

Adaptive Homography-Based Visual Servo Tracking Control via a Quaternion Formulation

Guoqiang Hu, Nicholas Gans, Norman Fitz-Coy, and Warren Dixon

Abstract—In this paper, an adaptive homography-based visual servo tracking controller is developed for the camera-in-hand problem using a quaternion formulation to represent rotation tracking error. The desired trajectory in the tracking problem is encoded by a sequence of images (e.g., a video sequence), and Lyapunov methods are employed to facilitate the control design and the stability analysis. An adaptive estimation law is designed to compensate for the lack of unknown depth information. Experimental results are provided to demonstrate the performance of the visual servo tracking controller.

Index Terms—Adaptive control, Lyapunov methods, quaternion representation, tracking control, visual servoing.

I. INTRODUCTION

VISUAL SERVO control is a broad area of mainstream research. Different approaches have been proposed to solve the visual servo control problem, including image-based, position-based, and combined approaches. Image-based visual servo control consists of a feedback signal that is composed of pure image-space information. Position-based visual servo control utilizes reconstructed Euclidean information in the feedback loop. A review of these approaches is provided in [1]–[3]. Other approaches have been proposed that use combinations of image and reconstructed Euclidean information (e.g., [4]), with the predominate combined method described as 2.5-D or homography-based¹ visual servo control (e.g., [5]–[7], etc.). From a review of the three traditional approaches (see [3] for an in-depth discussion), one significant advantage of homography-based visual servo controllers is that singularities in the image Jacobian are avoided (i.e., the image Jacobian is

typically upper triangular and invertible in homography-based approaches).

After examining the literature, it is evident that much of the previous visual servo controllers have only been designed to address the regulation problem. Motivated by the need for new advancements to meet visual servo tracking applications, previous research has concentrated on developing different path planning techniques [8]–[12]. In [7], a new formulation of the tracking control problem was provided. A homography-based adaptive visual servo controller is developed to enable a robot end-effector to track a prerecorded time-varying reference trajectory determined by a sequence of images. In [7], the axis–angle representation is used to represent the rotation error system. Due to the computational singularity limitation of the axis–angle extraction algorithm (see [13]), rotation angles of $\pm\pi$ were not considered. In [14] and [15], an adaptive control framework is developed that rejects bounded but otherwise unknown disturbances for the tracking problem of a maneuvering target using only visual measurement from a monocular camera fixed on an aerial vehicle, provided that the reference command has an additive intelligent excitation signal.

This paper considers the previously unexamined problem of 6-DOF visual servo tracking control with a nonsingular rotation parameterization. A homography is constructed from image pairs and decomposed via textbook methods (e.g., [16] and [17]) to get a rotation matrix. Once the rotation matrix has been determined, the corresponding unit quaternion can be obtained by globally nonsingular and numerically robust algorithms (see [18]). An error system is constructed in terms of the unit quaternion. An adaptive controller is then developed and proven to make a camera track a desired trajectory that is defined by a sequence of desired images. The controller contains an adaptive feedforward term to compensate for the unknown distance from the camera to the observed features. A quaternion-based Lyapunov function is developed to facilitate the control design and the stability analysis. The contribution of this paper is the development of the quaternion-based rotation tracking controller. Several other homography-based translation controllers (e.g., the controller in [7]) could be combined with the developed rotation controller.

This paper is organized as follows. In Section II, geometric relationships are developed to relate the Euclidean coordinates of the feature points in different camera coordinate frames, and a projective homography is developed to relate the corresponding pixel coordinates. In Section III, the control objective is formulated in terms of unit quaternion representation. In Section IV, the controller is developed, and closed-loop stability analysis is given based on Lyapunov methods. In Section V, a virtual reality testbed for unmanned systems is used to perform a tracking experiment to show the performance

Manuscript received May 21, 2007. Manuscript received in final form March 23, 2008. First published April 28, 2009; current version published December 23, 2009. Recommended by Associate Editor N. Hovakimyan. This work was supported in part by NSF CAREER Award CMS-0547448, by AFOSR Contracts F49620-03-1-0381 and F49620-03-1-0170, by AFRL Contract FA4819-05-D-0011, and by Research Grant US-3715-05 from BARD, the United States–Israel Binational Agricultural Research and Development Fund at the University of Florida, Gainesville.

G. Hu is with the Department of Mechanical and Nuclear Engineering, Kansas State University, Manhattan, KS 66506-5205 USA (e-mail: gqhu@ksu.edu).

N. Gans, N. Fitz-Coy, and W. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: ngans@ufl.edu; nfc@ufl.edu; wdixon@ufl.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2008.2009227

¹The geometric concept of homography is a one-to-one and onto transformation or mapping between two sets of points. In computer vision, homography refers to the mapping between points in two Euclidean planes (Euclidean homography) or to the mapping between points in two images (projective homography).

of the proposed visual servo tracking controller. Concluding remarks are provided in Section VI.

II. MODEL FORMULATION

A. Euclidean Relationships

Without loss of generality,² the subsequent development for the camera-in-hand³ problem is based on the assumption that an object (e.g., the end-effector of a robot manipulator, the wing of an aircraft, and corners of a tumbling satellite) has four coplanar and noncolinear feature points denoted by $O_i \forall i = 1, 2, 3,$ and 4. The plane defined by the four feature points is denoted by π , as shown in Fig. 1. The coordinate frame \mathcal{F} in Fig. 1 is affixed to a camera viewing the object, the stationary coordinate frame \mathcal{F}^* denotes a reference location for the camera, and the coordinate frame \mathcal{F}_d represents the desired time-varying location of the camera. The vectors $\bar{m}_i(t), \bar{m}_i^*, \bar{m}_{di}(t) \in \mathbb{R}^3$ in Fig. 1 are defined as

$$\begin{aligned} \bar{m}_i &\triangleq [x_i(t) \ y_i(t) \ z_i(t)]^T \\ \bar{m}_i^* &\triangleq [x_i^* \ y_i^* \ z_i^*]^T \\ \bar{m}_{di} &\triangleq [x_{di}(t) \ y_{di}(t) \ z_{di}(t)]^T \end{aligned} \quad (1)$$

