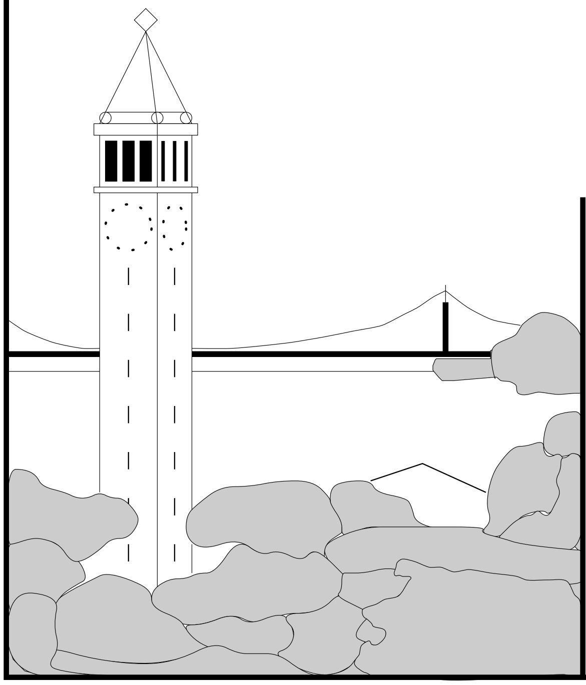
Toward Geometric Visual Servoing

Noah John Cowan[*] *and Dong Eui Chang*[†]
[*] *University of California, Berkeley, CA 94720, USA*
[†] *University of California, Santa Barbara, CA 93106, USA*



Report No. UCB/CSD-02-1200

September 26, 2002

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Toward Geometric Visual Servoing*

Noah John Cowan* and Dong Eui Chang†
*University of California, Berkeley, CA 94720, USA
†University of California, Santa Barbara, CA 93106, USA

September 26, 2002

Abstract

This paper presents a global diffeomorphism from a suitably defined "visible set" of rigid body configurations — a subset of SE(3) — to an image-space. The mapping between the visible set and the image-space is given by the projection of a set of features of a specially designed visual target. The target is a sphere marked with a feature point and a vector tangent to the sphere at the feature point. We show how the construction of a diffeomorphism to image-space should pave the way for developing global, dynamic visual servoing systems using Navigation Functions. This technical report is an addendum to the paper by the same authors [4].

*The first author was supported in part by DARPA/ONR under grants N00014-98-1-0747 and N66001-00-C8026, and NSF under grant ECS-9873474.

1 Introduction

There is a large and growing body of algorithms for "visual servoing" (VS) — motion control using visual feedback. Traditionally, VS algorithms generate motor reference velocities to register a camera's current view of a scene with a previously stored view (for a tutorial, see [7]).

We seek to move VS toward a systematic theory by characterizing the geometry of "visible" configurations of a visual target relative to a camera. In particular, for a specific target geometry we present a *diffeomorphism* — a smooth and smoothly invertible transformation — from an appropriately defined visible set of configurations to an image space. We believe this transformation will enable the construction of purely image-based, global dynamic VS algorithms.

1.1 Background

A significant challenge involves representing rigid motions in terms of visually measured quantities. Ideally, such a representation should enable effective encoding of

- **Configuration and State,** e.g. position and velocity or position and momentum for Lagrangian or Hamiltonian systems.
- **Tasks and goals,** e.g. trajectories in the state space or points in the configuration space.
- **Obstacles,** e.g. the edge of the field-of-view (FOV) for VS systems.
- **Uncertainty,** e.g. sensor and actuator noise or parametric error.

There are several candidate representations of image-based rigid motion to consider from the literature. The classical approach to "2D VS" employs the projection, treated as a vector in $\mathbb{R}^n$, of an arbitrary set of feature points [7]. The redundancy of using extra feature points seems to confer robustness to measurement noise in any one of the feature measurements. However, the movement of features is constrained by the underlying rigid motion, rendering image-based control and motion planning in image space challenging for large deviations from a goal. Notwithstanding those challenges, Corke and Hutchinson [2] created a 2D kinematic algorithm for 6DOF VS that seems (empirically) to have a very large basin of attraction while keeping features in the FOV. Their algorithm employs a clever choice of image features which helped motivate the choice of features used in this paper.

A more recent approach uses partial pose reconstruction: 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. Using this technique, researchers developed six DOF VS algorithms robust to calibration uncertainty [12, 15]. It is worth noting that the methods used require sufficient point correspondences between views to fully reconstruct a geometric model of the visual target [10]. Application of this method to contexts besides full six DOF VS remains a challenge.

Alternatively, one may recover the complete pose of a camera with respect to a target by exploiting a model of the target [11]. Vision-based controllers using full pose reconstruction are often referred to as "3D VS" algorithms. Model based pose reconstruction requires fewer feature points than the model-free approach described above, and has the added advantage of fully recovering feature depth, effectively reducing the camera to a "virtual Cartesian sensor." Representing visibility obstacles, such as the FOV or self-occlusions is less parsimonious, but can be done [5]. Formal results demonstrating parametric robustness of VS systems using this method remain elusive.

Generalized image-based coordinates have proven extremely effective in a few narrow contexts [3, 5, 16]. Generalized coordinates describe kinematic motion with one variable per mechanical DOF. Lagrange's equations, for example, are usually written using such coordinates. Hence, this approach enables the expression of *dynamical* equations of motion in terms of measured quantities on the image plane. Obstacles such the FOV and self-occlusions often appear as the boundary of a compact manifold in image-space and hence their avoidance may be cast as an instance of dynamical obstacle avoidance [5]. Although quite robust in practice, formal guarantees of robustness to noise or parametric uncertainty for this framework remains an open problem.

1.2 Contribution

To date, global image-based representations of configuration have been applied only to three DOF systems. This paper builds on previous results in a key way: we present an image-based, geometric representation of six DOF rigid motion. Our development of a global representation of "visible" rigid motions viewed through the projection of a set of features should help pave the way for new global, dynamic VS systems.

Organization. In Section 2, we employ a specific target geometry — a sphere with a few markings — to create a global image-based representation of motion for six DOF VS. Included in our development is a simple, purely image-based representation of the so-called image Jacobian (made possible since, as we show, the image and task spaces are diffeomorphic). In Section 4, we suggest a method for using our diffeomorphism for kinematic or dynamic control, although there is much open work to be done in this endeavor. Finally, we give some concluding remarks in Section 5.

2 Six DOF Diffeomorphism to Image-space

We assume a visual target may be designed to our specifications, so we may explore new image-based representations of rigid motion. In cases in which we have the freedom to design visual targets — for example when designing docking stations for space craft, helicopter landing beacons, or visual targets for a factory setting — this approach may lead to novel target designs that ease the control problem. More generally, it is hoped that the insight drawn from taking this approach may enable us to reinterpret target geometries over which we have have less design freedom.

Consider the problem of moving a rigid target object in six DOF relative to a perspective camera. The rigid target considered is as follows:

1. **A spherical body.** Consider 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 resolving two rotational DOF's from the location of the feature point on the image.

3. **A unit vector tangent to the body.** The final degree of freedom is resolved by considering the orientation on the image of a projected vector attached to our feature point on the body.

Zhang and Ostrowski [16, 17] developed the idea of projecting a spherical body to an image plane for VS of a blimp relative to a large ball. Using a "flat" image plane, the resulting image is an ellipse, which they approximate as a circle by assuming that a slice of the spherical body parallel to the image plane is projected. The present paper builds on that work, employing a more 'exact' diffeomorphism to the image-space, as well as incorporating additional markings on the body whose projection encodes rotational information.

2.1 Notation and Definitions

At the risk of burdening the reader with formalism, we present the following definitions to enable a precise geometric description of the domain and range of a camera viewing rigid motions.

An affine point $p \in \mathbb{A}^3$ has homogeneous coordinates $p = \begin{bmatrix} p_1 & p_2 & p_3 & 1 \end{bmatrix}^T$ with respect to some rigid frame. Note that $T\mathbb{A}^3 = \mathbb{A}^3 \times \mathbb{R}^3$, and that $\mathbb{R}^3$ acts on points to translate them in the usual way, so that if $\boldsymbol{v} =$

Table 1: List of symbols.

