Geometric Visual Servoing

Noah J. Cowan, *Member, IEEE,* and Dong Eui Chang

***Abstract*—This paper presents a global diffeomorphism from a *visible set* of rigid-body configurations, a subset of SE(3), to an *image space*. Using the diffeomorphism, we develop an image-based, essentially global, dynamic visual servoing algorithm that keeps features in the field of view and avoids self-occlusions. The approach is geometric in the sense that the visible set and its corresponding image are differentiable manifolds, and the diffeomorphism is global. The mapping to image space and the resulting Jacobian rely on a specific target geometry, a sphere with a known radius marked with an "arrow" feature point. The paper presents simulation experiments for a more typical visual target comprised of a collection of isolated feature points. In this setting, the diffeomorphism to image space is approximate, nevertheless, the simulations converge for a wide variety of target geometries and initial conditions.**

***Index Terms*—Geometric control, vision-based control, visual servoing (VS).**

I. Introduction

FOR a specially designed target viewed by a perspective camera, we construct a *diffeomorphism*, a smooth and smoothly invertible transformation, from a visible subset of SE(3) to an image space. We also compute the resulting Jacobian matrix, which does not explicitly depend on feature depths.

The image Jacobian, which relates rigid body motion to the motion of image features, usually involves a mixture of image and Cartesian variables [1]. Were it possible, constructing the Jacobian without Cartesian variables would enable true visual servoing (VS), closing feedback loops completely in the image plane. The diffeomorphism in this paper enables the next best thing: a model-based Jacobian expressed in terms of image variables, at the expense of requiring an engineered visual target with one known parameter. That parameter is used explicitly in the computation of the Jacobian matrix, though the controller we present is (empirically) robust to substantial error in that parameter, as well as the model structure.

Building on the framework in [2], we characterize a so-called visible set, a subset of the rigid transformations of a body relative to a camera that keep all features visible, and then calculate its image under the camera diffeomorphism. In this paper, we present the image Jacobian in a coordinate-free manner, without resorting to local coordinates. Finally, we apply the machinery of *navigation functions* (NFs) [3], a special class of artificial potential functions, to create global, dynamic VS systems that keep features fully visible.

VS requires representing rigid motions in terms of visually measured quantities. We review three candidate approaches: "2-D," "3-D," and "2 1/2-D." For an experimental comparison that highlights some of the respective advantages, see Gans *et al.* [4].

Classical 2-D VS algorithms attempt to drive the projection of an arbitrary set of rigidly constrained feature points in straight lines to their visual goals, by projecting the image error through a pseudoinverse of the image Jacobian matrix [1]. However, the underlying rigid motion constrains the movement of projected features, rendering image-based control and motion planning in image space, challenging for large deviations from the goal. To overcome these challenges, Corke and Hutchinson [5] created a 2-D kinematic algorithm for six-degree-of-freedom (DOF) VS that (empirically) has a very large basin of attraction, while keeping features in the field of view (FOV). They classify their algorithm as partitioned, since it controls some DOFs directly on the basis of abstracted features, such as the total projected area and the orientation of two distinguished features. Their choice of features helped motivate the choice of features used in this paper, and we show that a formal diffeomorphism to the abstracted image features enables the computation of an image-based Jacobian matrix. Path planning [6], [7] together with local image-based feedback [7] provides another avenue to avoid local minima.

One may recover the complete pose of a camera with respect to a target by exploiting a model of the target [8]. Vision-based controllers using full pose reconstruction are referred to as 3-D VS algorithms. Pose reconstruction on the current and desired views of a scene produces two pose estimates of the camera. Since parametric uncertainties affect both the current and desired pose estimates in the same way, 3-D servo laws are known to be robust to camera and target calibration uncertainties (though, to the best of our knowledge, there are no formal bounds on the uncertainties that can be tolerated). One of the challenges to 3-D methods involves representing visibility obstacles, such as self-occlusions or a finite FOV [2].

Given a sufficient number of feature points, the relative pose, up to a scale in translation, between two views may be determined without exploiting a geometrical model of the points, and such approaches have been dubbed 2 1/2-D VS [9]. Researchers developed 6-DOF kinematic 2 1/2-D VS algorithms robust to calibration uncertainty [9]–[12], with large basins of attraction. Malis and Chaumette also incorporated image FOV constraints for a single base point of interest [9], without requiring an explicit model of the visual target or depth estimates. Recently, Chesi and Vicino [13] proposed a method to control eye-in-hand

Manuscript received August 24, 2004; revised February 24, 2005. This paper was recommended for publication by Associate Editor F. Chaumette and Editor S. Hutchinson upon evaluation of the reviewers' comments. This paper was presented in part at the Conference on Control Problems in Robotics, Las Vegas, NV, December 2002.

N. J. Cowan is with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: ncowan@jhu.edu).

D. E. Chang was with the Centre Automatique et Systèmes, Ecole Nationale Supérieure des Mines de Paris, 75272 Paris Cedex 06, France. He is now with the Department of Applied Mathematics, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: dechang@math.uwaterloo.ca).

Digital Object Identifier 10.1109/TRO.2005.853491

servoing that causes the camera to follow nearly circular arcs, while maintaining the features within the FOV, and Zanne *et al.* describe a global planner and control system that maintains feature visibility for kinematic eye-in-hand VS [12].

This paper uses an explicit target model to compute an image Jacobian, without the need to directly recover the depth of each feature in the scene. Zhang and Ostrowski [6], [14] designed a similar model-based VS strategy to control a blimp in three DOFs. When applicable, the *model-based, geometric VS framework* presented in this paper provides several enhancements over existing methods:

1) application of NFs enables control of *dynamic* (as well as kinematic) plants;
2) eye-in-hand and fixed-camera VS setups fall within the same mathematical framework;
3) essentially global[1] dynamical convergence is guaranteed, while maintaining feature visibility.

While we suspect at least some robustness to parametric error (a suspicion supported by numerical experiments), significant work remains to show this formally, and to characterize how much parametric error can be tolerated.

II. Six-DOF Diffeomorphism to Image Space

We posit a specific visual target, designed as follows.

1) **A spherical body, of radius** ϱ. As the body moves away from the camera, its projection gets smaller. Roughly speaking, the position and size of the body's image encodes the position of the center of the body relative to the camera.
2) **A single point on the body**. Adding a visible point to the body breaks the visual symmetry, allowing us to resolve two rotational DOFs from the location of the feature point on the image.
3) **A unit vector tangent to the body**. The final DOF is resolved by considering the orientation on the image of a projected vector attached to our feature point on the body.

Such a target might be useful for factories, spacecraft docking stations, and helicopter landing beacons. For unstructured settings, Section V demonstrates how to apply the results to a visual target consisting of a "cloud of points."

Zhang and Ostrowski [6], [14] developed the idea of projecting a spherical body to an image plane for 3-DOF VS of a blimp relative to a large ball. This paper builds on that work by incorporating additional markings on the body to encode rotational information. The features used in this paper are also motivated by the partitioned features used by Corke and Hutchinson [5].

A. Notation and Definitions

The notation for this paper is summarized in Table I.

Consider a perspective ("pinhole") camera with right-handed rigid frame $\mathcal{F}_c$ such that its origin is located at the pinhole (or optical center), and the camera frame's z axis is aligned with the optical axis. The camera observes features of a rigid body, affixed with rigid frame $\mathcal{F}_b$. Let

$$H = (R, d) \in \mathrm{SE}(3), \quad \text{where} \quad R \in \mathrm{SO}(3),\ d \in \mathbb{R}^3$$

denote the rigid transformation of $\mathcal{F}_b$ relative to $\mathcal{F}_c$. A point expressed with respect to the body frame as p^b appears as $p^c = Rp^b + d$ with respect to the camera frame. Similarly, if $v^b \in \mathbb{R}^3$ is a vector in the body frame, then $v^c = Rv^b$ is the same vector with respect to the camera frame. In an abuse of notation, we will often write $p^c = Hp^b$, implicitly using homogeneous coordinates. One last notational abuse entails the conflation of a point p with the vector from the origin of a frame $\mathcal{F}_a$ to that point. We will write both point and vector as p^a.

Let $p = (p_1, p_2, p_3) \in \mathbb{R}^3$ denote a point expressed in the camera frame. Assuming a fully calibrated camera, recall that the image-plane projection is given by $(u, v) = (1/p_3)(p_1, p_2) \in \mathbb{R}^2$. This can be normalized to define a unit vector $(u, v, 1)/\sqrt{1 + u^2 + v^2} = p/||p||$, and thus, for simplicity, we consider a panoramic pinhole model given by

$$\begin{aligned} \pi &: \left(\mathbb{R}^3 - \{0\}\right) \to \mathrm{S}^2 \\ &: p \mapsto \frac{p}{||p||} \end{aligned} \tag{1}$$

where $\mathrm{S}^2 = \{v \in \mathbb{R}^3 : v \cdot v = 1\} \subset \mathbb{R}^3$. To keep features within a finite FOV, one may introduce an appropriate image-space obstacle into the controller design (see Section IV).