where $x_i(t), y_i(t), z_i(t) \in \mathbb{R}$, $x_i^*, y_i^*, z_i^* \in \mathbb{R}$, and $x_{di}(t), y_{di}(t), z_{di}(t) \in \mathbb{R}$ denote the Euclidean coordinates of the feature points O_i expressed in the frames \mathcal{F} , \mathcal{F}^* , and \mathcal{F}_d , respectively. From standard Euclidean geometry, relationships between $\bar{m}_i(t), \bar{m}_i^*$, and $\bar{m}_{di}(t)$ can be determined as

$$\bar{m}_i = x_f + R\bar{m}_i^* \quad \bar{m}_{di} = x_{fd} + R_d\bar{m}_i^* \quad (2)$$

where $R(t), R_d(t) \in SO(3)$ denote the orientations of \mathcal{F}^* with respect to \mathcal{F} and \mathcal{F}_d , respectively, and $x_f(t), x_{fd}(t) \in \mathbb{R}^3$ denote translation vectors from \mathcal{F} to \mathcal{F}^* and \mathcal{F}_d to \mathcal{F}^* expressed in the coordinates of \mathcal{F} and \mathcal{F}_d , respectively. As also shown in Fig. 1, $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π expressed in \mathcal{F}^* . The constant distance from the origin of \mathcal{F}^* to π along the unit normal is denoted by $d^* \triangleq n^{*T}\bar{m}_i^* \in \mathbb{R}$.

The normalized Euclidean coordinates, denoted by $m_i(t), m_i^*, m_{di}(t) \in \mathbb{R}^3$, are defined as

$$\begin{aligned} m_i &\triangleq \frac{\bar{m}_i}{z_i} = \begin{bmatrix} \frac{x_i}{z_i} & \frac{y_i}{z_i} & 1 \end{bmatrix}^T \\ m_i^* &\triangleq \frac{\bar{m}_i^*}{z_i^*} = \begin{bmatrix} \frac{x_i^*}{z_i^*} & \frac{y_i^*}{z_i^*} & 1 \end{bmatrix}^T \\ m_{di} &\triangleq \frac{\bar{m}_{di}}{z_{di}} = \begin{bmatrix} \frac{x_{di}}{z_{di}} & \frac{y_{di}}{z_{di}} & 1 \end{bmatrix}^T \end{aligned} \quad (3)$$

²Image processing techniques can often be used to select coplanar and noncoplanar feature points within an image. However, if four coplanar feature points are not available, then the subsequent development can also exploit the virtual parallax method [19], [20] where the noncoplanar points are projected onto a virtual plane.

³The camera-in-hand configuration refers to the case when the camera is attached to a moving robotic system (e.g., held by the robot end-effector). The control method developed in this paper can be extended to the camera-to-hand (i.e., fixed camera) problem [21], which refers to the case when the camera is stationary and observing moving targets (e.g., a fixed camera observing a moving robot end-effector).

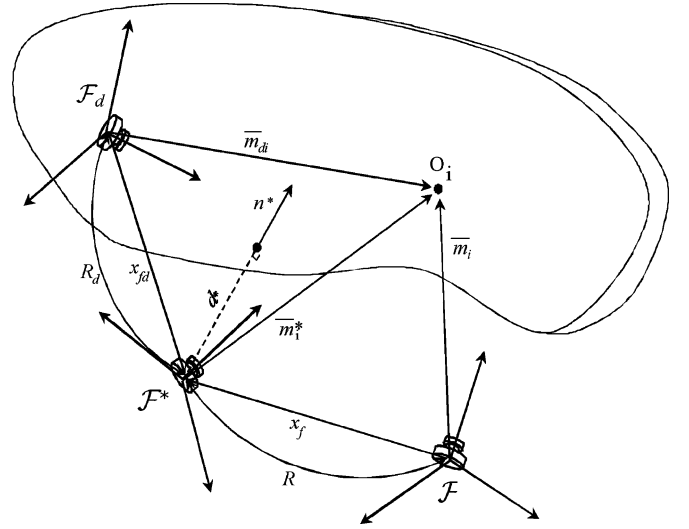


Fig. 1. Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances.

with the standard assumption that $z_i(t), z_i^*, z_{di}(t) > \varepsilon$, where ε is an arbitrarily small positive constant. From (3), the relationships in (2) can be expressed as

$$\begin{aligned} m_i &= \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(R + \frac{x_f}{d^*} n^{*T}\right)}_H m_i^* \\ m_{di} &= \underbrace{\frac{z_i^*}{z_{di}}}_{\alpha_{di}} \underbrace{\left(R_d + \frac{x_{fd}}{d^*} n^{*T}\right)}_{H_d} m_i^* \end{aligned} \quad (4)$$

where $\alpha_i(t), \alpha_{di}(t) \in \mathbb{R}$ are scaling terms and $H(t), H_d(t) \in \mathbb{R}^{3 \times 3}$ denote the Euclidean homographies.

B. Projective Relationships

Each feature point O_i has projected pixel coordinates $p_i(t) \in \mathbb{R}^3$, $p_i^* \in \mathbb{R}^3$, and $p_{di}(t) \in \mathbb{R}^3$ in \mathcal{F} , \mathcal{F}^* , and \mathcal{F}_d , respectively, defined by

$$\begin{aligned} p_i &\triangleq [u_i \ v_i \ 1]^T \\ p_i^* &\triangleq [u_i^* \ v_i^* \ 1]^T \\ p_{di} &\triangleq [u_{di} \ v_{di} \ 1]^T \end{aligned} \quad (5)$$

where $u_i(t), v_i(t), u_i^*, v_i^*, u_{di}(t), v_{di}(t) \in \mathbb{R}$. The projected pixel coordinates $p_i(t), p_i^*$, and $p_{di}(t)$ are related to the normalized task-space coordinates $m_i(t), m_i^*$, and $m_{di}(t)$ by the following global invertible transformation (i.e., the pinhole camera model):

$$p_i = A m_i \quad p_i^* = A m_i^* \quad p_{di} = A m_{di} \quad (6)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a constant, upper triangular, and invertible intrinsic camera-calibration matrix that is explicitly defined as [17]

$$A \triangleq \begin{bmatrix} \alpha & -\alpha \cot \phi & u_0 \\ 0 & \frac{\beta}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

In (7), $u_0, v_0 \in \mathbb{R}$ denote the pixel coordinates of the principal point (i.e., the image center that is defined as the pixel coordinates of the intersection of the optical axis with the image plane), $\alpha, \beta \in \mathbb{R}$ represent the product of the camera pixel dimensions and the focal length, and $\phi \in \mathbb{R}$ is the skew angle between the camera axes.