Symbol	Description
$o, p, b, \ldots \in \mathbb{E}^3$	Euclidean points (Roman)
$\boldsymbol{v}, \boldsymbol{e}, \ldots \in \mathbb{R}^3$	vectors (boldfaced)
$\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \in \mathbb{R}^3$	standard basis
$\pi : \{\mathbb{E}^3 - o_c\} \to \mathrm{S}^2$	image projection model – spherical panoramic camera
$\mathcal{F} = \{o, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$	rigid coordinate frame, $o \in \mathbb{E}^3$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \in \mathbb{R}^3$
$\mathcal{F}_c, \mathcal{F}_b$	camera frame and body frame
$p^b, \boldsymbol{v}^b$	point, p, and vector, $\boldsymbol{v}$, with respect to $\mathcal{F}_b$
$H \in \mathrm{SE}(3)$	rigid transformation of $\mathcal{F}_b$, relative to $\mathcal{F}_b$
$R \in \mathrm{SO}(3)$	rotation effected by H, columns $R = \begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3\end{bmatrix}$
$\boldsymbol{d} \in \mathbb{R}^3$	translation effected by H
$p^c = Hp^b, \boldsymbol{v}^c = R\boldsymbol{v}^b$	point, p, and vector, $\boldsymbol{v}$, with respect to $\mathcal{F}_c$
$\nu : \mathrm{SE}(3) \to \mathbb{R}$	measure of feature visibility, (5)
$\mathcal{V} \subset \mathrm{SE}(3)$	set of "visible" configurations, $H \in \mathcal{V} \iff \nu(H) > 0$
$\lambda \in (0,1)$	radius on image sphere of body, (3)
$\boldsymbol{s} \in \mathrm{S}^2$	unit vector pointing toward body centroid, (3)
$Q \in \mathrm{SO}(3)$	image-based rotation, columns $Q = \begin{bmatrix}\boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{q}_3\end{bmatrix}$, (7)
$(Q, \lambda, \boldsymbol{s}) = c(H)$	camera map, (8)
$\mathcal{I}$	image feature space, $\mathcal{I} \subset \mathrm{SO}(3) \times (0,1) \times \mathrm{S}^2$, (9)

$\begin{bmatrix}v_1 & v_2 & v_3\end{bmatrix}^T \in \mathbb{R}^3$ and $p \in \mathbb{A}^3$, then $p + \boldsymbol{v} = \begin{bmatrix}p_1 + v_1 & p_2 + v_2 & p_3 + v_3 & 1\end{bmatrix}^T \in \mathbb{A}^3$. Two points cannot be "added" together, but if $p, b \in \mathbb{A}^3$ then $\boldsymbol{v} = p - b \in \mathbb{R}^3$ is the vector such that $p = b + \boldsymbol{v}$. Adding the usual metric structure to affine space $\mathbb{A}^3$ yields Euclidean space $\mathbb{E}^3$ where the distance between two points is given by the two norm of their difference, $\|p - b\|$ (a measure independent of the choice of rigid frame).

A rigid frame, $\mathcal{F}$, is defined by its origin, $o \in \mathbb{E}^3$, and three mutually orthogonal unit vectors, $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \in \mathbb{R}^3$, that create a right-handed frame. Consider a full perspective ("pinhole") camera with frame $\mathcal{F}_c$ such that o_c is located at the pinhole (or optical center), with $\boldsymbol{k}_c$ aligned with the optical axis. The pinhole camera projects points in the open half space "in front" of the camera to an image-plane pair, given by via the map, $\pi^+\{\mathbb{E}^3 : (p - o_c) \cdot \boldsymbol{k}_c > 0\} \to \mathbb{R}^2$, expressed in camera frame coordinates

$$\pi^+(p) = \frac{f}{p_3}\begin{bmatrix}p_1 \\ p_2\end{bmatrix} \quad p_3 > 0\,, \tag{1}$$

where f is the camera focal length. The camera observes features of a rigid body, affixed with rigid frame $\mathcal{F}_b$. Let

$$H = \begin{bmatrix}R & \boldsymbol{d} \\ 0^T & 1\end{bmatrix} \in \mathrm{SE}(3), \quad \text{where} \quad R = \begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_1 & \boldsymbol{r}_3\end{bmatrix} \in \mathrm{SO}(3), \boldsymbol{d} \in \mathbb{R}^3,$$

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 $p^c = Hp^b$ with respect to the camera frame. Similarly, if $\boldsymbol{v}^b$ is a vector in the body frame, then $\boldsymbol{v}^c = R\boldsymbol{v}^b$ is the same vector with respect to the camera frame.

Hamel *et. al.* [6] remap the image plane to a sphere to recover some symmetry that is "broken" by a flat image plane. This approach has also been used in the structure from motion (SFM) literature [1]. Let $\boldsymbol{p} = (p - o_c)$ and note that the unit vector, $\boldsymbol{p}/\|\boldsymbol{p}\|$ may be recovered from the image-plane pair in (1) since

$$\frac{\boldsymbol{p}}{\|\boldsymbol{p}\|} = \begin{bmatrix}\pi^+(p) \\ f\end{bmatrix} \Big/ \left\| \begin{bmatrix}\pi^+(p) \\ f\end{bmatrix} \right\|, \quad p_3 > 0$$

with respect to the camera frame. Of course, this assumes that we know the parameter f (or, more generally, all so-called "intrinsic" camera parameters, omitted to simplify the presentation). Motivated by this observation, we consider for convenience a "panoramic" spherical camera

$$\begin{aligned} \pi &: (\mathbb{E}^3 - \{o_c\}) \to \mathrm{S}^2 \\ &: p \mapsto \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|} \quad \text{where} \quad \boldsymbol{p} = (p - o_c)\,. \end{aligned} \tag{2}$$

For the purposes of this paper, $\mathrm{S}^2 = \{\boldsymbol{v} \in \mathbb{R}^3 : \boldsymbol{v} \cdot \boldsymbol{v} = 1\} \subset \mathbb{R}^3$. For the camera map, S^2 corresponds to the unit tangent space of $\mathbb{E}^3$ at o_c, namely "the set of unit vectors originating from the camera origin." To keep features within a finite FOV, one may introduce an appropriate image-space "obstacle" into the controller design (see Section 4).

2.2 Image-based Translation

Attach the body frame at the center of the sphere, so that the location of the body relative to the camera origin is given by $o_b - o_c = \boldsymbol{d}$. If $\|\boldsymbol{d}\| > \varrho$ — *i.e.* the body 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

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

(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 in the direction of

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

and is readily measurable from the projection of the body.

Let $\mathcal{B} := \{\boldsymbol{d} \in \mathbb{R}^3 : \|\boldsymbol{d}\| > \varrho\}$ denote the translations of the body origin that keep it a body radius away from the camera. We now have a diffeomorphism — a smooth and smoothly invertible function — from locations of the body to image measurements, $c_1 : \mathcal{B} \to (0,1) \times \mathrm{S}^2$, given by

$$\boxed{c_1 : \boldsymbol{d} \mapsto (\lambda, \boldsymbol{s})\,.} \tag{3}$$

The inverse of c_1 is given simply by

$$c_1^{-1}(\lambda, \boldsymbol{s}) = \frac{\varrho}{\lambda}\boldsymbol{s}\,. \tag{4}$$

2.3 Image-based Rotation

To break the rotational symmetry of our spherical rigid body, attach a visible feature point, b, to its surface, and a unit vector $\boldsymbol{a}$ tangent to the body at that point. For convenience, align the body frame so that origin coincides with the center of the body, and the unit vector $(b - o_b)/\varrho$ lies along the negative $\boldsymbol{k}_b$ axis. Hence, in the body frame $b^b = \begin{bmatrix} 0, & 0, & -\varrho, & 1 \end{bmatrix}^T$.

As we will show, the projection of b to the image-sphere, $\boldsymbol{q}_1 = \pi(b)$, encodes two rotational degrees of freedom. We encode the final degree-of-freedom by projecting a unit vector or "arrow", $\boldsymbol{a}$, tangent to the body at the point b. In practice, the vector $\boldsymbol{a}$ may be approximated by two distinguishable points on the surface of the sphere. Again

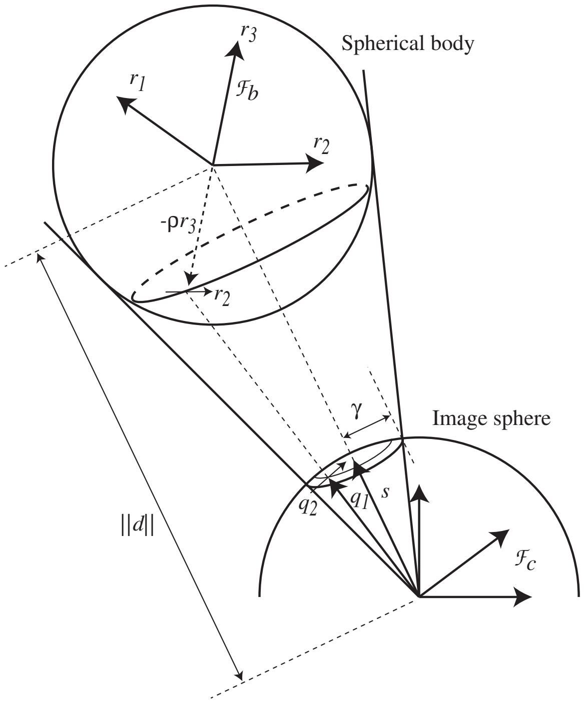


Figure 1: Projection of a spherical body with a feature point on it to the image-sphere. The image-plane measurement is given by $y = (Q, \lambda, \boldsymbol{s}) = c(H)$.

for convenience we assume the vectors body-fixed representation is simply $\boldsymbol{a}^b = \boldsymbol{e}_2$. Let $\boldsymbol{b} = b - o_c$ denote the vector from the camera origin to the body point b. Recalling that the rotation matrix R has columns $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)$, then with respect to the camera frame, we have

$$b^c = \begin{bmatrix} \boldsymbol{b}^c \\ 1 \end{bmatrix} = H b^b, \quad \text{where} \quad \boldsymbol{b}^c = \boldsymbol{d} - \varrho \boldsymbol{r}_3, \quad \text{and} \quad \boldsymbol{a}^c = R\boldsymbol{a}^b = \boldsymbol{r}_2 \,.$$

Note that $(b - o_b) \cdot \boldsymbol{a} = -\varrho\, \boldsymbol{e}_3 \cdot \boldsymbol{e}_2 = 0$.