Given a manifold $\mathcal{X}$, then $T_x\mathcal{X}$ is the tangent space to the manifold at the point $x \in \mathcal{X}$ and $T\mathcal{X} = \cup_{x \in \mathcal{X}} T_x\mathcal{X}$ is the tangent bundle. Consider a point $p \in \mathbb{R}^3$, together with a vector $a \in T_p\mathbb{R}^3$, located at p. Together, the pair define a point in the tangent bundle of $\mathbb{R}^3$, namely, $(p, a) \in T\mathbb{R}^3$. One can show that the tangent mapping $T\pi : T\mathbb{R}^3 \to T\mathrm{S}^2$ is given by

$$T\pi : (p, a) \mapsto (\pi(p), T_p\pi \cdot a) \in T\mathrm{S}^2 \tag{2}$$

where

$$T_p\pi = \frac{1}{||p||}\left(I_{3\times 3} - \pi(p)\pi(p)^T\right) \tag{3}$$

TABLE I
List of Symbols

Symbol	Description
$e_1, e_2, \ldots \in \mathbb{R}^n$	standard basis
$\mathcal{F}_c, \mathcal{F}_b$	camera / body frame
p^a, v^a	point and vector relative to $\mathcal{F}_a$
$H = (R, d) \in \mathrm{SE}(3)$	rigid transformation of $\mathcal{F}_b$ relative to $\mathcal{F}_c$
$r_i,\ i = 1, 2, 3$	columns of rotation, R
$\nu : \mathrm{SE}(3) \to \mathbb{R}$	visibility function, (8)
$\mathcal{V} \subset \mathrm{SE}(3)$	visible configurations, (8)
$\pi : (\mathbb{R}^3 - \{0\}) \to \mathrm{S}^2$	image projection model, (1)
$\lambda \in (0, 1)$	projected body radius, (6)
$s \in \mathrm{S}^2$	projected centroid, (6)
$Q \in \mathrm{SO}(3)$	projected rotation, (11)
$y = (Q, \lambda, s) = c(H)$	camera map, (12)
$\mathcal{Y} \subset \mathrm{SO}(3) \times [0, 1] \times \mathrm{S}^2$	image feature space, (13)
$C(y) = T_H c\|_{H=c^{-1}(y)}$	image-based Jacobian (18)
$\lambda_{\min}, \lambda_{\max} \in (0, 1)$	safe limits for λ (26)
$\theta \in (0, \pi/2)$	FOV occlusion angle (26)
$\varepsilon \in (0, \lambda_{\min})$	self occlusion safety margin (26)
$\mathcal{Z} = \mathcal{B} \times \mathrm{S}^1 \times [-1, 1] \times \mathcal{B}$	occlusion-free model space (27)
$z = (v, \alpha, \zeta, u) \in \mathcal{Z} = f(y)$	model space diffeomorphism (28)
$F(y) = T_y f$	Jacobian of f (31)

[1]All initial conditions but a set of measure zero converge. Note that for dynamical systems evolving on certain manifolds, such as SE(3), global convergence via smooth feedback is impossible [15].

is the Jacobian matrix of the mapping π at p. It will be useful later to define the unit tangent map $T^1\pi : T\mathbb{R}^3 \to T^1\mathrm{S}^2$, where $T^1\mathrm{S}^2$ denotes the unit tangent bundle of S^2, given by $T^1\pi : (p, a) \mapsto (\pi(p), v)$, where

$$v = \frac{T_p\pi \cdot a}{\|T_p\pi \cdot a\|}. \tag{4}$$

Note that this mapping has a singularity when a points along the ray defined by p and the camera origin.

Two distinct points, say $b, p \in \mathbb{R}^3$, might be modeled as a base point $b \in \mathbb{R}^3$ and a vector $a = p - b \in T\mathbb{R}^3$. One can show that $T^1\pi(b, a) = (\pi(b), v)$, where

$$v = \frac{T_b\pi \cdot a}{\|T_b\pi \cdot a\|} = \frac{T_b\pi \cdot (\pi(p) - \pi(b))}{\|T_b\pi \cdot (\pi(p) - \pi(b))\|} \tag{5}$$

and thus $v \in T^1_{\pi(b)}\mathrm{S}^2$ can be easily constructed from measured features.

B. Diffeomorphism to Image Space

Attach the body frame to the center of the sphere, so that the location of the body relative to the camera origin is given by the rigid translation d. If $\|d\| > \varrho$, i.e., the body of radius ϱ remains bounded away from the camera origin, then the surface of the body double covers a topological disc on S^2 via the map π. The edge of the disc, a planar slice of the image sphere, is a perfect circle of radius[2]

$$\lambda = \frac{\varrho}{\|d\|}, \quad \varrho < \|d\| < \infty.$$

(The circle radius λ appears dimensionless, because the image sphere was normalized to unit radius). The center of the circle on the image sphere is given by

$$s = \frac{d}{\|d\|}.$$

Let $\mathcal{R}_\varrho := \{d \in \mathbb{R}^3 : \|d\| > \varrho\}$ denote the body positions bounded away from the camera origin. We now have a diffeomorphism $c_1 : \mathcal{R}_\varrho \to (0, 1) \times \mathrm{S}^2$ from locations of the body to image measurements, given by

$$c_1 : d \mapsto (\lambda, s). \tag{6}$$

The inverse of c_1 is given by

$$d = c_1^{-1}(\lambda, s) = \frac{\varrho}{\lambda} s. \tag{7}$$

Attach a visible feature point b to the body's surface. For convenience, align the body frame so that its origin coincides with the center of the body, and assume that the point lies on the negative body frame z axis. Hence, in the body frame, $b^b = -\varrho e_3 = (0, 0, -\varrho)^T$.

As we will show (see Fig. 1), the projection of b to the image sphere, $q_1 = \pi(b)$, encodes two rotational DOFs. We encode the final DOF by projecting a unit vector or "arrow" a tangent to the body at the point b. In practice, any two distinguishable points on the surface of the sphere can be used to approximate the vector a (this is discussed further in Section V). Again, for convenience, we assume the vector's body-fixed representation is simply $a^b = e_2$. Let b^c denote the location of the body point

[2]To compute λ, draw a 2-D cross-section that contains both the camera and body origins, and note from "similar triangles" that $\lambda : 1 = \varrho : \|d\|$.

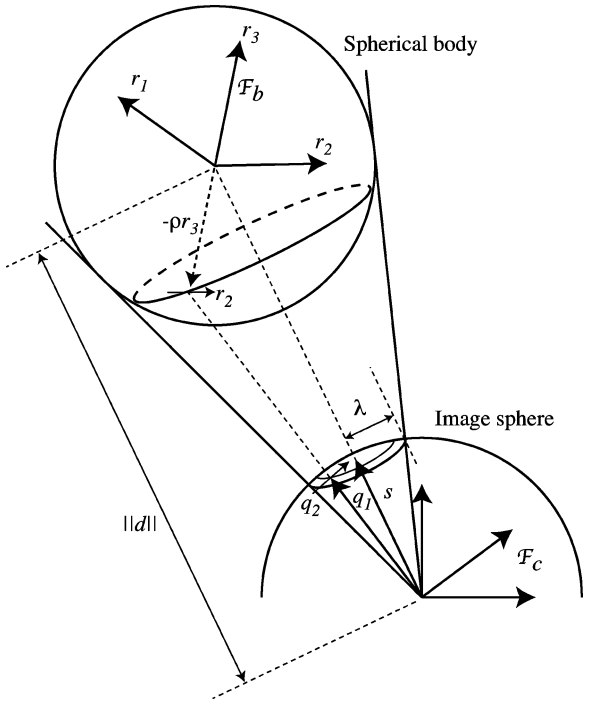


Fig. 1. Projection of a spherical body with a feature point on it to the image sphere. The image-plane measurement is given by $y = (Q, \lambda, s) = c(H)$.

b in the camera frame, and a^c represent the arrow vector in the camera frame. Recalling that the rotation matrix R has columns (r_1, r_2, r_3), then with respect to the camera frame, we have

$$b^c = Rb^b + d = d - \varrho r_3, \quad \text{and} \quad a^c = Ra^b = r_2.$$

Note that $b^b \cdot a^b = -\varrho e_3 \cdot e_2 = 0$.

Some configurations cause the body to occlude the feature point b; in particular, we can see the point b only if $b^c \cdot r_3 > 0$. So we define a *visibility function* [2], [16], ν, and an associated *visible set* of rigid transformations, $\mathcal{V}$ by

$$\begin{aligned} \nu(H) &:= (d - \varrho r_3) \cdot r_3 \\ \mathcal{V} &:= \{H \in \mathrm{SE}(3) : \nu(H) > 0\}. \end{aligned} \tag{8}$$

Note that $\nu(H) > 0 \Longrightarrow \|d\| > \varrho$, i.e., $d \in \mathcal{R}_\varrho = \{d \in \mathbb{R}^3 : \|d\| > \varrho\}$.

The projection of $(b, a) \in T\mathbb{R}^3$ to the image sphere is modeled by

$$T\pi : (b, a) \mapsto (\pi(b), T_b\pi \cdot a) \in T\mathrm{S}^2.$$

We are not concerned with the length of the projection of a, only the direction. Hence, consider the unit tangent map $T^1\pi$ in (4), given by

$$T^1\pi : (b, a) \mapsto (q_1, q_2) \tag{9}$$

where

$$q_1 = \frac{b^c}{\|b^c\|} = \frac{d - \varrho r_3}{\|d - \varrho r_3\|}, \quad q_2 = \frac{\Gamma_{q_1} a^c}{\|\Gamma_{q_1} a^c\|} = \frac{\Gamma_{q_1} r_2}{\|\Gamma_{q_1} r_2\|} \tag{10}$$

and $\Gamma_{q_1} := (I - q_1 q_1^T)$. Geometrically, q_2 is a unit vector tangent to the image sphere at the point q_1. The unit vectors q_1 and q_2 are mutually orthogonal. Consider the plane containing the camera origin o_c, the point b, and the vector a. The unit vector

$q_3 = q_1 \times q_2$ is normal to that plane. Thus, we define a function $c_2 : \mathcal{V} \to \mathrm{SO}(3)$

$$c_2 : H \mapsto [q_1 \quad q_2 \quad q_3] = Q. \tag{11}$$

Proposition I: The function $c : \mathcal{V} \to \mathcal{Y}$, defined by

$$c(H) := (c_2(H), c_1(d)) \tag{12}$$

where

$$\mathcal{Y} = \left\{ (Q, \lambda, s) \in \mathrm{SO}(3) \times (0,1) \times \mathrm{S}^2 : q_1 \cdot s > \sqrt{1-\lambda^2} \right\} \tag{13}$$

is a diffeomorphism, i.e., $\mathcal{V} \simeq \mathcal{Y}$.