Assumption 1: The subsequent development is based on the standard assumption that the camera scale factors, focal length, and skew angle (i.e., the diagonal elements of A) are bounded such that the intrinsic camera-calibration matrix remains well conditioned and invertible.

Based on (6), the Euclidean relationship in (4) can be expressed in terms of the image coordinates as

$$\begin{aligned} p_i &= \alpha_i \underbrace{(AHA^{-1})}_{G} p_i^* \\ p_{di} &= \alpha_{di} \underbrace{(AH_dA^{-1})}_{G_d} p_{di}^*. \end{aligned} \quad (8)$$

Based on the feature-point pairs $(p_i^*, p_i(t))$ and $(p_{di}^*, p_{di}(t))$, the projective homography up to a scalar multiple (i.e., G and G_d) can be determined [7]. Various methods can then be applied (e.g., see [22] and [23]) to decompose the Euclidean homographies to obtain the rotation matrices $R(t)$ and $R_d(t)$ and the depth ratios $\alpha_i(t)$ and $\alpha_{di}(t)$.

III. CONTROL OBJECTIVE

The control objective is for a camera to track a desired trajectory that is determined by a sequence of images (e.g., teach by showing [2]). This objective is based on the assumption that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion) and that the camera is calibrated (i.e., A is known). The signals in (5) are the only required measurements to develop the controller.

Among the outcomes of the homography decomposition are the rotation matrices $R(t)$ and $R_d(t)$. From these rotation matrices, several different rotation representations can be utilized to develop the error system. In previous homography-based visual servo control literature, the axis-angle representation has been used to describe the rotation error. In this paper, the unit quaternion parameterization will be used to describe the rotation matrix. This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a global nonsingular parameterization of the corresponding rotation matrices.

The unit quaternion is a 4-D vector, which can be defined as [24]

$$q \triangleq [q_0 \quad q_v^T]^T. \quad (9)$$

In (9), $q_v(t) \triangleq [q_{v1}(t) \quad q_{v2}(t) \quad q_{v3}(t)]^T$ and $q_0(t), q_{vi}(t) \in \mathbb{R} \forall i = 1, 2, 3$. The unit quaternion must also satisfy the following nonlinear constraint:

$$q^T q = 1. \quad (10)$$

Given the rotation matrices $R(t)$ and $R_d(t)$, the corresponding unit quaternions $q(t)$ and $q_d(t)$ can be calculated by using the numerically robust method presented in [18], based on the corresponding relationships

$$R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^\times \quad (11)$$

$$R_d(q_d) = (q_{0d}^2 - q_{vd}^T q_{vd}) I_3 + 2q_{vd} q_{vd}^T - 2q_{0d} q_{vd}^\times \quad (12)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes the skew-symmetric form of the vector $q_v(t)$

$$q_v^\times = \begin{bmatrix} 0 & -q_{v3} & q_{v2} \\ q_{v3} & 0 & -q_{v1} \\ -q_{v2} & q_{v1} & 0 \end{bmatrix} \quad \forall q_v = \begin{bmatrix} q_{v1} \\ q_{v2} \\ q_{v3} \end{bmatrix}. \quad (13)$$

To quantify the error between the actual and desired camera orientations, the mismatch between the rotation matrices $R(t)$ and $R_d(t)$ is defined as

$$\tilde{R} = RR_d^T. \quad (14)$$

Based on (11), (12), and (14), $\tilde{R}(t)$ can be expressed as

$$\tilde{R} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times \quad (15)$$

where the error quaternion $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as

$$\begin{aligned} \tilde{q}_0 &= q_0 q_{0d} + q_v^T q_{vd} \\ \tilde{q}_v &= q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd}. \end{aligned} \quad (16)$$

The definition of $\tilde{q}_0(t)$ and $\tilde{q}_v(t)$ in (16) makes $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ a unit quaternion based on the fact that $q(t)$ and $q_d(t)$ are two unit quaternions.

The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as [6], [7]

$$e = p_e - p_{ed} \quad (17)$$

where $p_e(t), p_{ed}(t) \in \mathbb{R}^3$ are defined, respectively, as

$$\begin{aligned} p_e &= [u_i \quad v_i \quad -\ln(\alpha_i)]^T \\ p_{ed} &= [u_{di} \quad v_{di} \quad -\ln(\alpha_{di})]^T \end{aligned} \quad (18)$$

where $i \in \{1, \dots, 4\}$.

In the Euclidean space (see Fig. 1), the tracking objective can be quantified as

$$\tilde{R}(t) \rightarrow I_3, \quad \text{as } t \rightarrow \infty \quad (19)$$

$$\|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (20)$$

Since $\tilde{q}(t)$ is a unit quaternion, (15), (16), and (19) can be used to quantify the rotation tracking objective as the desire to regulate $\tilde{q}_v(t)$ as

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (21)$$

The subsequent section will target the control development based on the objectives in (20) and (21).