Some configurations cause the body to occlude the feature point, b. This occurs when $(b - o_c) \cdot (o_b - b)$ becomes negative. Hence, we define a "visibility" function [5], ν, and associated "visible set" of rigid transformations, $\mathcal{V}$ by

$$\nu(H) := (\boldsymbol{d} - \varrho \boldsymbol{r}_3) \cdot \boldsymbol{r}_3 \quad \text{and} \quad \mathcal{V} := \{H \in \mathrm{SE}(3) : \nu(H) > 0\} \,. \tag{5}$$

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

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

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

We are not concerned with the length of the projection of $\boldsymbol{a}$, only the direction. Hence, consider the unit tangent map $T^1\pi$ represented in the camera frame by

$$T^1\pi : (b, \boldsymbol{a}) \mapsto (\boldsymbol{q}_1, \boldsymbol{q}_2) \quad \text{where} \tag{6}$$
$$\boldsymbol{q}_1 = \frac{\boldsymbol{b}^c}{\|\boldsymbol{b}^c\|} = \frac{\boldsymbol{d} - \varrho \boldsymbol{r}_3}{\|\boldsymbol{d} - \varrho \boldsymbol{r}_3\|} \quad \text{and} \quad \boldsymbol{q}_2 = \frac{\Gamma_{\boldsymbol{q}_1} \boldsymbol{a}^c}{\|\Gamma_{\boldsymbol{q}_1} \boldsymbol{a}^c\|} = \frac{\Gamma_{\boldsymbol{q}_1} \boldsymbol{r}_2}{\|\Gamma_{\boldsymbol{q}_1} \boldsymbol{r}_2\|}$$
$$\text{where} \quad \Gamma_{\boldsymbol{q}_1} := (I - \boldsymbol{q}_1 {\boldsymbol{q}_1}^T)\,.$$

Geometrically, $\boldsymbol{q}_2$ is a unit vector tangent to the image-sphere at the point $\boldsymbol{q}_1$. The unit vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are mutually orthogonal. Consider the plane containing the camera origin o_c, the point b, and the vector $\boldsymbol{a}$. The unit vector

$$\boldsymbol{q}_3 = \boldsymbol{q}_1 \times \boldsymbol{q}_2$$

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

$$\boxed{c_2 : H \mapsto \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{q}_3 \end{bmatrix} = Q,} \tag{7}$$

identifying $T^1\mathrm{S}^2$ with $\mathrm{SO}(3)$.

2.4 Diffeomorphism to Image-space

Claim 1. *The function $c : \mathcal{V} \to \mathcal{I}$, defined by*

$$c(H) := (c_2(H), c_1(\boldsymbol{d}))\ , \ \textit{where} \tag{8}$$
$$\mathcal{I} = \left\{ (Q, \lambda, \boldsymbol{s}) \in \mathrm{SO}(3) \times (0,1) \times \mathrm{S}^2 : \boldsymbol{q}_1 \cdot \boldsymbol{s} > \sqrt{1 - \lambda^2} \right\} \tag{9}$$
$$\textit{and } Q = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{q}_3 \end{bmatrix}$$

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

The proof is given in Appendix A

3 Image Jacobian

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

$$\begin{tikzcd}
T\mathcal{V} \arrow[r, shift left, "Tc"] \arrow[d] & T\mathcal{I} \arrow[l, shift left, "Tc^{-1}"] \arrow[d] \\
\mathcal{V} \arrow[r, "c"] & \mathcal{I} \\
T^*\mathcal{V} \arrow[u] \arrow[r, shift left, leftarrow, "T^*c"'] & T^*\mathcal{I} \arrow[u] \arrow[l, shift left, leftarrow, "T^*c^{-1}"']
\end{tikzcd}$$

We identify the tangent space $T\mathrm{SE}(3)$ of the Lie group $\mathrm{SE}(3)$ with[1]

$$T\mathrm{SE}(3) \simeq \mathrm{SE}(3) \times \mathfrak{se}(3) \simeq \mathrm{SE}(3) \times (\mathbb{R}^3 \circledS \mathbb{R}^3), \tag{10}$$

[1] The Lie algebra $\mathbb{R}^3 \circledS \mathbb{R}^3$ is $\mathbb{R}^3 \times \mathbb{R}^3$ with the Lie bracket structure found in [13].

where $\mathfrak{se}(3)$ is the Lie algebra of SE(3). The identification occurs via "right translation," i.e.

$$(H, \dot{H}) \mapsto (H, \dot{H}H^{-1}) \mapsto (H, (\boldsymbol{\omega}, \boldsymbol{v})) \tag{11}$$

where

$$H = \begin{bmatrix} R & \boldsymbol{d} \\ 0^T & 1 \end{bmatrix}, \quad \dot{H} = \begin{bmatrix} \dot{R} & \dot{\boldsymbol{d}} \\ 0^T & 0 \end{bmatrix} \quad \text{and} \quad \begin{cases} \boldsymbol{\omega} = \left(\dot{R}R^{-1}\right)^{\vee} \\ \boldsymbol{v} = -\dot{R}R^{-1}\boldsymbol{d} + \dot{\boldsymbol{d}} \end{cases}$$

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

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

where $\mathfrak{so}(3)$ is the Lie algebra of SO(3). More detail can be found in, for example [13].

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

$$T_y\mathcal{I} = T_Q\mathrm{SO}(3) \times T_\lambda(0,1) \times T_{\boldsymbol{s}}\mathrm{S}^2 \simeq \mathbb{R}^3 \times \mathbb{R} \times T_{\boldsymbol{s}}\mathrm{S}^2 \tag{12}$$

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

$$(Q, \dot{Q}) \mapsto (Q, \boldsymbol{\xi}) \quad \text{where} \quad \boldsymbol{\xi} = \left(\dot{Q}Q^{-1}\right)^{\vee}. \tag{13}$$

Hence, to compute $T_H c$ we find the mapping relating the tangent space identifications made above in (10) and (12), namely

$$(H, (\boldsymbol{\omega}, \boldsymbol{v})) \mapsto \left(y, (\boldsymbol{\xi}, \dot{\lambda}, \dot{\boldsymbol{s}})\right)$$

$$\text{where} \quad y = (Q, \lambda, \boldsymbol{s}) = c(H), \quad \begin{bmatrix} \boldsymbol{\xi} \\ \dot{\lambda} \\ \dot{\boldsymbol{s}} \end{bmatrix} = C(y) \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}$$

and

$$\begin{aligned} C(y) &:= T_H c|_{H=c^{-1}(\boldsymbol{y})} \\ &= \begin{bmatrix} I_{3\times 3} & \frac{1}{\beta}(\delta \boldsymbol{q}_1 \boldsymbol{q}_3^T - \boldsymbol{q}_2 \boldsymbol{q}_3^T + \boldsymbol{q}_3 \boldsymbol{q}_2^T) \\ 0_{1\times 3} & -\frac{\lambda^2}{\varrho} \boldsymbol{s}^T \\ -\hat{\boldsymbol{s}} & \frac{\lambda}{\varrho}(I_{3\times 3} - \boldsymbol{s}\boldsymbol{s}^T) \end{bmatrix}, \end{aligned} \tag{14}$$

where

$$\delta = \frac{\boldsymbol{s} \cdot \boldsymbol{q}_2}{\sqrt{\lambda^2 - \sin^2\phi}}, \quad \beta = \frac{\varrho}{\lambda}\left(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi}\right),$$

$$\cos\phi = \boldsymbol{s} \cdot \boldsymbol{q}_1 \quad \text{and} \quad \sin\phi = \sqrt{1 - (\boldsymbol{s} \cdot \boldsymbol{q}_1)^2}.$$

The construction of C is straight forward. The details are given in Appendix B.

To compute Tc^{-1}, and T^*c is now straight forward. Using the above representations, we have

$$T_y c^{-1} = \left(C(y)^T C(y)\right)^{-1} C(y)^T \quad \text{and} \quad T_y^* c = C(y)^T. \tag{15}$$

Note that the expression for $T_y c^{-1}$ is *not* a pseudo-inverse. The possible confusion arises since the six dimensional tangent space $T_y\mathcal{I}$ is locally embedded in $\mathbb{R}^7$. It should be noted that in many image-based visual servoing strategies employ the pseudo-inverse of the image Jacobian since the image feature points are treated as though moving freely in $\mathbb{R}^n$.

4 Controller

For the present work, we consider the case of so-called "eye-in-hand" VS, wherein the camera moves relative to the body which serves as an inertial reference frame. Let $(\boldsymbol{\Omega}, \boldsymbol{V})$ denote the angular and spatial velocities, respectively, of the camera relative to the fixed, inertial body frame. Let $G = H^{-1}$ denote the transformation of the camera frame, $\mathcal{F}_c$, relative to the inertial body frame, $\mathcal{F}_b$. Note that

$$\begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0^T & 0 \end{bmatrix} = G^{-1}\dot{G} = -\dot{H}H^{-1} = -\begin{bmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0^T & 0 \end{bmatrix}, \tag{16}$$

effectively mapping the identification of $T\mathrm{SE}(3)$ given by the right translation of $\dot{H}$ in (10) and (11) to the left translation of $\dot{G} = \frac{d}{dt}(H^{-1})$. Note that this relationship clears up, once and for all, the kinematic distinction between "eye-in-hand" servoing and the so-called "fixed-camera" configuration, wherein the camera is fixed and the body is moving.