The proof is given in Appendix I.

III. Image Jacobian

To be applicable to VS, we present a representation of the tangent map $Tc : T\mathcal{V} \to T\mathcal{Y}$, its inverse Tc^{-1}, and the cotangent map $T^*c : T^*\mathcal{Y} \to T^*\mathcal{V}$, with the following commutative diagram in mind:

$$\begin{array}{ccc} T\mathcal{V} & \underset{Tc^{-1}}{\overset{Tc}{\rightleftarrows}} & T\mathcal{Y} \\ \downarrow & & \downarrow \\ \mathcal{V} & \xrightarrow{\quad c \quad} & \mathcal{Y} \\ \uparrow & & \uparrow \\ T^*\mathcal{V} & \underset{T^*c}{\overset{T^*c^{-1}}{\rightleftarrows}} & T^*\mathcal{Y} \end{array}$$

We identify the tangent space $T\mathrm{SE}(3)$ of the Lie group SE(3) with

$$T\mathrm{SE}(3) \simeq \mathrm{SE}(3) \times \mathfrak{se}(3) \simeq \mathrm{SE}(3) \times (\mathbb{R}^3 \circledS \mathbb{R}^3) \tag{14}$$

where $\mathfrak{se}(3) \simeq \mathbb{R}^3 \circledS \mathbb{R}^3$ is the Lie algebra of SE(3) with Lie bracket

$$[(\mathbf{x}, \mathbf{y}), (\mathbf{a}, \mathbf{b})] = (\mathbf{x} \times \mathbf{a}, \mathbf{x} \times \mathbf{b} - \mathbf{a} \times \mathbf{y})$$

for all $(\mathbf{x}, \mathbf{y}), (\mathbf{a}, \mathbf{b}) \in \mathbb{R}^3 \times \mathbb{R}^3$. More detail can be found in [17], for example. The identification occurs via right translation, i.e.,

$$(H, \dot{H}) \mapsto (H, \dot{H}H^{-1}) \mapsto (H, (\boldsymbol{\omega}, \boldsymbol{v})) \tag{15}$$

where

$$H = \begin{bmatrix} R & d \\ 0^T & 1 \end{bmatrix}, \quad \dot{H} = \begin{bmatrix} \dot{R} & \dot{d} \\ 0^T & 0 \end{bmatrix}$$

and

$$\boldsymbol{\omega} = (\dot{R}R^{-1})^\vee, \quad \boldsymbol{v} = -\dot{R}R^{-1}d + \dot{d}$$

and the "wedge/hat" isomorphism $\mathbb{R}^3 \simeq \mathfrak{so}(3)$ is defined by

$$\widehat{\ }: \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 \end{bmatrix} \leftrightarrow \begin{bmatrix} 0 & -\boldsymbol{\omega}_3 & \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 & 0 & -\boldsymbol{\omega}_1 \\ -\boldsymbol{\omega}_2 & \boldsymbol{\omega}_1 & 0 \end{bmatrix} :^\vee$$

where $\mathfrak{so}(3)$ is the Lie algebra of SO(3).

Similarly, for each $y = (Q, \lambda, s) \in \mathcal{Y} \subset \mathrm{SO}(3) \times (0,1) \times \mathrm{S}^2$, we have the following identification:

$$\begin{aligned} T_y\mathcal{Y} &= T_Q\mathrm{SO}(3) \times T_\lambda(0,1) \times T_s\mathrm{S}^2 \\ &\simeq \mathbb{R}^3 \times \mathbb{R} \times T_s\mathrm{S}^2 \end{aligned} \tag{16}$$

where we identify $T_Q\mathrm{SO}(3)$ with $\mathfrak{so}(3) \simeq \mathbb{R}^3$, again via right translation

$$(Q, \dot{Q}) \mapsto (Q, \xi) \quad \text{where} \quad \xi = (\dot{Q}Q^{-1})^\vee. \tag{17}$$

Hence, to compute the tangent map Tc we find the mapping relating the tangent space identifications made above in (14) and (16), namely

$$\begin{bmatrix} \xi \\ \dot{\lambda} \\ \dot{s} \end{bmatrix} = C(y) \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}$$

where $y = (Q, \lambda, s) = c(H)$ and

$$C(y) = \begin{bmatrix} I_{3\times3} & \frac{1}{\beta}\left(\delta q_1 q_3^T - q_2 q_3^T + q_3 q_2^T\right) \\ 0_{1\times3} & -\frac{\lambda^2}{\varrho} s^T \\ -\hat{s} & \frac{\lambda}{\varrho}(I_{3\times3} - ss^T) \end{bmatrix} \tag{18}$$

where

$$\delta = \frac{s \cdot q_2}{\sqrt{\lambda^2 - \sin^2\phi}}, \quad \beta = \frac{\varrho}{\lambda}(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi}),$$
$$\cos\phi = s \cdot q_1 \quad \text{and} \quad \sin\phi = \sqrt{1 - (s \cdot q_1)^2}.$$

The construction of C is given in Appendix II.

Some explanation may help clarify why we do not have a 6×6 Jacobian matrix. Since SE(3) is a Lie group, its tangent bundle has a trivial cross-product structure. Thus, rigid velocity vectors are written in $\mathbb{R}^6$, resulting in six columns to C. However, we have embedded S^2 into $\mathbb{R}^3$ as follows:

$$\mathrm{S}^2 = \left\{ s \in \mathbb{R}^3 \middle| \, \|s\| = 1 \right\}.$$

With this embedding, we express S^2 (and thus, $T\mathrm{S}^2$) globally, but this leads to an extra row in C.

A set of local coordinates on S^2 yields a 6×6 Jacobian matrix, as follows. Let $\psi : U \subset \mathrm{S}^2 \subset \mathbb{R}^3 \to \psi(U) \subset \mathbb{R}^2$ be some local chart, for example, $(\theta_1, \theta_2) = \psi(s)$ might be the "azimuth and elevation." Then, in local coordinates

$$J(y) = \begin{bmatrix} I_{3\times3} & 0 & 0_{3\times3} \\ 0_{1\times3} & 1 & 0_{1\times3} \\ 0_{2\times3} & 0 & \frac{\partial\psi}{\partial s} \end{bmatrix} C(y)$$

where $C(y)$, given in (18), is a full-rank Jacobian matrix $J(y) \in \mathbb{R}^{6\times6}$, locally representing Tc. Using these coordinates, Tc^{-1} and Tc^* can be computed locally as J^{-1} and J^T, respectively. We take a similar approach in Section IV-D.

For a coordinate-free computation of the inverse and cotangent maps, see [18].

IV. Controller Design

A. Plant Model: Eye-in-Hand or Fixed Camera Servoing

We consider both *eye-in-hand* VS, wherein the camera moves relative to a stationary body, and *fixed-camera* VS, for which a stationary camera observes an actuated moving body. Let $G = H^{-1}$ denote the transformation of the camera frame $\mathcal{F}_c$ relative to the inertial body frame $\mathcal{F}_b$. Let (Ω, V) denote the body-frame velocity of the camera, moving relative to the inertial target frame, and note that

$$\begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0^T & 0 \end{bmatrix} = G^{-1}\dot{G} = -\dot{H}H^{-1} = -\begin{bmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0^T & 0 \end{bmatrix} \tag{19}$$

mapping the identification of $T\mathrm{SE}(3)$ given by the right translation of $\dot{H}$ in (14) and (15) to the left translation of $\dot{G} = (d/dt)(H^{-1})$. Note that this relationship clarifies the kinematic distinction between eye-in-hand and fixed-camera servoing.

For simplicity, we posit a fully actuated purely kinematic plant model

$$\dot{G} = G\begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0^T & 0 \end{bmatrix} \quad \text{or} \quad \dot{H} = \begin{bmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0^T & 0 \end{bmatrix} H \tag{20}$$

where we treat $(\boldsymbol{\omega}, \boldsymbol{v}) = -(\boldsymbol{\Omega}, \boldsymbol{V}) \in \mathbb{R}^3 \circledS \mathbb{R}^3$ as control inputs.

B. Servoing via Navigation Functions

The diffeomorphism c, the visible set $\mathcal{V}$, and its relatively simple image $\mathcal{Y}$ provide substantial leverage into the VS problem. Given a desired configuration H^*, measured through its image $y^* = (Q^*, \lambda^*, s^*) = c(H^*)$, there are many possible image-based control strategies we can employ to achieve our objective of driving $H \to H^*$.

One possible open-loop control strategy involves planning a path $y_d(t) \in \mathcal{Y}$ that moves from the initial configuration to the goal state and following the path via

$$\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = T_y c^{-1} \begin{bmatrix} \xi_d \\ \dot{\lambda}_d \\ \dot{s}_d \end{bmatrix} \tag{21}$$

where $(\xi_d, \dot{\lambda}_d, \dot{s}_d)$ is the desired velocity $\dot{y}_d$, expressed using the tangent space identification in (16). The minus sign in the above expression arises due to the relation in (19).