IV. CONTROL DEVELOPMENT

A. Open-Loop Error System

Define the actual angular velocity of the camera expressed in \mathcal{F} as $\omega_c(t) \in \mathbb{R}^3$, the desired angular velocity of the camera expressed in \mathcal{F}_d as $\omega_{cd}(t) \in \mathbb{R}^3$, and the relative angular velocity of the camera with respect to \mathcal{F}_d expressed in \mathcal{F} as $\tilde{\omega}_c(t) \in \mathbb{R}^3$, where

$$\tilde{\omega}_c = \omega_c - \tilde{R}\omega_{cd}. \quad (22)$$

The time derivatives of $q(t)$ and $q_d(t)$ can be developed, respectively, as [24]

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c \quad (23)$$

$$\begin{bmatrix} \dot{q}_{0d} \\ \dot{q}_{vd} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_{vd}^T \\ q_{0d} I_3 + q_{vd}^\times \end{bmatrix} \omega_{cd}. \quad (24)$$

Remark 1: As stated in [7, Remark 3], a sufficiently smooth function can be used to fit a sequence of feature points to generate the desired trajectory $p_{di}(t)$; hence, it is assumed that $p_{ed}(t)$ and $\dot{p}_{ed}(t)$ are bounded functions of time. From the projective homography introduced in (8), $p_{di}(t)$ can be expressed in terms of the *a priori* known functions $\alpha_{di}(t)$, $H_d(t)$, $R_d(t)$, and $x_{fd}(t)/d^*$. Since these signals can be obtained from the prerecorded sequence of images, sufficiently smooth functions can also be generated for these signals. For example, the feature-point trajectories could be fit with a sufficiently smooth spline function, or a curve fit could be applied to the homography decomposition. Hence, in practice, the *a priori* developed smooth functions $\alpha_{di}(t)$, $R_d(t)$, and $x_{fd}(t)/d^*$ can be constructed as bounded functions with bounded time derivatives. Based on the assumption that $R_d(t)$ is a bounded first-order differentiable function with a bounded derivative, the algorithm for computing quaternions in [25] can be used to conclude that $(q_{0d}(t), q_{vd}^T(t))^T$ are bounded first-order differentiable functions with a bounded derivative; hence, $(q_{0d}(t), q_{vd}^T(t))^T$ and $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are bounded. In the subsequent tracking control development, the desired signals $\dot{p}_{ed}(t)$ and $\dot{q}_{vd}(t)$ will be used as a feedforward control term. These signals can be obtained from an analytical derivative of a curve fit of the feature-point trajectory or could be numerically computed by a backward difference algorithm. To avoid the computational singularity in $\theta_d(t)$, the desired trajectory in [7] was generated by carefully choosing the smooth function such that the workspace is limited to $(-\pi, \pi)$. Unlike [7], the use of the quaternion alleviates the restriction on the desired trajectory $p_d(t)$.

Based on (24), the signal $\omega_{cd}(t)$ can be calculated as

$$\omega_{cd} = 2(q_{0d}\dot{q}_{vd} - q_{vd}\dot{q}_{0d}) - 2q_{vd}^\times \dot{q}_{vd} \quad (25)$$

where $(q_{0d}(t), q_{vd}^T(t))^T$ and $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are bounded, so $\omega_{cd}(t)$ is also bounded. Based on (15), (16), (23), and (24), the open-loop rotation error system can be developed as

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_c - \tilde{R}\omega_{cd}) \quad (26)$$

where $\tilde{q}(t) = (\tilde{q}_0(t), \tilde{q}_v^T(t))^T$.

Based on (5), (6), (17), and (22), and the fact that [26]

$$\dot{\tilde{m}}_i = -v_c + \tilde{m}_i^\times \omega_c \quad (27)$$

where $v_c(t) \in \mathbb{R}^3$ denotes the actual linear velocity of the camera expressed in \mathcal{F} , the open-loop translation error system can be derived as [7]

$$z_i^* \dot{e} = -\alpha_i L_v v_c + (L_v m_i^\times \omega_c - \dot{p}_{ed}) z_i^* \quad (28)$$

where $L_v(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$L_v = \left(A - \begin{bmatrix} 0 & 0 & u_0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & -\frac{x_i}{z_i} \\ 0 & 1 & -\frac{y_i}{z_i} \\ 0 & 0 & 1 \end{bmatrix}. \quad (29)$$

The auxiliary term $L_v(t)$ is an invertible upper triangular matrix.

B. Closed-Loop Error System

Based on the open-loop rotation error system in (26) and the subsequent Lyapunov-based stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega (I_3 + \tilde{q}_v^\times)^{-1} \tilde{q}_v + \tilde{R}\omega_{cd} = -K_\omega \tilde{q}_v + \tilde{R}\omega_{cd} \quad (30)$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. Based on (22), (26), and (30), the closed-loop rotation error system can be determined as

$$\begin{aligned} \dot{\tilde{q}}_0 &= \frac{1}{2} \tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2} (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v. \end{aligned} \quad (31)$$

The contribution of this paper is the development of the quaternion-based rotation tracking controller. Several other homography-based translation controllers could be combined with the developed rotation controller. For completeness, we illustrate how the translational controller and adaptive update law in [7] can be used to complete the 6-DOF tracking result.

Based on (28), the translation control input $v_c(t)$ is designed as [7]

$$v_c = \frac{1}{\alpha_i} L_v^{-1} (K_v e + \hat{z}_i^* (L_v m_i^\times \omega_c - \dot{p}_{ed})) \quad (32)$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. In (32), the parameter estimate $\hat{z}_i^*(t) \in \mathbb{R}$ for the unknown constant z_i^* is defined as [7]

$$\dot{\hat{z}}_i^* = \gamma e^T (L_v m_i^\times \omega_c - \dot{p}_{ed}) \quad (33)$$

where $\gamma \in \mathbb{R}$ denotes a positive constant adaptation gain. The controller in (32) does not exhibit a singularity since $L_v(t)$ is invertible and $\alpha_i(t) > 0$.