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

$$\dot{G} = G \begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0^T & 0 \end{bmatrix} \tag{17}$$

where we treat $(\boldsymbol{\Omega}, \boldsymbol{V}) \in \mathbb{R}^3 \circledS \mathbb{R}^3$ as control inputs. We generalize this to a dynamical free rigid body in Appendix C. One possible control strategy involves planning a path $y_d(t) \in \mathcal{I}$ that moves from the initial configuration to the goal state and following the path via

$$\begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{bmatrix} = -T_y c^{-1} \begin{bmatrix} \boldsymbol{\xi}_d \\ \dot{\lambda}_d \\ \dot{\boldsymbol{s}}_{\boldsymbol{d}} \end{bmatrix} \tag{18}$$

where $\begin{bmatrix} \boldsymbol{\xi}_d, & \dot{\lambda}_d, & \dot{\boldsymbol{s}}_{\boldsymbol{d}} \end{bmatrix}^T$ is the desired velocity $\dot{y}_d$, expressed using the tangent space identification in (12). The minus sign in the above expression arises due to the identification made above in (16).

4.1 Visual Servoing via Navigation Functions

The diffeomorphism c, the visible set $\mathcal{V}$, and its relatively simple image $\mathcal{I}$, provide tremendous leverage into the VS problem. Given a desired configuration $G^* = (H^*)^{-1}$, measured through its image $y^* = (Q^*, \lambda^*, \boldsymbol{s}^*) = c(H^*)$, there are many possible image-based control strategies we can employ to achieve our objective of driving $G \to G^*$.

An open-loop strategy, such as the one above in (18), 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 Navigation Functions (NF's) [8, 9, 14]. A substantial benefit of using NF's is that they allow us to "lift" our kinematic controller to second order settings with little additional effort, while maintaining similar convergence guarantees (as we do for this problem in Appendix C). Moreover, these methods have already proven practicable for dynamic VS [5].

Let $\mathcal{D} \subset \mathcal{I}$ be compact "safe" domain. If we carefully design an artificial potential function $\varphi : \mathcal{D} \to [0,1]$, then by letting

$$\dot{y}_d = -\nabla\varphi \tag{19}$$

the control law given by (18) drives G so that y converges to y^*, except for a set of measure zero. The following definition, adapted from [8], gives a set of conditions that guarantee essentially global convergence of the above controller (18), with $\dot{y}_d$ given in (19).

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

1. *φ takes its unique minimum at $\varphi(y^*) = 0$;*

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 (18) will ensure convergence $y \xrightarrow{t\to\infty} y^*$ from all initial conditions in $\mathcal{D}$. For more information, see [8].

4.2 Computing a Safe Domain and Navigation Function

The next step is to compute a compact domain $\mathcal{D} \subset \mathcal{I}$ that is "safe" with respect to the FOV of our camera system in the sense that if $G^{-1} = H \in c^{-1}(\mathcal{D})$ then all the necessary features are visible. To illustrate, we treat the FOV as a cone originating at the camera origin, with center along $\boldsymbol{e}_3$, as shown in Figure 2. This cone reduces to a constraint on $\boldsymbol{s}$ and λ, namely

$$f(y) := \lambda \boldsymbol{s} \cdot \boldsymbol{e}_3 - \sqrt{(1-\lambda^2)(1-(\boldsymbol{s}\cdot\boldsymbol{e}_3)^2)} \geq \cos\theta$$

where θ is the angle from $\boldsymbol{e}_3$ to the edge of the FOV cone. Additionally, we constrain $\lambda \in [\lambda_{\min}, \lambda_{\max}] \subset (0,1)$ where the parameters $\lambda_{\min}$ and $\lambda_{\max}$ effectively keep the camera from moving too far from or too close to the camera body, respectively. Finally, we keep $\boldsymbol{q}_1$ from being too close to the edge of the projected circle, namely $\boldsymbol{q}_1 \cdot \boldsymbol{s} + \epsilon \geq \sqrt{1-\lambda^2}$. Putting these constraints together yields the compact manifold

$$\begin{aligned}\mathcal{D} = \big\{ y = (Q, \lambda, \boldsymbol{s}) \in \mathrm{SO}(3) \times [\lambda_{\min}, \lambda_{\max}] \times \mathrm{S}^2 :& \\ f(y) \geq \cos\theta, \lambda_{\min} \leq \lambda \leq \lambda_{\max} \big\} \subset \mathcal{I}\,,& \\ \text{where} \quad \theta \in (0, \pi/2)\,, \quad 0 < \lambda_{\min} < \lambda_{\max} < 1\,.&\end{aligned} \tag{20}$$

Clearly $\mathcal{D} \subset \mathcal{I}$. Given this domain, one must construct an NF on $\mathcal{D}$. The construction of φ represents work in progress, however, we conjecture that given the relatively simple geometry of $\mathcal{D}$, that constructing a suitable NF should be straight forward. In fact, we believe (but have not yet formally shown) that $\mathcal{D} \simeq [0,1]^5 \times \mathrm{S}^1$ which is the same topology for which an NF has already been constructed for VS by the first author and colleagues [5].

5 Conclusion

In this paper, 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 provide tremendous leverage because they shed light on the *geometry* of occlusion free servoing as well as provide a clear pathway to construct *global dynamical* visual servoing systems by using, for example, Navigation Functions.

A global, sensor-based representation of the configuration space leaves many open doors. For example, the control of underactuated and kinematically nonholonomic systems becomes possible in sensor space. The work presented in this paper represents only the tip of the iceberg. Now that we now know it is possible to globally represent rigid motion using image coordinates, we would like to construct a more general class of diffeomorphisms to the image plane that does not require designing special visual targets. We believe that with proper insight, the projection of a collection of rigidly connected feature points may be interpreted geometrically, again enabling a global representation of visible configurations. For example, perhaps depth can be described in terms of "moments", as suggested by Hamel and Mahoney [6], and orientation can be described in terms of the projection of two or three feature points.

Acknowledgments

The first author was supported in part by DARPA/ONR under grants N00014-98-1-0747 and N66001-00-C8026, and NSF under grant ECS-9873474.

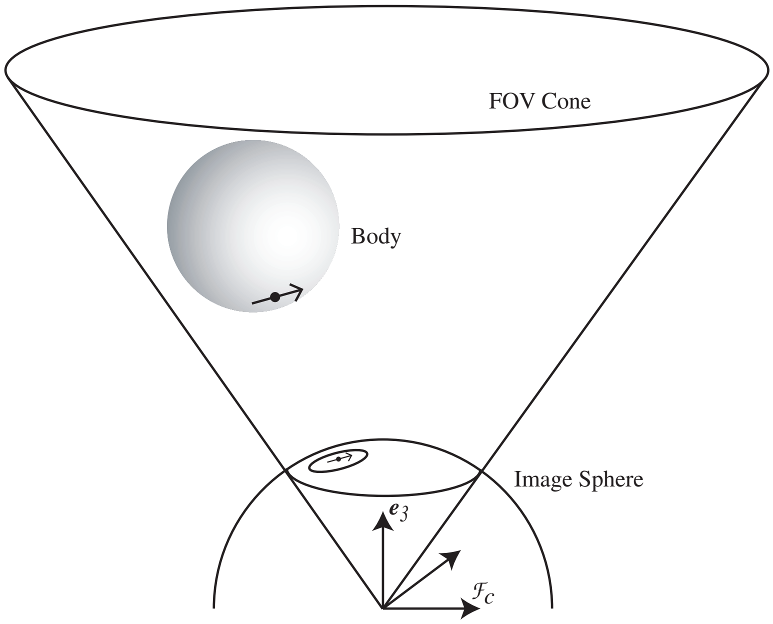


Figure 2: A simple model of the FOV cone.

A 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 $\mathcal{I}$.

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 $\mathcal{V}$. In particular:

1. Equation (3) depends on $\|\boldsymbol{d}\|^{-1}$. However, $H \in \mathcal{V}$ implies $\|\boldsymbol{d}\| > \varrho$.
2. Equation (6) depends on $\|\boldsymbol{d} - \varrho \boldsymbol{r}_3\|^{-1}$. Visibility implies $\|\boldsymbol{d}\| > \varrho$, which in turn implies $\|\boldsymbol{d} - \varrho \boldsymbol{r}_3\| \geq \|\boldsymbol{d}\| - \|\varrho \boldsymbol{r}_3\| = \|\boldsymbol{d}\| - \varrho > 0$.
3. Equation (6) depends on $\|\Gamma_{\boldsymbol{q}_1} \boldsymbol{r}_2\|^{-1}$. This blows up iff $\boldsymbol{q}_1 = \pm \boldsymbol{r}_2$, i.e.

$$\boldsymbol{q}_1 = \frac{\boldsymbol{d} - \varrho \boldsymbol{r}_3}{\|\boldsymbol{d} - \varrho \boldsymbol{r}_3\|} = \pm \boldsymbol{r}_2$$

   and hence, from (5)

$$\nu(H) = (\boldsymbol{d} - \varrho \boldsymbol{r}_3) \cdot \boldsymbol{r}_3 = \pm \|\boldsymbol{d} - \varrho \boldsymbol{r}_3\| \, \boldsymbol{r}_2 \cdot \boldsymbol{r}_3 = 0 \,, \\ \implies H \notin \mathcal{V} \,.$$

   This contradiction implies that $\|\Gamma_{\boldsymbol{q}_1} R \boldsymbol{a}^b\| > 0$ for $H \in \mathcal{V}$.