An open-loop strategy, such as the one above, may be undesirable. However, the generation of $\dot{y}_d$ can also be conceived as a feedback law, for example, by using the method of NFs [3], [15], [19]. A substantial benefit of using NFs is that they allow us to address second-order settings with little additional effort, while maintaining similar convergence guarantees [19]. Moreover, these methods have already proven practicable for dynamic VS [2]. The following definition gives a set of conditions that guarantee essentially global convergence of an NF-based controller.

Definition I [19]: Let $\mathcal{D}$ be a smooth compact connected manifold with boundary, and $x^* \in \overset{\circ}{\mathcal{D}}$ be a point in its interior. A Morse function $\varphi \in C^2[\mathcal{D}, [0,1]]$ is called an NF if:

1) φ takes its unique minimum at $\varphi(x^*) = 0$;
2) φ achieves its maximum of unity uniformly on the boundary, i.e., $\partial\mathcal{D} = \varphi^{-1}(1)$.

For any function satisfying the above definition, the controller given by $\dot{x} = -\nabla\varphi$ will ensure convergence $x \overset{t\to\infty}{\longrightarrow} x^*$ from essentially all initial conditions in $\mathcal{D}$ [19].

Before defining an NF, we first construct a compact domain $\mathcal{Y}_{\text{safe}} \subset \mathcal{Y}$, as well as its preimage $\mathcal{V}_{\text{safe}} = c^{-1}(\mathcal{Y}_{\text{safe}}) \subset \mathrm{SE}(3)$. Then, we introduce a geometrically simple model space $\mathcal{Z}$ that is diffeomorphic to $\mathcal{Y}_{\text{safe}}$, and construct the NF $\widetilde{\varphi} : \mathcal{Z} \to [0,1]$ on this model space, with the following diagram in mind:

$$\mathcal{V}_{\text{safe}} \xrightarrow{c} \mathcal{Y}_{\text{safe}} \xrightarrow{f} \mathcal{Z} \xrightarrow{\widetilde{\varphi}} \mathbb{R}, \qquad \varphi = \widetilde{\varphi} \circ f \circ c \tag{22}$$

where $f \circ g(x) = f(g(x))$ denotes function composition. Since the dynamical convergence properties of NFs are invariant under diffeomorphisms, we may use the changes of coordinates f and c to "pull the NF back" to the workspace. This can be done by either pulling back the gradient vector field through the inverse tangent map

$$\nabla\varphi = T(f \circ c)^{-1}\nabla\widetilde{\varphi} = (Tc^{-1})(Tf^{-1})\nabla\widetilde{\varphi} \tag{23}$$

or by pulling back the covector through the cotangent map

$$\mathbf{d}\varphi = T^*(f \circ c)\mathbf{d}\widetilde{\varphi} = (T^*c)(T^*f)\mathbf{d}\widetilde{\varphi}. \tag{24}$$

In local coordinates, (23) corresponds to using the "inverse Jacobian matrix" and (24) is the "transposed Jacobian matrix" of the function $f \circ c$.

Now, given $\varphi : \mathcal{V}_{\text{safe}} \to [0,1]$, with unique minimum at y^*, and some other technical properties discussed below, by letting

$$\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = -\nabla\varphi \tag{25}$$

the control law given by (25) drives $H \in \mathcal{V}_{\text{safe}}$ to H^*, ensuring that y converges to y^* for all initial conditions, except for a measure-zero set. Using an appropriate damping term and the covector $\mathbf{d}\varphi$, similar convergence results are readily obtained for second-order Lagrangian plant models [2], [3], [15], [19].

C. Computing a Safe Domain

The next step is to compute a compact domain $\mathcal{Y}_{\text{safe}} \subset \mathcal{Y}$ that is "safe" with respect to the FOV of our camera system in the sense that if $H \in c^{-1}(\mathcal{Y}_{\text{safe}})$, then all the necessary features are visible. To illustrate, we treat the FOV as a cone originating at the camera origin, with center along e_3, as shown in Fig. 2.

We assume that as long as the center of the sphere is within a (conservative) cone, that the entire body is fully visible.[3] This can be codified by the inequality constraint $s \cdot e_3 \geq \cos\theta$, where θ is the angle from e_3 to the edge of a conservative cone within the FOV. Additionally, we constrain $\lambda \in [\lambda_{\min}, \lambda_{\max}] \subset (0,1)$, where the parameters $\lambda_{\min}$ and $\lambda_{\max}$ bound the camera

[3]It is straightforward, albeit tedious, to find coordinates on SE(3) for the visible domain that results from a rectangular image plane [2]. Such a parameterization in image coordinates y represents work in progress, so we resort at present to a somewhat conservative subset of visible configurations.

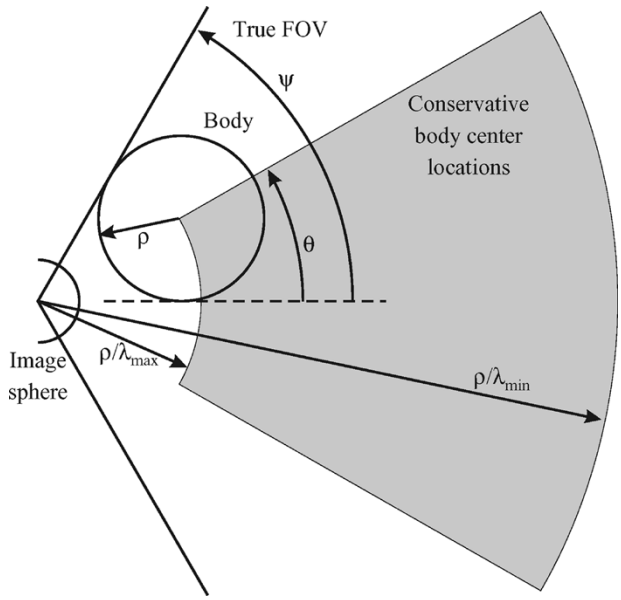


Fig. 2. 2-D cross section of a simplified FOV cone. ψ is the true FOV angle, θ conservatively limits the angle of s relative to the optical axis, $0 < \lambda_{\min} < \lambda_{\max} \leq \sin(\psi - \theta)$ prevent the body from being "too close to" or "too far from" the camera.

displacement from being "too far from" or "too close to" the camera, respectively. Finally, we keep q_1 from being too close to the edge of the projected circle, namely, $q_1 \cdot s \geq \sqrt{1-(\lambda-\varepsilon)^2}$, where $0 < \varepsilon < \lambda_{\min}$ is a small positive constant safety margin. Putting these constraints together yields the compact manifold

$$\mathcal{Y}_{\text{safe}} = \Big\{ y = (Q, \lambda, s) \in \text{SO}(3) \times [\lambda_{\min}, \lambda_{\max}] \times \text{S}^2 : s \cdot e_3 \geq \cos\theta, q_1 \cdot s \geq \sqrt{1-(\lambda-\varepsilon)^2} \Big\} \subset \mathcal{Y} \quad (26)$$

where $\theta \in (0, \pi/2), 0 < \varepsilon < \lambda_{\min}$, and $0 < \lambda_{\min} < \lambda_{\max} < 1$.

D. Model Space for Occlusion-Free Servoing

Because $\mathcal{Y}_{\text{safe}}$ is somewhat complicated geometrically, the next step involves designing a model space $\mathcal{Z}$ for $\mathcal{Y}_{\text{safe}}$, and a diffeomorphism $f : \mathcal{Y}_{\text{safe}} \to \mathcal{Z}$. Consider the model space

$$\mathcal{Z} = \mathcal{B} \times \text{S}^1 \times [-1, 1] \times \mathcal{B} \quad (27)$$

where $\mathcal{B} = \{u \in \mathbb{R}^2 : \|u\| \leq 1\}$ is a closed, planar unit disk.

We define the diffeomorphism $f : \mathcal{Y}_{\text{safe}} \to \mathcal{Z}$ as follows. Let $y = (Q, \lambda, s) \in \mathcal{Y}_{\text{safe}}$, where[4]

$$\begin{aligned} Q &= [q_1, q_2, q_3] \in \text{SO}(3) \\ q_i &= (q_{i1}, q_{i2}, q_{i3})^T, \quad i = 1, 2, 3 \\ \lambda &\in \mathbb{R} \quad \text{and} \quad s = (s_1, s_2, s_3)^T \in \text{S}^2. \end{aligned}$$

[4]The indexes for the elements of Q, e.g., $q_1 = (q_{11}, q_{12}, q_{13})^T$, are not the usual matrix indexing convention. It is helpful to think of q_{ji} as the ith element of the vector q_j.