Based on (28) and (32), the translation closed-loop error system can be listed as

$$z_i^* \dot{e} = -K_v e + (L_v m_i^\times \omega_c - \dot{p}_{ed}) z_i^* \quad (34)$$

where $\tilde{z}_i^*(t) \in \mathbb{R}$ denotes the following parameter estimation error:

$$\tilde{z}_i^* = z_i^* - \hat{z}_i^*. \quad (35)$$

C. Stability Analysis

Theorem 1: The controller given in (30) and (32), along with the adaptive update law in (33), ensures asymptotic tracking in the sense that

$$\|\tilde{q}_v(t)\| \rightarrow 0, \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (36)$$

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable nonnegative function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{z_i^*}{2} e^T e + \frac{1}{2\gamma} z_i^{*2}. \quad (37)$$

The time derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v \\ &\quad + e^T (-K_v e + (L_v m_i^\times \omega_c - \dot{p}_{ed}) \tilde{z}_i^*) \\ &\quad - \tilde{z}_i^{*T} (L_v m_i^\times \omega_c - \dot{p}_{ed}) \\ &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times + (1 - \tilde{q}_0) I_3) K_\omega \tilde{q}_v - e^T K_v e \\ &= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e \end{aligned} \quad (38)$$

where (31) and (33)–(35) were utilized. It can be seen from (38) that $\dot{V}(t)$ is negative semidefinite. Based on (37) and (38), $e(t)$, $\tilde{q}_v(t)$, $\tilde{q}_0(t)$, $\tilde{z}_i^*(t) \in \mathcal{L}_\infty$ and $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_2$. Furthermore, it can be proved that $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$ [21]. Since $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$, Barbalat's lemma [27] can be used to conclude the result given in (36).

V. EXPERIMENT RESULTS

Experiments were performed to test robustness and performance in the presence of signal noise, measurement error, calibration error, etc. The experiments were performed in a virtual reality testbed in which a 3-D environment is projected onto large monitors and viewed by a physical camera. Communication between the camera and control processing computers and the environment rendering computers allows closed-loop control of the virtual scene.

A. Testbed Configurations

A block diagram describing the experimental testbed is shown in Fig. 2. The experiment testbed is based on a virtual environment generated by a virtual reality simulator, which is composed of workstations and a database server running virtual reality software. The virtual reality simulator is currently capable of displaying three simultaneous displays. A picture of the displays is shown in Fig. 3.

The virtual reality simulator utilizes MultiGen-Paradigm's Vega Prime, an OpenGL-based commercial software package for Microsoft Windows. The virtual environment is an accurate recreation of a U.S. Army urban warfare training ground. The environment has a dense polygon count, detailed textures, and the effects of soft shadows, resulting in very realistic images, and can be seen in Fig. 3.

The visual sensor in the experiment is a Sony XCD-710CR color firewire camera fitted with a 12.5-mm lens. The camera captures the images on the large screens, as shown in Fig. 3. The images are processed in a vision processing workstation. An application written in C++ acquires images from the camera and

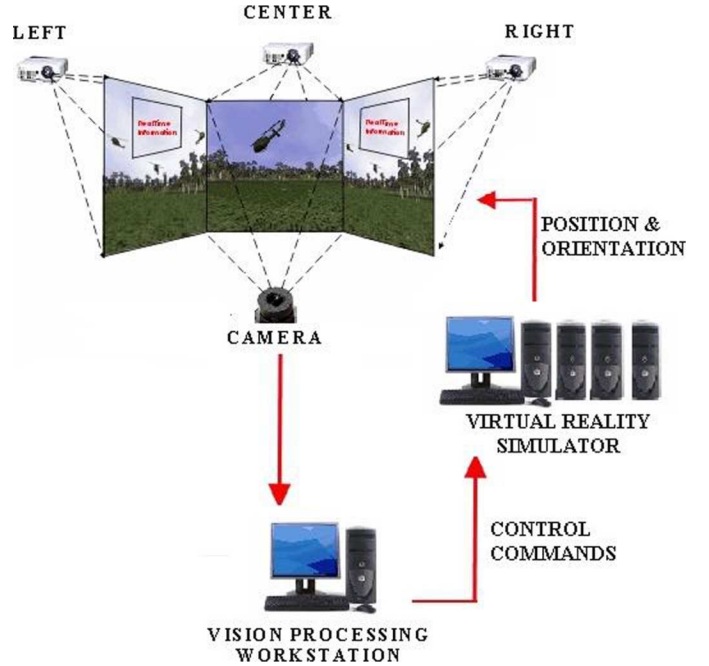


Fig. 2. Block diagram of the hardware-in-the-loop experiment testbed.

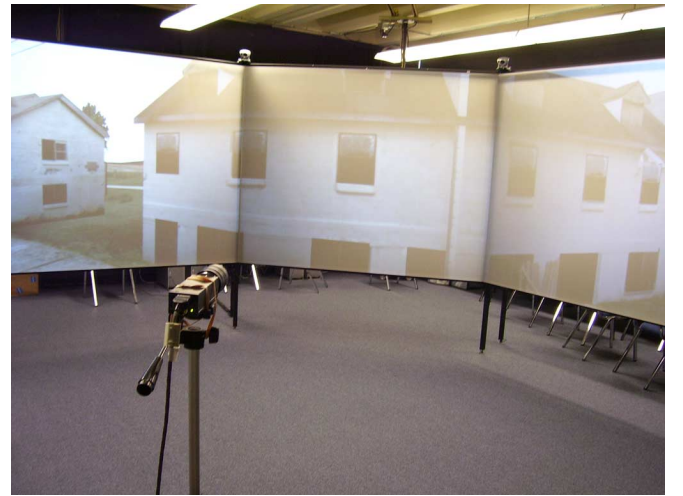


Fig. 3. Sony XCD-710CR color firewire camera pointed at the virtual environment.

processes the images to locate and track the feature points (the initial feature points were chosen manually, and then, the application will identify and track the feature points on its own). The C++ application generates the current and desired pixel coordinates, which can be used to formulate the control command.

The controller developed in Section IV also runs in this workstation, which is programmed in Matlab. The Matlab application exchanges data with the image processing application via shared memory buffers. The Matlab application reads the current and desired pixel coordinates from the shared memory buffers and writes linear and angular camera velocity inputs into the shared memory buffers. The C++ application writes the current and desired pixel coordinates into the shared memory and reads camera velocity input from the shared memory buffer.