Hence, c is smooth on $\mathcal{V}$.

The image of $\mathcal{V}$ indeed is contained in $\mathcal{I}$.

To see that $c(\mathcal{V}) \subset \mathcal{I}$, let $(Q, \lambda, \boldsymbol{s}) = c(H)$. By construction of c, we have that $(Q, \lambda, \boldsymbol{s}) \in \mathrm{SO}(3) \times (0,1) \times \mathrm{S}^2$. To show that $\boldsymbol{q}_1 \cdot \boldsymbol{s} > \sqrt{1-\lambda^2}$, note from (6) that

$$\boldsymbol{q}_1 \cdot \boldsymbol{s} = \frac{(\boldsymbol{d} - \varrho \boldsymbol{r}_3) \cdot \boldsymbol{s}}{\|\boldsymbol{d} - \varrho \boldsymbol{r}_3\|} = \frac{\|\boldsymbol{d}\| - \varrho \boldsymbol{s} \cdot \boldsymbol{r}_3}{\sqrt{\varrho^2 - 2\boldsymbol{d} \cdot \boldsymbol{r}_3 + \|\boldsymbol{d}\|^2}} = \frac{1 - \lambda\alpha}{\sqrt{1 - 2\lambda\alpha + \lambda^2}}$$

where $\alpha = \boldsymbol{s} \cdot \boldsymbol{r}_3$. From (5), $\alpha > \lambda$ to ensure $\nu(H) > 0$. Since the right hand side reaches its minimum of $\sqrt{1-\lambda^2}$ for $\alpha = \lambda$, we have that $\boldsymbol{q}_1 \cdot \boldsymbol{s} > \sqrt{1-\lambda^2}$.

The function c is bijective.

Consider any $H \in \mathcal{V}$, with rotation $R = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_1 & \boldsymbol{r}_1 \end{bmatrix}$ and translation $\boldsymbol{d}$ as usual. Let $c(H) = (Q, \lambda, \boldsymbol{s}) \in \mathcal{I}$, where $Q = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{q}_3 \end{bmatrix}$. Given $(\lambda, \boldsymbol{s})$, recovering the translation from (4) is trivial, namely $\boldsymbol{d} = c_1^{-1}(\lambda, \boldsymbol{s}) \in \mathcal{B}$. Recovering the rotation from $(Q, \lambda, \boldsymbol{s})$ requires a bit more care.

Consider the triangle defined by o_c, o_b and b. The points o_c and $o_b = o_c + \boldsymbol{d}$ are known, but b is unknown. Let $\boldsymbol{b} = b - o_c$ and note that

$$\boldsymbol{b}^c = \beta \boldsymbol{q}_1\,, \quad \text{for some} \quad \beta \in \mathbb{R}\,.$$

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

$$\cos\phi = \boldsymbol{q}_1 \cdot \boldsymbol{s} > \sqrt{1-\lambda^2} > 0 \implies \phi \in [0, \pi/2)\,.$$

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

$$\beta^2 + \|\boldsymbol{d}\|^2 - 2\beta\|\boldsymbol{d}\|\cos\phi = \varrho^2 \implies \beta = \frac{\varrho}{\lambda}\left(\cos\phi + \sigma\sqrt{\lambda^2 - \sin^2\phi}\right)\,.$$

where $\sigma = \pm 1$. In general, there may be zero, one or real solutions for β. However, the requirement that $\boldsymbol{s} \cdot \boldsymbol{q}_1 > \sqrt{1-\lambda^2}$ implies that $\phi \in [0, \phi_{\max})$, where $\phi_{\max} = \arccos\sqrt{1-\lambda^2}$ (where arccos is taken in the first quadrant). This implies $\sin^2\phi < \lambda$, and thus there are two algebraic solutions for β. Note that $\boldsymbol{r}_3 = (\boldsymbol{d} - \boldsymbol{b})/\varrho$, and thus

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

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

$$\boldsymbol{b} = \|\boldsymbol{b}\|\boldsymbol{q}_1\,, \quad \text{where} \quad \|\boldsymbol{b}\| = \frac{\varrho}{\lambda}\left(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi}\right)\,. \tag{21}$$

Thus, $\boldsymbol{r}_3 = (\boldsymbol{d} - \boldsymbol{b})/\varrho$.

From (6), $\boldsymbol{r}_2 \in \operatorname{span}\{\boldsymbol{q}_1, \boldsymbol{q}_2\}$, namely

$$\boldsymbol{r}_2 = \alpha_1 \boldsymbol{q}_1 + \alpha_2 \boldsymbol{q}_2 \tag{22}$$

for some α_1 and α_2. Note from (6) that $\alpha_2 > 0$. Moreover, $\boldsymbol{r}_3 \cdot \boldsymbol{r}_2 = 0$, hence

$$\overbrace{\begin{bmatrix} \boldsymbol{r}_3^T & 0 & 0 \\ I & -\boldsymbol{q}_1 & -\boldsymbol{q}_2 \end{bmatrix}}^{M,\,4\times 5} \overbrace{\begin{bmatrix} \boldsymbol{r}_2 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}}^{v,\,5\times 1} = 0 \quad \text{and} \quad \|\boldsymbol{r}_2\| = 1,\, \alpha_2 > 0\,.$$

The matrix M has a one-dimensional kernel since $\boldsymbol{q}_1, \boldsymbol{q}_2$ are linearly independent and hence there are two possible solutions to $Mv = 0$ for $\boldsymbol{r}_2, \alpha_2, \alpha_2$ subject to $\|\boldsymbol{r}_2\| = 1$, the ambiguity of which is eliminated since $\alpha_2 > 0$.

Combining the above computations yields the unique inverse to c,

$$\begin{aligned} c^{-1} : (Q, \lambda, \boldsymbol{s}) \mapsto H = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 & \boldsymbol{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where} \\ \boldsymbol{d} = \frac{\varrho}{\lambda}\boldsymbol{s}, \quad \boldsymbol{r}_3 = \frac{1}{\lambda}\left(\boldsymbol{s} - \left(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi}\right)\boldsymbol{q}_1\right), \\ \boldsymbol{r}_2 = \left(\boldsymbol{q}_2 - \frac{\boldsymbol{r}_3 \cdot \boldsymbol{q}_2}{\boldsymbol{r}_3 \cdot \boldsymbol{q}_1}\boldsymbol{q}_1\right) \Big/ \left\|\boldsymbol{q}_2 - \frac{\boldsymbol{r}_3 \cdot \boldsymbol{q}_2}{\boldsymbol{r}_3 \cdot \boldsymbol{q}_1}\boldsymbol{q}_1\right\|, \quad \text{and} \quad \boldsymbol{r}_1 = \boldsymbol{r}_2 \times \boldsymbol{r}_3\,. \end{aligned} \tag{23}$$

The function c^{-1} is smooth.

Finally, we need only show that c^{-1} is smooth. But, 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 $\boldsymbol{r}_2$. First, note that $\boldsymbol{r}_3 \cdot \boldsymbol{q}_1 = \boldsymbol{r}_3 \cdot \boldsymbol{b}/\|\boldsymbol{b}\| = \nu(H)/\|\boldsymbol{b}\| > 0$. Also, since $\boldsymbol{q}_2$ and $\boldsymbol{q}_1$ 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 $c : \mathcal{V} \approx \mathcal{I}$. □

B Computation of the Image Jacobian

We compute the image Jacobian matrix given by (14) algebraically in B.1, and then verify this geometrically in Section B.2.

B.1 Algebraic Computation of Image Jacobian

B.1.1 Computation of $\dot{\lambda}$.

Recall

$$\dot{\boldsymbol{d}} = \boldsymbol{\omega} \times \boldsymbol{d} + \boldsymbol{v} \tag{24}$$

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

$$\dot{\lambda} = -\frac{\varrho}{\|\boldsymbol{d}\|^3}\boldsymbol{d} \cdot \dot{\boldsymbol{d}} = -\frac{\lambda^2}{\varrho}\boldsymbol{s} \cdot \boldsymbol{v}\,. \tag{25}$$

B.1.2 Computation of $\dot{\boldsymbol{s}}$.

Recall $\boldsymbol{s} = \boldsymbol{d}/\|\boldsymbol{d}\|$. Equation (24) implies

$$\dot{\boldsymbol{s}} = -\boldsymbol{s} \times \boldsymbol{\omega} + \frac{\lambda}{\varrho}(I - \boldsymbol{s}\boldsymbol{s}^T)\boldsymbol{v}. \tag{26}$$

B.1.3 Computation of $\boldsymbol{\xi} = (\dot{Q}Q^{-1})^\vee$.

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

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

to compute

$$\boldsymbol{\xi} = Q(Q^{-1}\dot{Q})^\vee. \tag{27}$$

Proceeding in this manner, we have

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

which implies

$$(Q^{-1}\dot{Q})^\vee = \begin{bmatrix} -\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3 \\ -\boldsymbol{q}_3 \cdot \dot{\boldsymbol{q}}_1 \\ \boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_1 \end{bmatrix}. \tag{28}$$

It follows that we need to compute the three quantities $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_1$, $\boldsymbol{q}_3 \cdot \dot{\boldsymbol{q}}_1$, $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3$ in terms of $\boldsymbol{\omega}$ and $\boldsymbol{v}$ in order to get an expression of $\boldsymbol{\xi}$ in terms of $\boldsymbol{\omega}$ and $\boldsymbol{v}$. We remark that we chose $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_1$, $\boldsymbol{q}_3 \cdot \dot{\boldsymbol{q}}_1$, $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3$ rather then the other possible three quantities because the involved computation is relatively simple.