We construct the map

$$f : (Q, \lambda, s) \mapsto (v, \alpha, \zeta, u) = z$$

as follows:

$$\begin{aligned} v &= \frac{(q_1 \cdot x_1, q_1 \cdot x_2)}{(\lambda - \varepsilon)} \\ e^{i\alpha} &= \frac{(q_{21} + i q_{22})}{\sqrt{(q_{21})^2 + (q_{22})^2}} \\ \zeta &= \frac{(2\lambda - \lambda_{\max} - \lambda_{\min})}{(\lambda_{\max} - \lambda_{\min})} \\ u &= \frac{(s_1, s_2)}{\sin\theta} \end{aligned} \quad (28)$$

where $u = (u_1, u_2)$, $v = (v_1, v_2)$, and

$$\{x_1, x_2, x_3\} = \left\{ \frac{s \times e_2}{\|s \times e_2\|}, \frac{s \times (s \times e_2)}{\|s \times e_2\|}, s \right\} \quad (29)$$

is an orthonormal basis for $\mathbb{R}^3$. Note that $\|u\|^2 \sin^2\theta = (s_1^2 + s_2^2) = 1 - s_3^2$, namely, $\|u\| \leq 1$ if and only if $s \cdot e_3 \geq \cos\theta$. Similarly, $\|v\|^2(\lambda-\varepsilon)^2 = (q_1 \cdot x_1)^2 + (q_1 \cdot x_2)^2$, and since $(q_1 \cdot x_1, q_1 \cdot x_2, q_1 \cdot x_3) \in \mathbb{R}^3$ is a unit vector, we have $\|v\|^2(\lambda-\varepsilon)^2 = 1 - (q_1 \cdot x_3)^2 = 1 - (q_1 \cdot s)^2$, namely, $\|v\| \leq 1$ if and only if $s \cdot q_1 \geq \sqrt{1-(\lambda-\varepsilon)^2}$. Finally, by construction, $\zeta \in [-1, 1]$ if and only if $\lambda \in [\lambda_{\min}, \lambda_{\max}]$. Summarizing, we have

$$\begin{aligned} \|u\| \leq 1 &\iff s \cdot e_3 \geq \cos\theta \\ \|v\| \leq 1 &\iff s \cdot q_1 \geq \sqrt{1-(\lambda-\varepsilon)^2} \\ \zeta \in [-1, 1] &\iff \lambda \in [\lambda_{\min}, \lambda_{\max}] \end{aligned}$$

i.e., $y \in \mathcal{Y}_{\text{safe}}$ if and only if $z \in \mathcal{Z}$.

Let $X = [x_1, x_2, x_3]$ be the orthogonal matrix whose columns are the basis vectors x_i, $i = 1, 2, 3$. Then the inverse, f^{-1}, is given by

$$\begin{aligned} s &= \left(u_1 \sin\theta, u_2 \sin\theta, \sqrt{1 - \|u\|^2 \sin^2\theta} \right)^T \\ \lambda &= \frac{((\lambda_{\max} - \lambda_{\min})\zeta + \lambda_{\max} + \lambda_{\min})}{2} \\ q_1 &= X \left(v_1(\lambda-\varepsilon), v_2(\lambda-\varepsilon), \sqrt{1 - \|v\|^2(\lambda-\varepsilon)^2} \right)^T \\ \tilde{q}_2 &= (\tilde{q}_{21}, \tilde{q}_{22}, \tilde{q}_{23})^T \\ &= \left(\cos\alpha, \sin\alpha, -\frac{(q_{11}\tilde{q}_{21} + q_{12}\tilde{q}_{22})}{q_{13}} \right)^T \\ q_2 &= \frac{\tilde{q}_2}{\|\tilde{q}_2\|} \\ q_3 &= q_1 \times q_2. \end{aligned} \quad (30)$$

The auxiliary vector $\tilde{q}_2$ is introduced for readability, to divide the construction of q_2 into two steps.

We compute Tf by using the tangent-space identification for $T\mathcal{Y}_{\text{safe}} \subset T\mathcal{Y}$ given by (16), and $T_z\mathcal{Z}$ is identified with $\mathbb{R}^6$ in the obvious way, namely, let $z = (v_1, v_2, \alpha, \zeta, u_1, u_1) \in \mathcal{Z}$ so

that $\dot{z} = (\dot{v}_1, \dot{v}_2, \dot{\alpha}, \dot{\zeta}, \dot{u}_1, \dot{u}_1) \in T_z\mathcal{Z} \simeq \mathbb{R}^6$. Let $F(y) = T_{yf}$ denote the Jacobian. Then, by direct computation

$$F(y) = \begin{bmatrix} \frac{(q_1 \times x_1)^T}{\lambda - \varepsilon} & -\frac{q_1 \cdot x_1}{(\lambda-\varepsilon)^2} & \\ \frac{(q_1 \times x_2)^T}{\lambda - \varepsilon} & -\frac{q_1 \cdot x_2}{(\lambda-\varepsilon)^2} & \\ \frac{(q_2 \times (e_3 \times q_2))^T}{(q_{21})^2 + (q_{22})^2} & 0 & \cdots \\ 0^T & \frac{2}{\lambda_{\max} - \lambda_{\min}} & \\ 0^T & 0 & \\ 0^T & 0 & \end{bmatrix}$$
$$\begin{bmatrix} \frac{(e_2 \times q_1 + q_1 \cdot x_1 (x_1 \times e_2))^T}{(\lambda - \varepsilon)\varsigma} \\ \left(\frac{x_1 \times q_1}{\lambda - \varepsilon} + \frac{e_2 \times (q_1 \times s) + q_1 \cdot x_2 (x_1 \times e_2)}{(\lambda - \varepsilon)\varsigma} \right)^T \\ 0^T \\ 0^T \\ \csc\theta e_1^T \\ \csc\theta e_2^T \end{bmatrix} \quad (31)$$

where the x_i's are given in (29) and $\varsigma = \|s \times e_2\|$.

Remark: Note that $F(y)$ is a 6×7 matrix. As explained in Section III, this arises from the representation of $T_s\mathrm{S}^2$. It is worth noting that

$$J(y) = F(y)C(y) \in \mathbb{R}^{6\times 6} \quad (32)$$

full-rank matrix everywhere on $\mathcal{Z}$, since f and c are both diffeomorphisms.

E. Navigation Function Construction

There are no mutual constraints between the variables u, v, ζ, and α. In other words, the model space $\mathcal{Z}$ is the simple Cartesian product of four simpler spaces: $[-1, 1]$, S^1, and two copies of $\mathcal{B}$. Thus, we construct an NF on each space separately, and then compose the functions using the NF product defined in [20] to generate an NF on $\mathcal{Z}$.

The basic construction of NFs followed below follows Rimon and Koditschek [15]. For a given compact manifold with boundary $\mathcal{X}$, we define two basic building blocks: a *goal function*, $\gamma : \mathcal{X} \to [0, \infty)$, possessing a unique minimum at a *goal point*, $x^* \in \mathcal{X}$, and an *obstacle function*, $\beta : \mathcal{X} \to [0, \infty)$ that vanishes (only) on the boundary $\partial\mathcal{X}$, which is treated as an obstacle set. These two building blocks are assembled to create a function $\varphi : \mathcal{X} \to [0, 1]$ as follows:

$$\varphi := \frac{\gamma}{\gamma + \beta}. \quad (33)$$

This construction ensures that φ is uniformly maximal on $\partial\mathcal{X}$. Each NF candidate must be examined on a case-by-case basis.

1) NF on $\mathcal{B}$: The manifold $\mathcal{B} := \{u \in \mathbb{R}^2 : \|u\| \leq 1\}$ is the simplest form of a "sphere world," as defined by Koditschek and Rimon [15]. We define goal and obstacle functions, respectively, as

$$\gamma_1(u) := \|u - u^*\|^2, \quad \beta_1(u) := 1 - \|u\|^2 \quad (34)$$

where $u^* \in \overset{\circ}{\mathcal{B}}$ (the interior of $\mathcal{B}$). The function $\varphi_1 = \gamma_1/(\gamma_1 + \beta_1)$, as defined in (33), is an NF [15].

2) NF on S^1: It is easy to show that

$$\varphi_2(\alpha) = \frac{1}{2}(1 - \cos(\alpha - \alpha^*)) \quad (35)$$

is an NF on S^1.

3) NF on $[-1, 1]$: Let $\zeta \in [-1, 1]$. Define goal and obstacle functions, respectively, as

$$\gamma_3(\zeta) := (\zeta - \zeta^*)^2, \quad \beta_3(\zeta) := 1 - \zeta^2 \quad (36)$$

where $\zeta^* \in (-1, 1)$ is the goal point. One can easily show that $\varphi_3 = \gamma_3/(\gamma_3 + \beta_3)$, as in (33), is an NF.

4) Navigation Function Composition: Let φ_1, φ_2, and φ_3 be the NFs defined above, on the respective manifolds $\mathcal{B}$, S^1, and $[-1, 1]$, and let σ be the diffeomorphism $\sigma(\chi) := \chi/(1 + \chi)$. Then, according to [20], the function

$$\widetilde{\varphi}(z) := \sigma\left(\kappa_1 \frac{\varphi_1(v)}{1 - \varphi_1(v)} + \kappa_2 \frac{\varphi_2(\alpha)}{1 - \varphi_2(\alpha)} + \kappa_3 \frac{\varphi_3(\zeta)}{1 - \varphi_3(\zeta)} + \kappa_4 \frac{\varphi_1(u)}{1 - \varphi_1(u)} \right) \quad (37)$$

is an NF on the cross-product space $\mathcal{Z} = \mathcal{B} \times \mathrm{S}^1 \times [-1, 1] \times \mathcal{B}$. The constants $\kappa_i > 0$, $i = 1, 2, 3, 4$ are free design gains, and can be used to tune system performance.

V. 6-DOF Simulation Experiments

We simulated a VS system in which a camera views a rigid collection of isolated feature points, and the image-plane diffeomorphism c of Section II serves as an *approximate* diffeomorphism. In other words, we interpret the projection of a set of feature points as if it were a "sphere with arrow."