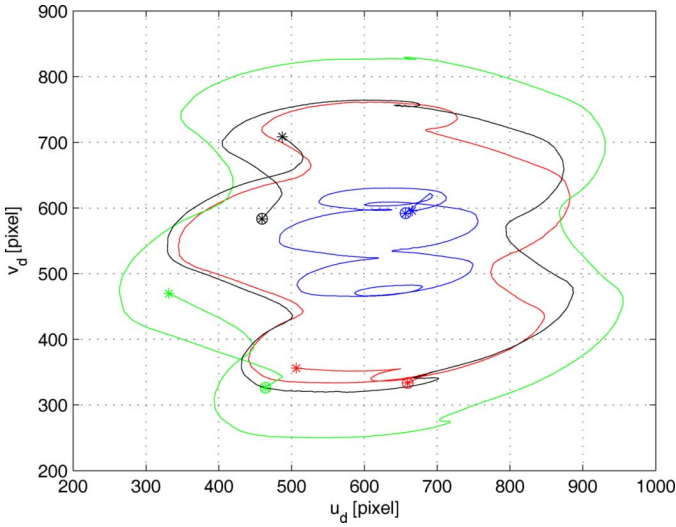


Fig. 4. Desired trajectory of $p_d(t)$ in the image space. In the figure, “O” denotes the initial positions of the four feature points in the image, and “*” denotes the corresponding final positions of the feature points.

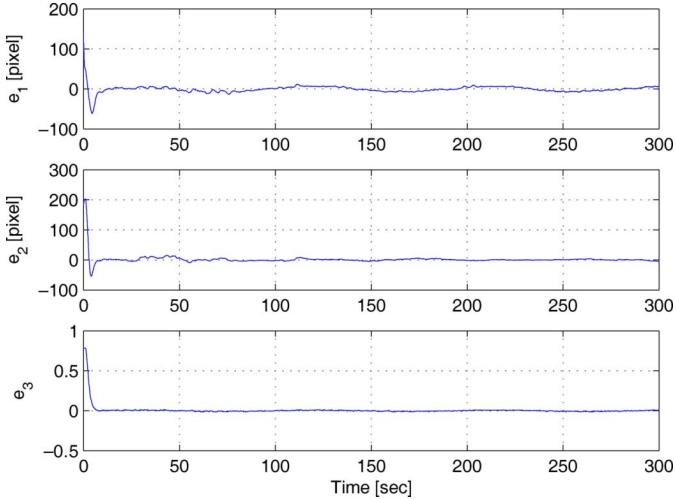


Fig. 5. Translational error $e(t)$ in the tracking experiment.

The linear and angular camera velocity control inputs are sent from the vision processing workstation to the virtual reality simulator via a TCP socket connection. This development makes extensive use of Intel’s Open Source Computer Vision Library [28] and the GNU Scientific Library [29].

Algorithms, such as the homography decomposition, are implemented as if the virtual environment is a true 3-D scene which the physical camera is viewing. Of course, the camera does not look at the 3-D scene directly. The camera views consist of a 3-D scene that is projected onto a 2-D plane and then onto the image plane. That is, the projective homography needed for control exists between the *on-screen* current image and the *on-screen* goal image, but what we are given are the *camera views of the on-screen* images. Thus, there exists an additional transformation between the points on the screen and the points in the camera image. A screen–camera-calibration matrix G_{cs} describes this relationship.

Every point on the screen corresponds to only one point in the image. Thus, the constant matrix G_{cs} is a homography

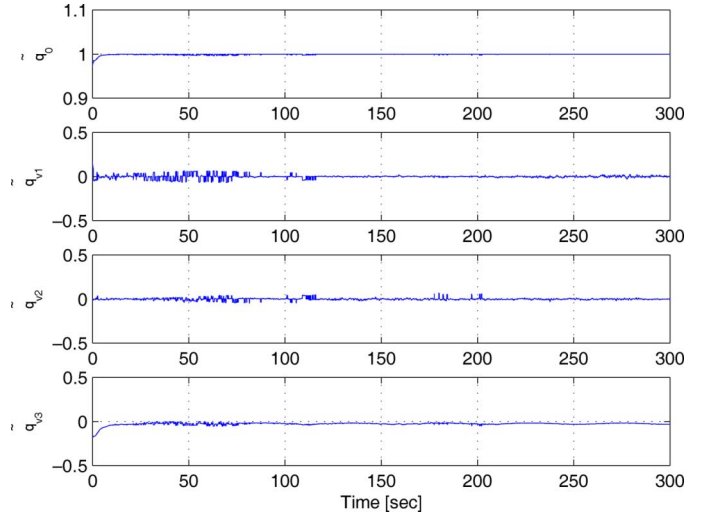


Fig. 6. Rotational quaternion error $\bar{q}(t)$ in the tracking experiment.

and can be determined through a calibration procedure. The screen–camera-calibration matrix is determined to be

$$G_{cs} = \begin{bmatrix} 0.9141 & 0.0039 & -90.9065 \\ -0.0375 & 0.9358 & -50.7003 \\ 0 & 0 & 1 \end{bmatrix}.$$

In addition to G_{cs} , the camera-calibration matrix A , corresponding to the virtual camera within Vega Prime, is still required. This matrix can be determined from the settings of the virtual reality program. In this experiment, A was determined to be

$$A = \begin{bmatrix} 1545.1 & 0 & 640 \\ 0 & 1545.1 & 512 \\ 0 & 0 & 1 \end{bmatrix}.$$

B. Tracking Experiment

The control objective in this tracking experiment is to send control commands to the virtual reality simulator such that the current pose of the feature points tracks the desired pose trajectory. The desired trajectory is determined from a recorded video (a sequence of images) of a target that performs both translation and rotation motion. The rotation in the desired trajectory is more than 360° , as shown in Fig. 4. No smoothing of the desired recorded feature-point trajectory was required in the experiment except for filtering the backward time derivative for $\dot{p}_{ed}(t)$. During the experiment, the images from the camera are processed with a frame rate of approximately 15 frames/s.

The control gains K_ω in (30) and K_v in (32), and the adaptation gain γ in (33) were selected, respectively, as

$$K_\omega = \text{diag}\{0.5, 0.5, 4\} \quad K_v = \text{diag}\{1, 0.5, 0.5\} \\ \gamma = 0.0003.$$

The resulting translational and rotational errors are shown in Figs. 5 and 6. The tracking error between the current and desired image-space trajectories is shown in Fig. 7.