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

$$\boldsymbol{q}_1 = \frac{(\boldsymbol{d} - \varrho \boldsymbol{r}_3)}{\|\boldsymbol{b}\|}$$

where $\boldsymbol{b} = \boldsymbol{d} - \varrho\boldsymbol{r}_3$. From (24) and $\dot{\boldsymbol{r}}_3 = \boldsymbol{\omega} \times \boldsymbol{r}_3$, it follows

$$\begin{aligned} \dot{\boldsymbol{q}}_1 &= \frac{1}{\|\boldsymbol{b}\|}(\dot{\boldsymbol{d}} - \varrho\dot{\boldsymbol{r}}_3) + (\boldsymbol{d} - \varrho\boldsymbol{r}_3)\frac{d}{dt}(\|\boldsymbol{b}\|^{-1}) \\ &= \boldsymbol{\omega} \times \boldsymbol{q}_1 + \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{v} + A\,\boldsymbol{q}_1 \end{aligned} \tag{29}$$

where $A = \|\boldsymbol{b}\|\frac{d}{dt}(\|\boldsymbol{b}\|^{-1})$. From (29), it follows

$$\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_1 = \boldsymbol{q}_3 \cdot \boldsymbol{\omega} + \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{q}_2 \cdot \boldsymbol{v} \tag{30}$$

$$\boldsymbol{q}_3 \cdot \dot{\boldsymbol{q}}_1 = -\boldsymbol{q}_2 \cdot \boldsymbol{\omega} + \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{q}_3 \cdot \boldsymbol{v}. \tag{31}$$

We now start to compute $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3$. Recall

$$\boldsymbol{r}_2 = \frac{\boldsymbol{q}_2 - \delta\boldsymbol{q}_1}{\|\boldsymbol{q}_2 - \delta\boldsymbol{q}_1\|}, \quad \delta := \frac{\boldsymbol{r}_3 \cdot \boldsymbol{q}_2}{\boldsymbol{r}_3 \cdot \boldsymbol{q}_1}. \tag{32}$$

This implies

$$\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\| = \frac{1}{\|\boldsymbol{q}_2 - \delta\boldsymbol{q}_1\|} = \frac{1}{\sqrt{1+\delta^2}}, \qquad \boldsymbol{q}_1 \cdot \boldsymbol{r}_2 = -\frac{\delta}{\sqrt{1+\delta^2}} \tag{33}$$

which will be used later in computations. From (32), we have $\boldsymbol{q}_3 = \boldsymbol{q}_1 \times \boldsymbol{q}_2 = (\boldsymbol{q}_1 \times \boldsymbol{r}_2)/\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|$. It follows

$$\dot{\boldsymbol{q}}_3 = \frac{1}{\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|}(\dot{\boldsymbol{q}}_1 \times \boldsymbol{r}_2 + \boldsymbol{q}_1 \times \dot{\boldsymbol{r}}_2) + B\,\boldsymbol{q}_3 \tag{34}$$

where $B = \|\boldsymbol{q}_1 \times \boldsymbol{r}_2\| \frac{d}{dt}(\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|^{-1})$. Hence,

$$\begin{aligned} \boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3 &= (1+\delta^2)^{\frac{1}{2}}(\boldsymbol{r}_2 \cdot (\boldsymbol{q}_2 \times \dot{\boldsymbol{q}}_1) - \dot{\boldsymbol{r}}_2 \cdot \boldsymbol{q}_3) \\ &= (1+\delta^2)^{\frac{1}{2}}(\boldsymbol{r}_2 \cdot (\boldsymbol{q}_2 \times \dot{\boldsymbol{q}}_1) + \boldsymbol{r}_2 \cdot \dot{\boldsymbol{q}}_3) \end{aligned} \tag{35}$$

where in the second equality we used the fact that $\boldsymbol{r}_2 \cdot \boldsymbol{q}_3 = 0$ from (32) implies $\boldsymbol{r}_2 \cdot \dot{\boldsymbol{q}}_3 = -\dot{\boldsymbol{r}}_2 \cdot \boldsymbol{q}_3$.

Using (29), we have

$$\begin{aligned} \boldsymbol{r}_2 \cdot (\boldsymbol{q}_2 \times \dot{\boldsymbol{q}}_1) &= -(\boldsymbol{q}_1 \cdot \boldsymbol{r}_2)(\boldsymbol{q}_2 \cdot \boldsymbol{\omega}) + \frac{1}{\|\boldsymbol{b}\|}(\boldsymbol{r}_2 \times \boldsymbol{q}_2) \cdot \boldsymbol{v} \\ &= -(\boldsymbol{q}_1 \cdot \boldsymbol{r}_2)(\boldsymbol{q}_2 \cdot \boldsymbol{\omega}) - \frac{\delta}{\|\boldsymbol{b}\|\sqrt{1+\delta^2}}(\boldsymbol{q}_3 \cdot \boldsymbol{v}) \end{aligned} \tag{36}$$

where we used $\boldsymbol{r}_2 \times \boldsymbol{q}_2 = -\delta(1+\delta^2)^{-\frac{1}{2}}\boldsymbol{q}_3$, which comes from (32) and (33).

From (34), $\boldsymbol{r}_2 \cdot \boldsymbol{q}_3 = 0$ and $\dot{\boldsymbol{r}}_2 = \boldsymbol{\omega} \times \boldsymbol{r}_2$, we get

$$\begin{aligned} \boldsymbol{r}_2 \cdot \dot{\boldsymbol{q}}_3 &= \frac{1}{\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|}\boldsymbol{q}_1 \cdot ((\boldsymbol{\omega} \cdot \boldsymbol{r}_2)\boldsymbol{r}_2 - \boldsymbol{\omega}) \\ &= \frac{1}{\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|}\boldsymbol{q}_1 \cdot (\|\boldsymbol{q}_1 \times \boldsymbol{r}_2\|(\boldsymbol{q}_2 \cdot \boldsymbol{\omega} - \delta \boldsymbol{q}_1 \cdot \boldsymbol{\omega})\boldsymbol{r}_2 - \boldsymbol{\omega}) \\ &= (\boldsymbol{q}_1 \cdot \boldsymbol{r}_2)(\boldsymbol{q}_2 \cdot \boldsymbol{\omega}) - \frac{(\boldsymbol{q}_1 \cdot \boldsymbol{\omega})}{\sqrt{1+\delta^2}} \end{aligned} \tag{37}$$

where we used (32) in the second equality and (33) in the third equality.

Plugging (36), (37) and (33) into (35), we get

$$\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3 = -\boldsymbol{q}_1 \cdot \boldsymbol{\omega} - \frac{\delta}{\|\boldsymbol{b}\|}\boldsymbol{q}_3 \cdot \boldsymbol{v} \tag{38}$$

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

$$\delta = \frac{\boldsymbol{s} \cdot \boldsymbol{q}_2}{\sqrt{\lambda^2 - \sin^2\phi}}, \quad \sin\phi = \sqrt{1 - (\boldsymbol{s} \cdot \boldsymbol{q}_1)^2}. \tag{39}$$

We are now in a position to compute ξ in (27). From (27) and (28) together with (30), (31), (38), we get

$$\boldsymbol{\xi} = \boldsymbol{\omega} + \frac{1}{\|\boldsymbol{b}\|}(\delta \boldsymbol{q}_1 \boldsymbol{q}_3^T - \boldsymbol{q}_2 \boldsymbol{q}_3^T + \boldsymbol{q}_3 \boldsymbol{q}_2^T)\boldsymbol{v} \tag{40}$$

where δ is given by (39) or (32), and $\|\boldsymbol{b}\|$ can in expressed in terms of $\boldsymbol{y} = (Q, \lambda, \boldsymbol{s})$ as follows:

$$\|\boldsymbol{b}\| = \frac{\varrho}{\lambda(\cos\phi - \sqrt{\lambda^2 - \sin^2\phi})}, \quad \left\{ \begin{array}{l} \cos\phi = \boldsymbol{s} \cdot \boldsymbol{q}_1, \\ \sin\phi = \sqrt{1 - (\boldsymbol{s} \cdot \boldsymbol{q}_1)^2} \end{array} \right. . \tag{41}$$

B.1.4 Computation of the Jacobian.

The tangent map of the map $\boldsymbol{y} = c(H)$ in the following relation

$$\begin{bmatrix} \boldsymbol{\xi} \\ \dot{\lambda} \\ \dot{\boldsymbol{s}} \end{bmatrix} = T_H c \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}$$

is written as

$$T_H c|_{H=c^{-1}(\boldsymbol{y})} = \begin{bmatrix} I_{3\times 3} & \frac{1}{||\boldsymbol{b}||}(\delta \boldsymbol{q}_1\boldsymbol{q}_3^T - \boldsymbol{q}_2\boldsymbol{q}_3^T + \boldsymbol{q}_3\boldsymbol{q}_2^T) \\ 0_{1\times 3} & -\frac{\lambda^2}{\varrho}\boldsymbol{s}^T \\ -\hat{\boldsymbol{s}} & \frac{\lambda}{\varrho}(I_{3\times 3} - \boldsymbol{s}\boldsymbol{s}^T) \end{bmatrix}$$

where δ is given by (39), and $||\boldsymbol{b}||$ is given by (41).