Let $P = [p_1, \ldots, p_N] \in \mathbb{R}^{3\times N}$ denote a set of features expressed in the body frame as p_i^b, and in the camera frame as $p_i^c = Rp_i^b + d$, $i = 1, \ldots, N$. To approximate $\lambda \in (0, 1)$, we take half the distance between the two most mutually distant projected points, namely

$$\breve{\lambda} = \max_{i,j \in 1,\ldots,N} \frac{\left\| \pi\left(p_i^c\right) - \pi\left(p_j^c\right) \right\|}{2} \quad (38)$$

where π is given in (1). To approximate s, we take the mean projected unit vector

$$\breve{s} = \frac{\left(\frac{1}{N} \sum_{i=1}^{N} \pi\left(p_i^c\right) \right)}{\left\| \frac{1}{N} \sum_{i=1}^{N} \pi\left(p_i^c\right) \right\|}. \quad (39)$$

To approximate the matrix Q, we approximate (9) with two features, say p_1, p_2, together treated as the feature with arrow. Specifically, we let

$$(\breve{q}_1, \breve{q}_2) = T^1 \pi(p_1, p_2 - p_1) \quad (40)$$

and therefore

$$\begin{aligned} \breve{q}_1 &= \frac{p_1^c}{\|p_1^c\|} \\ \breve{q}_2 &= \frac{\Gamma_{\breve{q}_1}\left(p_2^c - p_1^c\right)}{\left\|\Gamma_{\breve{q}_1}\left(p_2^c - p_1^c\right)\right\|} \\ \breve{q}_3 &= \breve{q}_1 \times \breve{q}_2 \end{aligned} \quad (41)$$

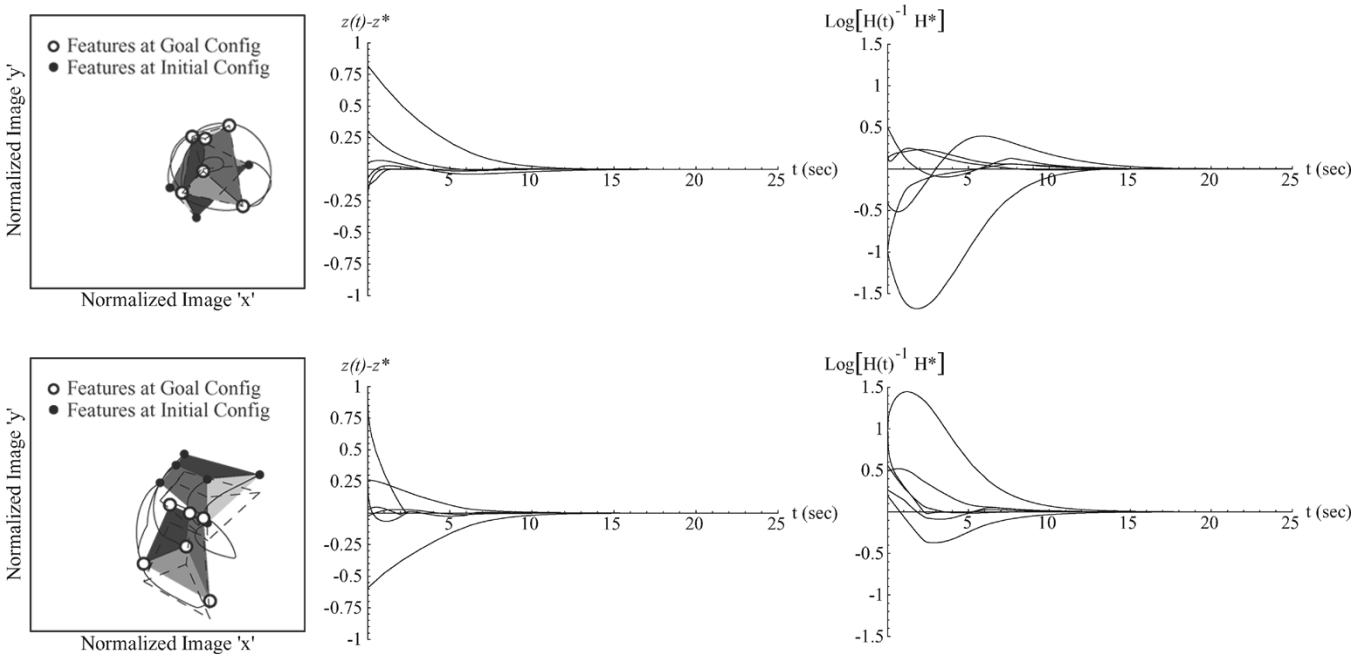


Fig. 3. Two typical experimental trials (Section V). The initial and desired views of the body are shown, together with the image-plane feature trajectories (left). In these trials, the initial and goal configurations differ by 135°–180° of rotation around the camera optical axis. The normalized feature errors in $\breve{\mathcal{Z}}$ (center) and the Cartesian error (right) are shown.

which generates the matrix $\breve{Q} = [\breve{q}_1, \breve{q}_2, \breve{q}_3] \in \mathrm{SO}(3)$. The complete image measurement is given by $\breve{y} = (\breve{Q}, \breve{\lambda}, \breve{s})$, and the image-based control law (25) is approximated with

$$\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = -\left(F(\breve{y})C(\breve{y})\right)^{-1} \nabla \widetilde{\varphi}(z)\Big|_{z=f(\breve{y})}$$
$$\dot{R} = \widehat{\boldsymbol{\omega}} R, \quad \dot{d} = \boldsymbol{\omega} \times d + \boldsymbol{v}. \tag{42}$$

Using Mathematica, we generated 250 trials with random visible initial and goal conditions, making a deliberate effort to test initial conditions near visibility boundaries. Also, each simulation used a set of six features, structured as follows:

$$P^b = \begin{bmatrix} 0 & 0 & -\delta_x & -\delta_x & \delta_x & \delta_x \\ -\frac{1}{2} & \frac{1}{2} & -1 & 1 & 1 & -1 \\ -\delta_z & -\delta_z & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 6}. \tag{43}$$

This structure of P was chosen to enable a systemic study of body geometries. We varied the feature parameters δ_x between 0.2 and 1 to test different body aspect ratios, and tested δ_z between 0 (corresponding to a planar object) and 1. Most choices of δ_x and δ_z were random, but we also hand-selected "bad" choices, such as $\delta_x = 0.2$ and $\delta_z = 0$, so that the object was not at all spherical. We approximated ϱ as the smallest radius sphere containing the points (which changes for each matrix P, as a function of δ_x and δ_z). Note that for our control law, we only need to know ϱ, not P. We set the gains $\kappa_i = 1, i = 1,2,3,4$ in the NF (37). We stopped each after 25 (simulated) seconds.[5] Fig. 3 depicts several simulations.

All simulations converged, even for planar objects. We also observed anecdotally that varying our estimate of ϱ used in (42) by up to a factor of two did not impair convergence, though we leave more careful experimental and formal parametric robustness analyses as future work.

[5]Note that convergence rates can be "sped up" or "slowed down" by tuning the gains κ_i, and by scaling the overall gradient in (42)

VI. Conclusion

We presented a global diffeomorphism from a large subset of configurations in SE(3), those that are *visible*, to an appropriately defined image space. Such constructions shed light on the geometry of occlusion-free servoing as well as provide a clear pathway to construct global, dynamical VS systems by using, for example, NFs, as done in this paper. Sensor-based representations of the configuration space also enables the control of underactuated and kinematically nonholonomic systems in sensor space [21], [22].

There are three important directions for future work: 1) designing diffeomorphisms on more natural feature constellations, for example, by formalizing the approximation made in Section V; 2) demonstrating, at least empirically, the robustness of this approach to camera intrinsic parameters; and 3) increasing the safe domain to be less conservative.

Appendix I
Proof of *Proposition 1*

(Proof that $c : \mathcal{V} \to \mathcal{I}$ is a diffeomorphism.) The proof proceeds in four parts. First, we show that c is smooth on $\mathcal{V}$. Next, we show that $c(\mathcal{V}) \subset \mathcal{I}$. Third, we show that c is bijective by explicitly computing its inverse c^{-1} on $\mathcal{I}$. Finally, we show that c^{-1} is smooth on $\mathcal{I}$.

A. The Function c is Smooth

The function c is composed of smooth functions away from the set where the arguments of $\|\cdot\|^{-1}$ become zero. But those arguments are nonzero on $\mathcal{V}$.

1) Equation (6) depends on $\|d\|^{-1}$. However, $H \in \mathcal{V}$ implies $\|d\| > \varrho$.

2) Equation (9) depends on $\|d-\varrho r_3\|^{-1}$. Visibility implies $\|d\| > \varrho$, which, in turn, implies $\|d - \varrho r_3\| \geq \|d\| - \|\varrho r_3\| = \|d\| - \varrho > 0$.
3) Equation (9) depends on $\|\Gamma_{q_1} r_2\|^{-1}$. This blows up if and only if $q_1 = \pm r_2$, i.e.,

$$q_1 = \frac{d - \varrho r_3}{\|d - \varrho r_3\|} = \pm r_2$$

and hence, from (8)

$$\nu(H) = (d - \varrho r_3) \cdot r_3 = \pm \|d - \varrho r_3\| r_2 \cdot r_3 = 0 \implies H \notin \mathcal{V}.$$

The contradiction implies $\|\Gamma_{q_1} R a^b\| > 0$ for $H \in \mathcal{V}$. Hence, c is smooth on $\mathcal{V}$.