In the tracking experiment, the steady-state error is approximately $[12 \text{ [pixel]} \quad 9 \text{ [pixel]} \quad 0.01]$. Note that this pixel error

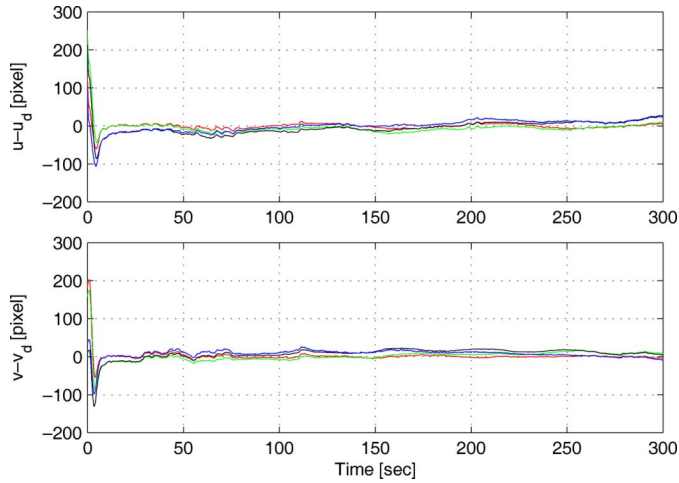


Fig. 7. Tracking error $p(t) - p_d(t)$ (in pixels) of the four feature points in the tracking experiment. The upper figure is for the $u(t) - u_d(t)$ component, and the bottom figure is for the $v(t) - v_d(t)$ component.

represents less than 1.5% of the image dimensions. This steady-state error is caused by the image noise and the camera-calibration error in the testbed. To find the tracking control error, two homographies are computed, i.e., the first between the reference image and the current image, and the second between the reference image and the desired image. Image noise affects each homography computation, so the error is magnified when comparing the noisy results from one homography (e.g., the homography computed from the actual images compared to the reference image) to the noisy results from another homography decomposition (e.g., the homography computed from the desired images compared to the reference image). Also, image noise is present in the derivative of the desired pixel coordinates, which is used as a feedforward term in the tracking controller. The derivative of the desired image trajectory was computed from a backward difference algorithm that was filtered by a Butterworth filter with a cutoff frequency of 5 Hz to reduce noise. Communication between the controller and the virtual reality system occurs via a TCP socket, introducing some amount of latency into the system.

C. Regulation Experiment

When the desired pose is a constant, the tracking problem becomes a regulation problem. The control objective in the regulation experiment is to send control commands to the virtual reality simulator such that the current pose of the feature points is regulated to the desired set pose. For the regulation problem (similar to the target tracking problem where the objective is to keep the feature points of a target in the center of the image), the derivative of the desired pixel coordinates is equal to zero, and only one homography is computed between the current image and the desired image. The influence of the image noise and calibration error is significantly reduced in this case.

In the experiment, the control gains K_ω in (30) and K_v in (32), and the adaptation gain γ in (33) were selected, respectively, as

$$K_\omega = \text{diag}\{0.4, 0.4, 0.9\} \quad K_v = \text{diag}\{0.5, 0.25, 0.25\} \\ \gamma = 0.005.$$

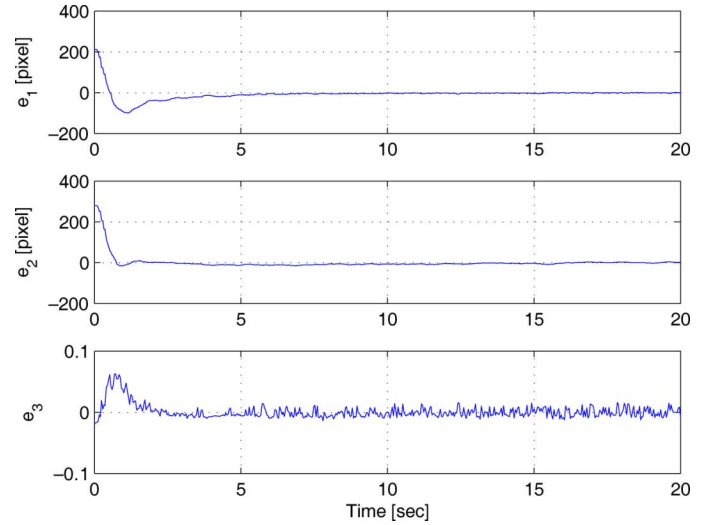


Fig. 8. Translational error $e(t)$ in the regulation experiment.

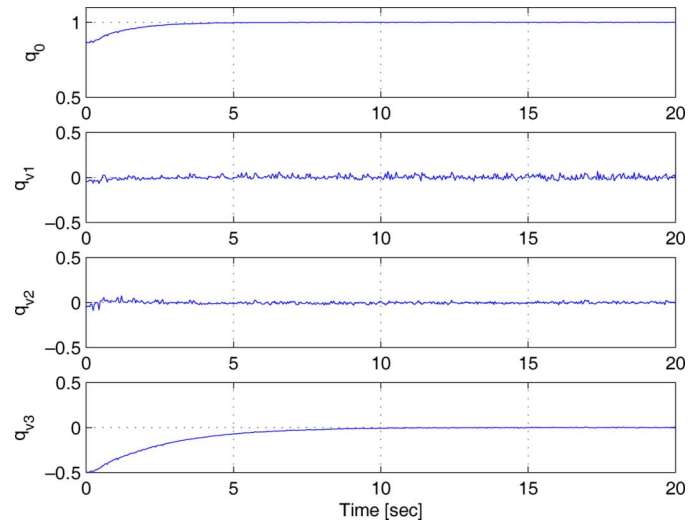


Fig. 9. Rotational quaternion error $q(t)$ in the regulation experiment.

The resulting translational and rotational errors are shown in Figs. 8 and 9, respectively. The regulation error between the current and desired set image-space poses is shown in Fig. 10.

