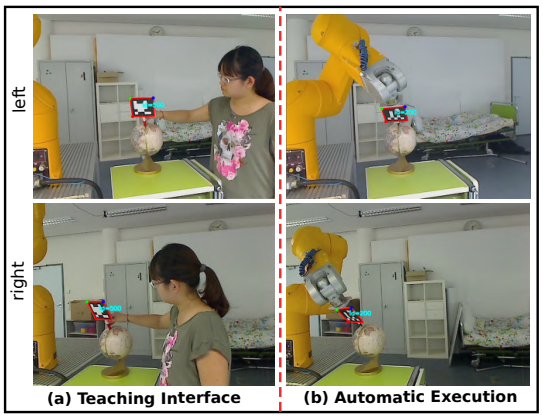

Fig. 7. Snapshot of the 6D visual tracking: (a) The human uses the AR marker to show the desired trajectories, (b) The robot executes a 6D visual tracking and shows identical linear and angular motions as taught.

This experiment consists of two phases: teaching and execution.

*1) **Teaching Interface**:* We provide a teaching interface (Fig. 7 (a)), where the user is holding an AR marker, detected by the stereo camera system to provide 2D image features. A red square and a marker ID (cyan) in the image shows the detection. In this task, the user moves the marker, creating some visual trajectories, e.g., 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 four corner points of the marker are recorded and saved. At some points, when the marker is lost or can not be detected, the last available detection is stored thereby, guaranteeing that the desired pose can be reached and is safe for the robot execution.

*2) **Automatic Execution**:* After the teaching phase, the robot can automatically execute the recorded visual trajec-

[6]www.roboticslibrary.org

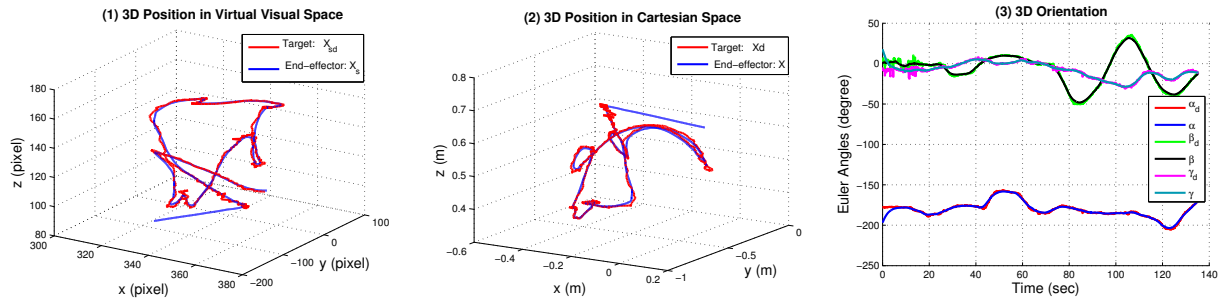

Fig. 8. Experiment results for 6D visual tracking: (1) 3D visual position (pixel) for robot end-effector and the desired one in virtual visual space, (2) 3D position (meter) in Cartesian Space,(3) Euler angles (degree) for robot end-effector and the desired target.

tories. Another AR marker with the same size is attached to the robot end-effector. The current robot position (W_s) is obtained from the visual features tracked from this marker. Target inputs to visual servoing are the 2D image features which were recorded in the teaching phase. From the recorded features we extract our desired visual feature vector $W_{s_d} = [x_{s_d}, y_{s_d}, z_{s_d}, \alpha_d, \beta_d, \gamma_d]^T$, which is used to create the error function (see Section III).

Visual servoing is accomplished by driving the error function to zero. In our case, the error function is $e = W_s - W_{s_d}$. The position components of e are from the new position vector X_s as (11) and the rotational components are obtained through 3D data. According to the properties of our image Jacobian and the control scheme, when the errors in the virtual visual space converge to zero, the errors in Cartesian space also 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 phase (Fig. 7). This experiment illustrates how the visual servoing system can track a given desired trajectory.