B.2 Geometric Computation of Image Jacobian

Let p^b denote an arbitrary point fixed in the body frame, expressed in the camera frame as $p = Hp^b$. Let $\boldsymbol{p} = p - o_c$ denote the vector from the camera origin to the point p. Note that

$$\dot{p} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ 0 \end{bmatrix} = \dot{H}p^b = SHp^b = SHp$$

hence, we may write

$$\dot{\boldsymbol{p}} = \begin{bmatrix} -\hat{\boldsymbol{p}} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} \quad \text{and} \quad \frac{d}{dt}||\boldsymbol{p}|| = \frac{1}{||\boldsymbol{p}||}\boldsymbol{p}\cdot\dot{\boldsymbol{p}} = \frac{1}{||\boldsymbol{p}||}\boldsymbol{p}\cdot\boldsymbol{v}\,.$$

Thus, it follows that

$$\frac{\boldsymbol{p}}{||\boldsymbol{p}||} = \frac{1}{||\boldsymbol{p}||}\dot{\boldsymbol{p}} - \boldsymbol{p}\frac{1}{||\boldsymbol{p}||^2}\frac{d}{dt}||\boldsymbol{p}|| = \frac{1}{||\boldsymbol{p}||}\begin{bmatrix} -\widehat{\boldsymbol{p}} & (I - ||\boldsymbol{p}||^{-2}\boldsymbol{p}\boldsymbol{p}^T) \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}.$$

B.2.1 Computation of $\dot{\lambda}$ and $\dot{s}$.

It follows from above that

$$\dot{\lambda} = \begin{bmatrix} 0^T & -\frac{\lambda^2}{\varrho}\boldsymbol{s}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} \quad \text{and} \quad \dot{\boldsymbol{s}} = \begin{bmatrix} -\hat{\boldsymbol{s}} & \frac{\lambda}{\varrho}(I - \boldsymbol{s}\boldsymbol{s}^T) \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}. \tag{42}$$

B.2.2 Computation of $\dot{\boldsymbol{q}}_1$.

Recalling that $\boldsymbol{q}_1 = \boldsymbol{b}/||\boldsymbol{b}||$ we have

$$\dot{\boldsymbol{q}}_1 = \begin{bmatrix} -\widehat{\boldsymbol{q}_1} & ||\boldsymbol{b}||^{-1}(I - \boldsymbol{q}_1\boldsymbol{q}_1^T) \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} \tag{43}$$

where $||\boldsymbol{b}||$ is given by (21).

B.2.3 Computation of $\dot{\boldsymbol{q}}_3$.

Consider the line defined by

$$\ell = \{b + \alpha \boldsymbol{r}_2, \forall \alpha \in \mathbb{R}\},$$

which, together with the camera origin, determines the plane Π. Let $l_0 \in \ell$ denote the point on the line closest to the camera origin, and $\boldsymbol{l} = l - o_c$ the corresponding vector from the camera origin, i.e $\boldsymbol{l}\cdot\boldsymbol{r}_2 = 0$. Hence

$$\boldsymbol{q}_3 = \frac{\boldsymbol{l}}{||\boldsymbol{l}||} \times \boldsymbol{r}_2, \quad \text{where} \quad \boldsymbol{l} = \left(I - \boldsymbol{r}_2\boldsymbol{r}_2^T\right)\boldsymbol{b}\,.$$

To compute $\dot{\boldsymbol{q}}_3$, we simply compute $\frac{d}{dt}(\boldsymbol{l}/||\boldsymbol{l}||)$ and $\dot{\boldsymbol{r}}_2$, and use the product rule. Although it is tempting to simply write

$$\frac{d}{dt}\frac{\boldsymbol{l}}{||\boldsymbol{l}||} = \frac{1}{||\boldsymbol{l}||}\begin{bmatrix}-\hat{\boldsymbol{l}} & (I - ||\boldsymbol{l}||^{-2}\boldsymbol{l}\,\boldsymbol{l}^T)\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{v}\end{bmatrix} \quad \text{(wrong!)},$$

it is important to note that $\boldsymbol{l}$ is not fixed with respect to the body — it also depends on the camera location. In fact, for pure translations parallel to $\boldsymbol{r}_2$, $\dot{\boldsymbol{l}} = 0$. More generally,

$$\frac{d}{dt}\frac{\boldsymbol{l}}{||\boldsymbol{l}||} = \frac{1}{||\boldsymbol{l}||}\begin{bmatrix}-\hat{\boldsymbol{l}} & \boldsymbol{q}_3\boldsymbol{q}_3{}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{v}\end{bmatrix}.$$

Noting that $\dot{\boldsymbol{r}}_2 = -\widehat{\boldsymbol{r}_2}\boldsymbol{\omega}$ yields

$$\dot{\boldsymbol{q}}_3 = -\boldsymbol{r}_2 \times \left(\frac{d}{dt}\frac{\boldsymbol{l}}{||\boldsymbol{l}||}\right) + \frac{1}{||\boldsymbol{l}||}\boldsymbol{l} \times \dot{\boldsymbol{r}_2} = \frac{1}{||\boldsymbol{l}||}\begin{bmatrix}-\widehat{\boldsymbol{l} \times \boldsymbol{r}_2} & -\widehat{\boldsymbol{r}_2}\boldsymbol{q}_3\boldsymbol{q}_3{}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{v}\end{bmatrix}.$$

Hence

$$\dot{\boldsymbol{q}}_3 = \begin{bmatrix}-\widehat{\boldsymbol{q}_3} & -||\boldsymbol{l}||^{-2}\boldsymbol{l}\boldsymbol{q}_3{}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{v}\end{bmatrix}. \tag{44}$$

B.2.4 Computation of $\dot{q}_2$.

Note that $\boldsymbol{q}_2 = \boldsymbol{q}_3 \times \boldsymbol{q}_1$, we have

$$\begin{aligned}
\dot{\boldsymbol{q}}_2 &= \boldsymbol{q}_3 \times \dot{\boldsymbol{q}}_1 - \boldsymbol{q}_1 \times \dot{\boldsymbol{q}}_3\\
&= \widehat{\boldsymbol{q}_3}\left(-\widehat{\boldsymbol{q}_1}\boldsymbol{\omega} + ||\boldsymbol{b}||^{-1}(I - \boldsymbol{q}_1\boldsymbol{q}_1^T)\boldsymbol{v}\right) - \widehat{\boldsymbol{q}_1}\left(-\widehat{\boldsymbol{q}_3}\boldsymbol{\omega} - ||\boldsymbol{l}||^{-2}\boldsymbol{l}\boldsymbol{q}_3 \cdot \boldsymbol{v}\right)\\
&= -\widehat{\boldsymbol{q}_2}\boldsymbol{\omega} + ||\boldsymbol{b}||^{-1}\left(\widehat{\boldsymbol{q}_3} - \boldsymbol{q}_2\boldsymbol{q}_1^T\right)\boldsymbol{v} + ||\boldsymbol{l}||^{-2}\,\widehat{\boldsymbol{q}_1}\boldsymbol{l}\,\boldsymbol{q}_3 \cdot \boldsymbol{v}\,.
\end{aligned}$$

B.2.5 Computation of ξ.

As in B.1, we compute

$$\boldsymbol{\xi} = Q(Q^{-1}\dot{Q})^{\vee}\,.$$

For convenience, we compute $\boldsymbol{q}_1 \cdot \dot{\boldsymbol{q}}_2$, $\boldsymbol{q}_1 \cdot \dot{\boldsymbol{q}}_3$ and $\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3$. Note that

$$||\boldsymbol{l}||^2 = ||\boldsymbol{b}||^2(1 - (\boldsymbol{r}_2 \cdot \boldsymbol{q}_1)^2), \quad \boldsymbol{q}_1 \cdot \boldsymbol{l} = ||\boldsymbol{b}||(1 - (\boldsymbol{r}_2 \cdot \boldsymbol{q}_1)^2),$$
$$\text{and} \quad \boldsymbol{q}_2 \cdot \boldsymbol{l} = -||\boldsymbol{l}||\boldsymbol{r}_2 \cdot \boldsymbol{q}_1\,,$$

and hence

$$\begin{aligned}
\boldsymbol{q}_1 \cdot \dot{\boldsymbol{q}}_2 &= -\boldsymbol{q}_1 \cdot (\boldsymbol{q}_2 \times \boldsymbol{\omega}) + ||\boldsymbol{b}||^{-1}\boldsymbol{q}_1 \cdot (\boldsymbol{q}_3 \times \boldsymbol{v}) = -\boldsymbol{q}_3 \cdot \boldsymbol{\omega} - ||\boldsymbol{b}||^{-1}\boldsymbol{q}_2 \cdot \boldsymbol{v},\\
\boldsymbol{q}_1 \cdot \dot{\boldsymbol{q}}_3 &= -\boldsymbol{q}_1 \cdot (\boldsymbol{q}_3 \times \boldsymbol{\omega}) - ||\boldsymbol{l}||^{-2}(\boldsymbol{q}_1 \cdot \boldsymbol{l})(\boldsymbol{q}_3 \cdot \boldsymbol{v}) = \boldsymbol{q}_2 \cdot \boldsymbol{\omega} - ||\boldsymbol{b}||^{-1}\boldsymbol{q}_3 \cdot \boldsymbol{v}
\end{aligned}$$

and, using $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ found in (23),

$$\begin{aligned}
\boldsymbol{q}_2 \cdot \dot{\boldsymbol{q}}_3 &= -\boldsymbol{q}_2 \cdot (\boldsymbol{q}_3 \times \boldsymbol{\omega}) - ||\boldsymbol{l}||^2(\boldsymbol{q}_2 \cdot \boldsymbol{l})(\boldsymbol{q}_3 \cdot \boldsymbol{v})\\
&= -\boldsymbol{q}_1 \cdot \boldsymbol{\omega} - ||\boldsymbol{b}||^{-1}\frac{\boldsymbol{r}_3 \cdot \boldsymbol{q}_2}{\boldsymbol{r}_3 \cdot \boldsymbol{q}_1}\boldsymbol{q}_3 \cdot \boldsymbol{v}\\
&= -\boldsymbol{q}_1 \cdot \boldsymbol{\omega} - ||\boldsymbol{b}||^{-1}\frac{\boldsymbol{q}_2 \cdot \boldsymbol{s}}{\sqrt{\lambda^2 - \sin^2\phi}}\boldsymbol{q}_3 \cdot \boldsymbol{v}
\end{aligned}$$