VI. CONCLUSION

An adaptive visual servo tracking controller is presented that achieves asymptotic tracking of a desired trajectory determined by a sequence of images. The presented controller is formulated based on a quaternion representation of a rotation matrix that is computed from a homography decomposition. A homography-based rotation and translation controller is proven to yield the tracking result through a Lyapunov-based stability analysis. Based on the results for the camera-in-hand configuration problem, a camera-to-hand extension can be used to enable an actuated planar patch (e.g., the end-effector of a robot manipulator, a satellite, etc.) to track a desired trajectory determined by a sequence of images.

From a theoretical perspective, the homography-based approach is one of the few methods to enable 6-DOF visual servo control with a single camera without knowledge of the

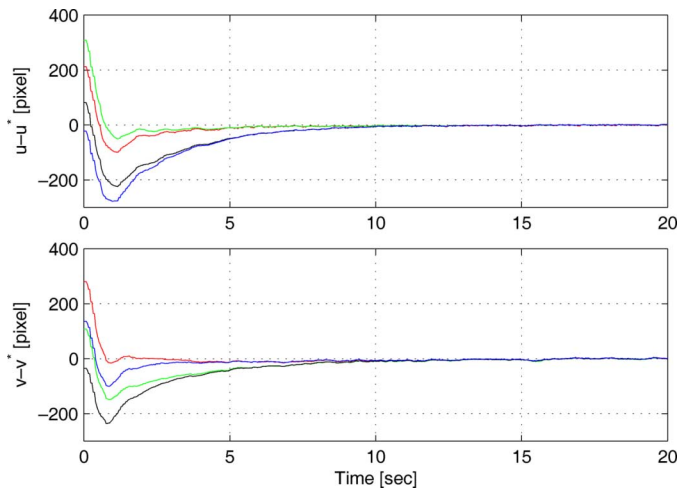


Fig. 10. Regulation error $p(t) - p^*$ (in pixels) of the four feature points in the regulation experiment. The upper figure is for the $u(t) - u^*(t)$ component, and the bottom figure is for the $v(t) - v^*(t)$ component.

target. The strength of the homography-based approach is that the unknown time-varying distance to the target along the camera focal axis can be related to an unknown static constant. This relationship facilitates the ability to use adaptive control methods to compensate for the unknown constant, thereby compensating for the unknown time-varying depth. The contribution of this result is the use of a quaternion representation to eliminate the potential numerical singularity caused by the typically used axis-angle representation when following a desired trajectory through specific rotations. The experimental results demonstrated the capability of the developed controller to track a given desired trajectory. Despite uncertainty in the camera-calibration feature-point tracking noise, and latency in the virtual environment, the experimental results were able to yield acceptable tracking results of less than 1.5% of the image dimensions. For the regulation problem, the noise in the system was less prevalent because the derivative of the image trajectory was not included in the controller, and the output from the homography was compared to static data versus noisy desired trajectory data. For the regulation problem, the rotation and translation errors were shown to converge to zero despite the lack of precise camera and camera-to-screen calibration.

REFERENCES

- [1] G. D. Hager and S. Hutchinson, "Special section on vision-based control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 649–650, Oct. 1996.
- [2] 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.
- [3] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
- [4] P. Corke and S. Hutchinson, "A new hybrid image-based visual servo control scheme," in *Proc. IEEE Conf. Decision Control*, 2000, pp. 2521–2527.
- [5] E. Malis, F. Chaumette, and S. Bodet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
- [6] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 176–186, Apr. 2002.
- [7] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 5, pp. 814–825, Sep. 2005.
- [8] N. J. Cowan and D. Koditschek, "Planar image-based visual servoing as a navigation problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 1720–1725.
- [9] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 521–533, Aug. 2002.
- [10] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 534–549, Aug. 2002.
- [11] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
- [12] A. Ruf, M. Tonko, R. Horaud, and H.-H. Nagel, "Visual tracking of an end-effector by adaptive kinematic prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1997, pp. 893–898.
- [13] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.
- [14] V. Stepanyan and N. Hovakimyan, "Adaptive disturbance rejection controller for visual tracking of a maneuvering target," *J. Guid. Control Dyn.*, vol. 30, no. 4, pp. 1090–1106, Jul./Aug. 2007.
- [15] V. Stepanyan and N. Hovakimyan, "Visual tracking of a maneuvering target," *J. Guid. Control Dyn.*, vol. 31, no. 1, pp. 66–80, 2008.
- [16] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.
- [17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York: Cambridge Univ. Press, 2000.
- [18] M. Shuster, "A survey of attitude representations," *J. Astronaut. Sci.*, vol. 41, no. 4, pp. 439–518, Oct. 1993.
- [19] B. Boufama and R. Mohr, "Epipole and fundamental matrix estimation using virtual parallax," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1995, pp. 1030–1036.
- [20] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 79–97, Jun. 2000.
- [21] G. Hu, S. Gupta, N. Fitz-Coy, and W. E. Dixon, "Lyapunov-based visual servo tracking control via a quaternion formulation," in *Proc. IEEE Conf. Decision Control*, 2006, pp. 3861–3866.
- [22] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recogn. Artif. Intell.*, vol. 2, no. 3, pp. 485–508, 1988.
- [23] Z. Zhang and A. R. Hanson, "Scaled Euclidean 3D reconstruction based on externally uncalibrated cameras," in *Proc. IEEE Symp. Comput. Vis.*, 1995, pp. 37–42.
- [24] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Non-linear Control of Engineering Systems: A Lyapunov-Based Approach*. Boston, MA: Birkhäuser, 2003.
- [25] G. Hu, W. E. Dixon, S. Gupta, and N. Fitz-Coy, "A quaternion formulation for homography-based visual servo control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2391–2396.
- [26] Y. Fang, A. Behal, W. E. Dixon, and D. M. Dawson, "Adaptive 2.5D visual servoing of kinematically redundant robot manipulators," in *Proc. IEEE Conf. Decision Control*, 2002, pp. 2860–2865.
- [27] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [28] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 122–125, Nov. 2000, 120.
- [29] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi, *GNU Scientific Library: Reference Manual*. Bristol, U.K.: Netw. Theory Ltd., 2005.