B. The Image of $\mathcal{V}$ is Contained in $\mathcal{I}$

To see that $c(\mathcal{V}) \subset \mathcal{I}$, let $(Q, \lambda, s) = c(H)$. By construction of c, we have that $(Q, \lambda, s) \in \mathrm{SO}(3) \times (0, 1) \times \mathrm{S}^2$. To show that $q_1 \cdot s > \sqrt{1 - \lambda^2}$, note from (9) that

$$\begin{aligned} q_1 \cdot s &= \frac{(d - \varrho r_3) \cdot s}{\|d - \varrho r_3\|} \\ &= \frac{\|d\| - \varrho s \cdot r_3}{\sqrt{\varrho^2 - 2d \cdot r_3 + \|d\|^2}} \\ &= \frac{1 - \lambda\alpha}{\sqrt{1 - 2\lambda\alpha + \lambda^2}} \end{aligned}$$

where $\alpha = s \cdot r_3$. From (8), $\alpha > \lambda$ to ensure $\nu(H) > 0$. Since the right-hand side reaches its minimum of $\sqrt{1 - \lambda^2}$ for $\alpha = \lambda$, we have that $q_1 \cdot s > \sqrt{1 - \lambda^2}$.

C. The Function c is Bijective

Consider any $H \in \mathcal{V}$, with rotation $R = [r_1\ r_2\ r_3]$ and translation d as usual. Let $c(H) = (Q, \lambda, s) \in \mathcal{J}$, where $Q = [q_1\ q_2\ q_3]$. Given (λ, s), recovering the translation from (7) is trivial, namely, $d = c_1^{-1}(\lambda, s) \in \mathcal{R}_\varrho$. Recovering the rotation from (Q, λ, s) requires a bit more care.

Consider the triangle defined by the camera and body origins, o_c and o_b, and the point on the surface of the sphere, b. Relative to the camera frame, the points $o_c^c = 0$ and $o_b^c = d$ are known, but b^c is unknown. However, since $q_1 = b^c / \|b^c\|$ is the projection of b to the image sphere, then

$$b^c = \beta q_1, \text{ for some (unknown) } \beta \in \mathbb{R}.$$

Moreover, b lies on the surface of the spherical body of radius ϱ centered at o_b. Let ϕ denote the known angle between $\overline{bo_c}$ and $\overline{o_b o_c}$, given in the camera frame by

$$\cos\phi = q_1 \cdot s > \sqrt{1 - \lambda^2} > 0 \implies \phi \in \left[0, \frac{\pi}{2}\right).$$

From the above, we know the lengths of two sides of the triangle o_c, o_b, b, namely $\|d\| = \varrho/\lambda$ and ϱ, one angle, ϕ. From the law of cosines

$$\beta^2 + \|d\|^2 - 2\beta\|d\|\cos\phi = \varrho^2 \implies \beta = \frac{\varrho}{\lambda}(\cos\phi + \sigma\sqrt{\lambda^2 - \sin^2\phi}), \quad \text{where } \sigma = \pm 1.$$

Since $\cos\phi > \sqrt{1 - \lambda^2}$, then $\sin^2\phi < \lambda^2$, and thus there are two distinct, real solutions for β. Note that $r_3 = (d - b)/\varrho$, and thus

$$\begin{aligned} \nu(H) &= b \cdot \frac{(d - b)}{\varrho} \\ &= \frac{\left(\beta\|d\|\cos\phi - \beta^2\right)}{\varrho} \\ &= \left(\frac{\beta}{\lambda}\right)\left(\cos\phi - \frac{\lambda\beta}{\varrho}\right) \\ &= \left(\frac{\beta}{\lambda}\right)\left(\cos\phi - (\cos\phi + \sigma\sqrt{\lambda^2 - \sin^2\phi})\right) \\ &= -\sigma\frac{\beta}{\lambda}\sqrt{\lambda^2 - \sin^2\phi}. \end{aligned}$$

It is easy to show that $\beta > 0$ for either choice of σ, hence, visibility implies $\sigma = -1$, allowing us to uniquely compute $b = \|b\| q_1$, where

$$\|b\| = \frac{\varrho}{\lambda}(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi}). \tag{44}$$

Thus, $r_3 = (d - b)/\varrho$, where d and b are in terms of (Q, λ, s).

From (9), $r_2 \in \mathrm{span}\{q_1, q_2\}$, namely

$$r_2 = \alpha_1 q_1 + \alpha_2 q_2 \tag{45}$$

for some α_1 and α_2. Note from (9) that $\alpha_2 > 0$. Moreover, $r_3 \cdot r_2 = 0$, hence

$$\overbrace{\begin{bmatrix} r_3^T & 0 & 0 \\ I & -q_1 & -q_2 \end{bmatrix}}^{M} \begin{bmatrix} r_2 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = 0, \quad \|r_2\| = 1, \quad \alpha_2 > 0.$$

To show that $M \in \mathbb{R}^{4\times 5}$ has a full rank, we show that M has no left null space. Suppose $[a_1, a_2^T] \in \mathbb{R}\times\mathbb{R}^3$ satisfies $[a_1, a_2^T]M = 0$. Then we have $a_2 \cdot q_1 = a_2 \cdot q_2 = 0$. So, $a_2 = kq_3$ for some number k. Hence, we have $a_1 r_3 + kq_3 = 0$. Note that r_3 and q_3 are not collinear, because $(r_3 \times q_3) \cdot q_2 = (r_3 \times (q_1 \times q_2)) \cdot q_2 = ((r_3 \cdot q_2)q_1 - (r_3 \cdot q_1)q_2) \cdot q_2 = r_3 \cdot q_1 = \beta\nu(H) > 0$, and therefore, $a_1 = k = 0$. Hence, $[a_1, a_2] = 0$, so M has a full rank. Since M is full rank, it has a 1-D right null space. Hence, there are two possible solutions to $Mv = 0$ for r_2, α_1, α_2 subject to $\|r_2\| = 1$; the ambiguity is eliminated, since $\alpha_2 > 0$.

Combining the above computations yields the unique inverse to c

$$\begin{aligned} c^{-1} : (Q, \lambda, s) \mapsto H &= \begin{bmatrix} r_1 & r_2 & r_3 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where} \\ r_3 &= \frac{1}{\lambda}\left(s - (\cos\phi - \sqrt{\lambda^2 - \sin^2\phi})q_1\right) \\ r_2 &= \frac{\left(q_2 - \frac{r_3 \cdot q_2}{r_3 \cdot q_1} q_1\right)}{\left\|q_2 - \frac{r_3 \cdot q_2}{r_3 \cdot q_1} q_1\right\|} \\ r_1 &= r_2 \times r_3 \quad \text{and} \quad d = \frac{\varrho}{\lambda} s. \end{aligned} \tag{46}$$

D. The Function c^{-1} is Smooth

Finally, we need to show that c^{-1} is smooth. However, c^{-1} is composed of smooth functions. There are two caveats.

1) Equations involving $1/\lambda$. This is fine, since $0 < \lambda < 1$.
2) Equation for r_2. First, note that $r_3 \cdot q_1 = r_3 \cdot b/\|b\| = \nu(H)/\|b\| > 0$. Also, since q_2 and q_1 are linearly independent, the denominator can never be zero, so this equation is smooth on $\mathcal{I}$.

Hence, c^{-1} is smooth, and c is a diffeomorphism $\mathcal{V} \simeq \mathcal{I}$. ∎

Appendix II
Image Jacobian

To compute the image Jacobian matrix given by (18), we compute $\xi = (\dot{Q}Q^{-1})^{\vee}$, $\dot{\lambda}$, and $\dot{s}$, in terms of (Q, λ, s) and $(\boldsymbol{\omega}, \boldsymbol{v})$.

A. Computation of $\dot{\lambda}$

Recall

$$\dot{d} = \boldsymbol{\omega} \times d + \boldsymbol{v} \tag{47}$$

where $\boldsymbol{\omega} = (\dot{R}R^{-1})^{\vee}$. Since $\lambda = \varrho/\|d\|$, we have

$$\dot{\lambda} = -\frac{\varrho}{\|d\|^3} d \cdot \dot{d} = -\frac{\lambda^2}{\varrho} s \cdot \boldsymbol{v}. \tag{48}$$

B. Computation of $\dot{s}$

Recall $s = d/\|d\|$. Equation (47) implies

$$\dot{s} = -s \times \boldsymbol{\omega} + \frac{\lambda}{\varrho}(I - ss^T)\boldsymbol{v}. \tag{49}$$

C. Computation of $\xi = (\dot{Q}Q^{-1})^{\vee}$

For simplicity, we will first compute $Q^{-1}\dot{Q}$, and then use the property

$$Sx = (S\hat{x}S^{-1})^{\vee} \quad \forall S \in \mathrm{SO}(3), \quad x \in \mathbb{R}^3$$

to compute

$$\xi = Q(Q^{-1}\dot{Q})^{\vee}. \tag{50}$$

Proceeding in this manner, we have

$$Q^{-1}\dot{Q} = \begin{bmatrix} 0 & q_1 \cdot \dot{q}_2 & q_1 \cdot \dot{q}_3 \\ q_2 \cdot \dot{q}_1 & 0 & q_2 \cdot \dot{q}_3 \\ q_3 \cdot \dot{q}_1 & q_3 \cdot \dot{q}_2 & 0 \end{bmatrix}$$

which implies

$$(Q^{-1}\dot{Q})^{\vee} = (-q_2 \cdot \dot{q}_3, -q_3 \cdot \dot{q}_1, q_2 \cdot \dot{q}_1)^T. \tag{51}$$

It follows that we need to compute the three quantities $q_2 \cdot \dot{q}_1$, $q_3 \cdot \dot{q}_1$, $q_2 \cdot \dot{q}_3$ in terms of $\boldsymbol{\omega}$ and $\boldsymbol{v}$ in order to get an expression of ξ in terms of $\boldsymbol{\omega}$ and v. We remark that we chose $q_2 \cdot \dot{q}_1$, $q_3 \cdot \dot{q}_1$, $q_2 \cdot \dot{q}_3$ rather than the other possible three quantities, because the involved computation is relatively simple.