Experimental results are depicted in Fig. 8. The plot (1) shows the 3D linear visual tracking in the virtual visual space while the second plot (2) depicts the target trajectory tracking in Cartesian space. Plot (3) illustrates the rotational motion tracking. The red lines in plots (1) and (2) are the target trajectories, which exhibit some noise and chattering due to the unsteady movement of the user. However, the blue lines which show the trajectories of the robot end-effector, are smooth and chatter free.

C. Experiment 2: 6D Uncalibrated VS in HRI Scenario

In this experiment, we integrate the proposed visual servoing system in a Human-Robot Interaction scenario (HRI). The robot manipulator follows, in real time, an AR marker manipulated by a user while avoiding environment constraints such as robot singularities and collision with itself or obstacles. The artificial potential field approach [28] is used to model these environment constraints. A coarse on-line estimation of camera parameters is computed using the real-time information generated by the robot (more details are shown in paper [24]).

Interaction results: This experiment demonstrates real time tracking of a moving target held by the user by the robot end-effector. Both translation and rotation motions are tracked in this system (Fig. 9 (a)). 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), Fig. 9 (b).

To demonstrate stability, we test our system under several environmental constraints. Fig. 9 (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. 9 (d) depicts the 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 allowed to go under the table) but it can still move in the x_b and y_b axes, and Fig. 9 (e) shows how the robot handles self-collisions. Fig. 9 (f) shows obstacle avoidance while continuing to track the target.

A video illustrating more details for all these experimental results can be seen at: http://youtu.be/zqmapL51g9I

VI. Conclusions

In this paper, we have investigated the control of translational and rotational motion for the end-effector of a robotic manipulator under visual feedback from fixed stereo cameras. We have proposed a new *virtual visual space* (measured in pixels) for visual servoing using an uncalibrated stereo vision system in combination with virtual orthogonal cameras. Using a 6D visual pose vector defined in this virtual space, we obtain a new full-rank image Jacobian that can avoid the well-known problems such as image space singularities and local minima. Moreover, the rotational and translational motions of robot end-effector are decoupled due to the diagonal image Jacobian. According to simulation results, these new features perform better than classical ones since the system combines the advantages of 2D and 3D visual servoing. Furthermore, the proposed algorithm was integrated in a practical human-robot-interaction scenario with environmental and kinematic constraints to generate a robot dynamic system with a trajectory free of collisions and singularities.

Future work includes further analysis of the limitations of the proposed scheme and more applications utilizing this new virtual visual space. The choice of the controller was not a major contribution of this work and further research could be done to evaluate the effects of using different controllers with the proposed VS approach. Also, we plan to implement this approach in more real world applications, by fusing this technique with different sensors, e.g. depth-cameras and force sensors.

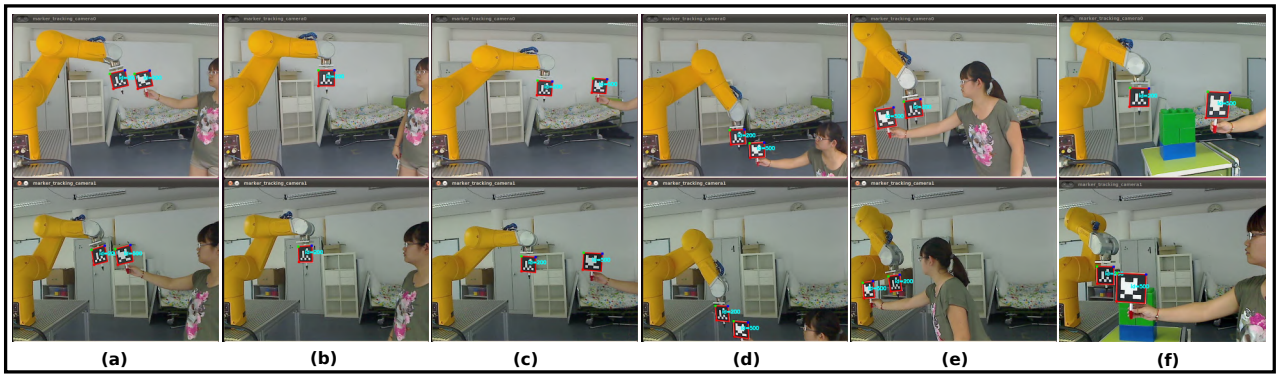

Fig. 9. System behaviors: (a) Position and orientation tracking, (b) case when the target is lost, (c) case with singularity avoidance, (d) case with table collision avoidance, (e) case with self-collision avoidance and (f) obstacle avoidance.