Let us first compute $\dot{q}_1$. Recall

$$q_1 = \frac{(d - \varrho r_3)}{\|b\|}$$

where $b = d - \varrho r_3$. From (47) and $\dot{r}_3 = \boldsymbol{\omega} \times r_3$, it follows:

$$\begin{aligned} \dot{q}_1 &= \frac{1}{\|b\|}(\dot{d} - \varrho \dot{r}_3) + (d - \varrho r_3)\frac{d}{dt}\left(\|b\|^{-1}\right) \\ &= \boldsymbol{\omega} \times q_1 + \frac{1}{\|b\|}\boldsymbol{v} + Aq_1 \end{aligned} \tag{52}$$

where $A = \|b\|(d/dt)(\|b\|^{-1})$. From (52), it follows:

$$q_2 \cdot \dot{q}_1 = q_3 \cdot \boldsymbol{\omega} + \frac{1}{\|b\|} q_2 \cdot \boldsymbol{v} \tag{53}$$

$$q_3 \cdot \dot{q}_1 = -q_2 \cdot \boldsymbol{\omega} + \frac{1}{\|b\|} q_3 \cdot \boldsymbol{v}. \tag{54}$$

We now compute $q_2 \cdot \dot{q}_3$. Recall

$$r_2 = \frac{q_2 - \delta q_1}{\|q_2 - \delta q_1\|}, \quad \delta := \frac{r_3 \cdot q_2}{r_3 \cdot q_1}. \tag{55}$$

This implies

$$\begin{aligned} \|q_1 \times r_2\| &= \frac{1}{\|q_2 - \delta q_1\|} \\ &= \frac{1}{\sqrt{1 + \delta^2}} \\ q_1 \cdot r_2 &= -\frac{\delta}{\sqrt{1 + \delta^2}} \end{aligned} \tag{56}$$

which will be used later in computations. From (55), we have $q_3 = q_1 \times q_2 = (q_1 \times r_2)/\|q_1 \times r_2\|$. It follows:

$$\dot{q}_3 = \frac{1}{\|q_1 \times r_2\|}(\dot{q}_1 \times r_2 + q_1 \times \dot{r}_2) + Bq_3 \tag{57}$$

where $B = \|q_1 \times r_2\|(d/dt)(\|q_1 \times r_2\|^{-1})$. Hence

$$\begin{aligned} q_2 \cdot \dot{q}_3 &= (1 + \delta^2)^{\frac{1}{2}} \left(r_2 \cdot (q_2 \times \dot{q}_1) - \dot{r}_2 \cdot q_3\right) \\ &= (1 + \delta^2)^{\frac{1}{2}} \left(r_2 \cdot (q_2 \times \dot{q}_1) + r_2 \cdot \dot{q}_3\right) \end{aligned} \tag{58}$$

where in the second equality, we used the fact that $r_2 \cdot q_3 = 0$ from (55) implies $r_2 \cdot \dot{q}_3 = -\dot{r}_2 \cdot q_3$.

Using (52), we have

$$\begin{aligned} r_2 \cdot (q_2 \times \dot{q}_1) &= -(q_1 \cdot r_2)(q_2 \cdot \boldsymbol{\omega}) + \frac{1}{\|b\|}(r_2 \times q_2) \cdot \boldsymbol{v} \\ &= -(q_1 \cdot r_2)(q_2 \cdot \boldsymbol{\omega}) \\ &\quad - \frac{\delta}{\|b\|\sqrt{1 + \delta^2}}(q_3 \cdot \boldsymbol{v}) \end{aligned} \tag{59}$$

where we used $r_2 \times q_2 = -\delta(1 + \delta^2)^{-1/2} q_3$, which comes from (55) and (56).

From (57), $r_2 \cdot q_3 = 0$ and $\dot{r}_2 = \boldsymbol{\omega} \times r_2$, we get

$$\begin{aligned} r_2 \cdot \dot{q}_3 &= \frac{1}{\|q_1 \times r_2\|} q_1 \cdot ((\boldsymbol{\omega} \cdot r_2) r_2 - \boldsymbol{\omega}) \\ &= \frac{1}{\|q_1 \times r_2\|} q_1 \cdot (\|q_1 \times r_2\|(q_2 \cdot \boldsymbol{\omega} - \delta q_1 \cdot \boldsymbol{\omega}) r_2 - \boldsymbol{\omega}) \\ &= (q_1 \cdot r_2)(q_2 \cdot \boldsymbol{\omega}) - \frac{(q_1 \cdot \boldsymbol{\omega})}{\sqrt{1 + \delta^2}} \end{aligned} \tag{60}$$

where we used (55) in the second equality, and (56) in the third equality.

Plugging (59), (60), and (56) into (58), we get

$$q_2 \cdot \dot{q}_3 = -q_1 \cdot \boldsymbol{\omega} - \frac{\delta}{\|b\|} q_3 \cdot \boldsymbol{v} \tag{61}$$

where δ can be expressed in terms of $y = (Q, \lambda, s)$ as

$$\delta = \frac{s \cdot q_2}{\sqrt{\lambda^2 - \sin^2 \phi}}, \quad \sin \phi = \sqrt{1 - (s \cdot q_1)^2}. \tag{62}$$

We are now in a position to compute ξ in (50). From (50) and (51), together with (53), (54), and (61), we get

$$\xi = \boldsymbol{\omega} + \frac{1}{||b||} \left(\delta q_1 q_3^T - q_2 q_3^T + q_3 q_2^T \right) \boldsymbol{v} \tag{63}$$

where δ is given by (62) or (55), and $||b||$ can be expressed in terms of $y = (Q, \lambda, s)$, as in (44).

ACKNOWLEDGMENT

Thanks to J. Piazzi, V. Kallem, D. Burschka, R. Vidal, and R. Ghrist for reading and critiquing the manuscript. Special thanks to the reviewers whose many insightful comments and suggestions contributed significantly to this paper and its presentation.

REFERENCES

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

[2] 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.

[3] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.

[4] N. R. Gans, S. A. Hutchinson, and P. I. Corke, "Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control," *Int. J. Robot. Res.*, vol. 22, no. 10, pp. 955–981, 2003.

[5] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 507–515, Aug. 2001.

[6] H. Zhang and J. P. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 199–208, Apr. 2002.

[7] 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.

[8] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 6, pp. 610–622, Nov. 2000.

[9] 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.

[10] 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.

[11] C. J. Taylor and J. P. Ostrowski, "Robust vision-based pose control," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, San Francisco, CA, 2000, pp. 2734–2740.

[12] P. Zanne, G. Morel, and F. Plestan, "Robust 3D vision based control and planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 5, Apr. 2004, pp. 4423–4428.

[13] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 724–735, Aug. 2004.

[14] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, pp. 618–623.

[15] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, pp. 412–442, 1990.

[16] N. J. Cowan and D. E. Koditschek, "Planar image based visual servoing as a navigation problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Detroit, MI, 1999, pp. 611–617.

[17] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*. New York: Springer, 1999.

[18] N. J. Cowan and D. E. Chang, "Geometric visual servoing," Johns Hopkins Univ., Baltimore, MD, Tech. Rep., 2005.

[19] D. E. Koditschek, "The application of total energy as a Lyapunov function for mechanical control systems," in *Dynamics and Control of Multibody Systems (Brunswick, ME, 1988)*. Providence, RI: Amer. Math. Soc., 1989, pp. 131–157.

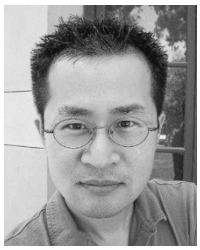
[20] N. J. Cowan, "Composing navigation functions on Cartesian products of manifolds with boundary," in *Proc. Workshop Algorithmic Found. Robot.*, Utrecht, The Netherlands, 2004, pp. 91–106.

[21] G. Kantor and A. Rizzi, "Feedback control of underactuated systems via sequential composition: Visually guided control of a unicycle," in *Proc. Int. Symp. Robot. Res.*, Siena, Italy, 2003, pp. 281–290.

[22] G. A. D. Lopes and D. E. Koditschek, "Level sets and stable manifold approximations for perceptually driven nonholonomically constrained navigation," in *Proc. IEEE Conf. Intell. Robot. Syst.*, 2004, pp. 1481–1486.

Noah J. Cowan (S'98–M'01) received the B.S. degree in electrical engineering from The Ohio State University, Columbus, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1997 and 2001.

He was a Postdoctoral Fellow with the Department of Integrative Biology, University of California at Berkeley, from 2001 to 2003. He is currently an Assistant Professor with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD. His research interests include sensor-based control, biologically inspired robot design, biomechanics, and neuroethology.



Dong Eui Chang received the B.S. degree in control and instrumentation engineering and the M.S. degree in electrical engineering, both from Seoul National University, Seoul, Korea, in 1994 and 1997, respectively, and the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, in 2002.

He was a Postdoctoral Fellow with the University of California, Santa Barbara, in 2003, with the Centre Automatique et Systèmes, Ecole Nationale Supérieure des Mines de Paris, Paris, France, in 2004, and with the University of Liege, Liege, Belgium, in 2005. He joined the Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada, in August 2005 as an Assistant Professor.

Dr. Chang received the Society for Industrial and Applied Mathematics (SIAM) Student Paper Award in 2002.