REFERENCES

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

[2] J.-K. Oh, S. Lee, and C.-H. Lee, "Stereo vision based automation for a bin-picking solution," *Int. J. Control, Autom. Syst.*, vol. 10, no. 2, pp. 362–373, 2012.

[3] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *The 3rd Int. Symp. Experimental Robotics*. London, UK: Springer-Verlag, 1993, pp. 182–192.

[4] 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.

[5] J. Wang and H. Cho, "Micropeg and hole alignment using image moments based visual servoing method," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1286–1294, Mar. 2008.

[6] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[7] A. Krupa, J. Gangloff, M. de Mathelin, C. Doignon, G. Morel, L. Soler, J. Leroy, and J. Marescaux, "Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser pointers," in *IEEE Int. Conf. Robot. Autom.*, vol. 4, May 2002, pp. 3769–3774.

[8] C. Collewet and F. Chaumette, "A contour approach for image-based control on objects with complex shape," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Nov. 2000, pp. 751–756.

[9] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1116–1127, Dec. 2005.

[10] Y. Zhao, W.-F. Xie, and S. Liu, "Image-based visual servoing using improved image moments in 6-dof robot systems," *Int. J. Control, Autom. Syst.*, vol. 11, no. 3, pp. 586–596, 2013.

[11] J. Feddema, C. 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.

[12] Y. Mezouar and F. Chaumette, "Optimal camera trajectory with image-based control." *Int. J. Robot. Res.*, vol. 22, no. 10, pp. 781–804, 2003.

[13] F. Chaumette and S. Hutchinson, "Visual servo control I: basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[14] F. Janabi-Sharifi, L. Deng, and W. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 5, pp. 967–983, Oct. 2011.

[15] N. Papanikolopoulos and P. Khosla, "Adaptive robotic visual tracking: theory and experiments," *IEEE Trans. Autom. Control*, vol. 38, no. 3, pp. 429–445, Mar. 1993.

[16] E. Nematollahi and F. Janabi-Sharifi, "Generalizations to control laws of image-based visual servoing," *Int. J. Optomechatronics*, vol. 3, no. 3, pp. 167–186, 2009.

[17] D. Kim, A. Rizzi, G. Hager, and D. Zoditschek, "A robust convergent visual servoing system," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Aug. 1995, pp. 348–353.

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

[19] B. Yoshimi and P. Allen, "Alignment using an uncalibrated camera system," *IEEE Trans. Robot. Autom.*, vol. 11, no. 4, pp. 516–521, Aug. 1995.

[20] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Sep. 1994, pp. 186–193.

[21] J. Piepmeier, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 143–147, Feb. 2004.

[22] S. Azad, Farahmand, Amir-Massoud, and M. Jagersand, "Robust jacobian estimation for uncalibrated visual servoing," in *IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5564–5569.

[23] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[24] C. Cai, E. Dean-Leon, N. Somani, and A. Knoll, "6d image-based visual servoing for robot manipulators with uncalibrated stereo cameras," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014.

[25] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (2. ed.)*. Cambridge University Press, 2004.

[26] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control-Second Edition*, 2004.

[27] 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 Int. Conf. Intell. Robots Syst.*, Nov. 2013.

[28] C. Cai, N. Somani, S. Nair, D. Mendoza, and A. Knoll, "Uncalibrated stereo visual servoing for manipulators using virtual impedance control," in *Int. Conf. Control, Autom., Robot. Vision*, Dec. 2014.