Hence

$$\boldsymbol{\xi} = \boldsymbol{\omega} + ||\boldsymbol{b}||^{-1}\left(\frac{\boldsymbol{q}_2 \cdot \boldsymbol{s}}{\sqrt{\lambda^2 - \sin^2\phi}}\boldsymbol{q}_1\boldsymbol{q}_3^T - \boldsymbol{q}_2\boldsymbol{q}_3^T + \boldsymbol{q}_3\boldsymbol{q}_2^T\right)\boldsymbol{v}$$

as before in Section B.1.

C Application to Dynamical Rigid Body Servoing

For second order systems, φ serves as an artificial potential function which generates the force $T_y^* c\,(\mathbf{d}\varphi)^T$. Adding a suitable damping force, $B : T\mathcal{V} \to T^*\mathcal{V}$, yields the controller

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix} = T_y^* c\,(\mathbf{d}\varphi)^T - B(\boldsymbol{\Omega}, \boldsymbol{V})\,, \quad \text{where} \quad y = c(H) = c(G^{-1}) \tag{45}$$

and $\boldsymbol{F} \in T_G^*\mathcal{V} \simeq \mathbb{R}^3 \circledS \mathbb{R}^3$ represents the torque and force applied to the camera. Assuming the rigid body is not subject to any additional external forces (or such an external potential or gravity) or assuming such forces can be cancelled by our controller, we model with the standard rigid body dynamical equations

$$\begin{aligned} \dot{G} &= G \begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0^T & 0 \end{bmatrix}, \\ \dot{\boldsymbol{\Pi}} &= \boldsymbol{\Pi} \times \boldsymbol{\Omega} + \boldsymbol{F}_1\,, \\ \dot{\mathbf{P}} &= \boldsymbol{P} \times \boldsymbol{\Omega} + \boldsymbol{F}_2\,, \end{aligned} \tag{46}$$

where $\boldsymbol{\Pi}, \boldsymbol{P}$ are the linear and angular momenta, respectively, given by

$$\boldsymbol{\Pi} = \mathbf{I}\boldsymbol{\Omega} \qquad \boldsymbol{P} = m\,\boldsymbol{V}$$

where m is the mass of the camera rigid body system, and $\mathbf{I} = \mathbf{I}^T > 0$ is the angular inertia matrix.

Note that for our controller, we do not need to know the inertia of the camera, and yet we are still guaranteed asymptotic convergence. In fact, one can show that convergence $H \xrightarrow{t\to\infty} H^*$ is guaranteed from all initially conditions on $T^*\mathcal{D}$ whose energy is less than 1. For more information, see [8].

C.0.6 If the camera frame is not at the center of mass.

Attach a frame $\mathcal{F}_c$ at the pinhole of the camera, $\mathcal{F}_{\mathrm{cm}}$ at the center of mass of the camera + robot system, and $\mathcal{F}_b$ to the center of the object being observed. Here we assume that $\mathcal{F}_b$ is the inertial frame. Let $G \in \mathrm{SE}(3)$ be the transformation from $\mathcal{F}_c$ to $\mathcal{F}_b$ and $G_0 \in \mathrm{SE}(3)$ be the transformation from $\mathcal{F}_{\mathrm{cm}}$ to $\mathcal{F}_c$. *Assume that G_0 does not change in time.* The composition $G_{\mathrm{cm}} := GG_0$ is the transformation from $\mathcal{F}_{\mathrm{cm}}$ to $\mathcal{F}_b$. The body-fixed velocity of the frame $\mathcal{F}_{\mathrm{cm}}$ is given by

$$G_{\mathrm{cm}}^{-1}\dot{G}_{\mathrm{cm}} = (GG_0)^{-1}\frac{d}{dt}(GG_0) = G_0^{-1}(G^{-1}\dot{G})G_0. \tag{47}$$

Let

$$G_{\mathrm{cm}}^{-1}\dot{G}_{\mathrm{cm}} = \begin{bmatrix} \widehat{\boldsymbol{\Omega}}_{\mathrm{cm}} & \boldsymbol{V}_{\mathrm{cm}} \\ 0 & 0 \end{bmatrix}, \quad G^{-1}\dot{G} = \begin{bmatrix} \widehat{\boldsymbol{\Omega}} & \boldsymbol{V} \\ 0 & 0 \end{bmatrix}, \quad G_0^{-1} = \begin{bmatrix} B_0 & b_0 \\ 0 & 1 \end{bmatrix}. \tag{48}$$

Notice that G_0^{-1} is the transformation from frame $\mathcal{F}_c$ to frame $\mathcal{F}_{\mathrm{cm}}$. By (47) and (48), it follows

$$(\boldsymbol{\Omega}_{\mathrm{cm}}, \boldsymbol{V}_{\mathrm{cm}}) = (B_0\boldsymbol{\Omega},\, -B_0\boldsymbol{\Omega} \times b_0 + B_0\boldsymbol{V}_c),$$

or

$$\begin{bmatrix} \boldsymbol{\Omega}_{\mathrm{cm}} \\ \boldsymbol{V}_{\mathrm{cm}} \end{bmatrix} = \begin{bmatrix} B_0 & 0 \\ \hat{b}_0 B_0 & B_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V}_c \end{bmatrix}.$$

Readers with knowledge of Lie groups will readily see that what we computed is the adjoint:

$$(\boldsymbol{\Omega}_{\mathrm{cm}}, \boldsymbol{V}_{\mathrm{cm}}) = \mathrm{Ad}_{G_0^{-1}}(\boldsymbol{\Omega}, \boldsymbol{V}_c).$$

We now study how torques and forces are transformed. Let $(\boldsymbol{\tau}_c, \boldsymbol{f}_c)$ be the force measured in the frame $\mathcal{F}_c$ and $(\boldsymbol{\tau}_{\mathrm{cm}}, \boldsymbol{f}_{\mathrm{cm}})$. One can easily see the following relationship:

$$\begin{bmatrix} \boldsymbol{\tau}_{\mathrm{cm}} \\ \boldsymbol{f}_{\mathrm{cm}} \end{bmatrix} = \left(\begin{bmatrix} B_0 & 0 \\ \hat{b}_0 B_0 & B_0 \end{bmatrix}^{-1} \right)^T \begin{bmatrix} \boldsymbol{\tau}_c \\ \boldsymbol{f}_c \end{bmatrix} = \begin{bmatrix} B_0 & \hat{b}_0 B_0 \\ 0 & B_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_c \\ \boldsymbol{f}_c \end{bmatrix},$$

which can be compactly written as

$$(\boldsymbol{\tau}_{\mathrm{cm}}, \boldsymbol{f}_{\mathrm{cm}}) = \mathrm{Ad}^*_{G_0}(\boldsymbol{\tau}_c, \boldsymbol{f}_c).$$

The equations of motion in frame $\mathcal{F}_{\mathrm{cm}}$ is given by

$$\begin{aligned} \boldsymbol{\Pi}_{\mathrm{cm}} &= \boldsymbol{\Pi}_{\mathrm{cm}} \times \boldsymbol{\Omega}_{\mathrm{cm}} + \boldsymbol{\tau}_{\mathrm{cm}} \\ \dot{\boldsymbol{P}}_{\mathrm{cm}} &= \boldsymbol{P}_{\mathrm{cm}} \times \boldsymbol{\Omega}_{\mathrm{cm}} + \boldsymbol{f}_{\mathrm{cm}} \end{aligned}$$

with $\boldsymbol{\Pi}_{\mathrm{cm}} = \mathbf{I}\boldsymbol{\Omega}_{\mathrm{cm}}$ and $\boldsymbol{P}_{\mathrm{cm}} = m\boldsymbol{V}_{\mathrm{cm}}$ where $\mathbf{I}$ is the inertia matrix of the camera + robot system with respect to frame $\mathcal{F}_{\mathrm{cm}}$ and m is the mass of the camera + robot system. Here, we assumed that the gravity effect has been cancelled out by control.

References

[1] Alessandro Chiuso, Roger Brocket, and Stefano Soatto. Optimal structure from motion: local ambiguities and global estimates. *International Journal of Computer Vision*, pages 195–228, September 2000.

[2] Peter I. Corke and Seth A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, 2001.

[3] Noah Cowan. Binocular visual servoing with a limited field of view. In *Mathematical Theory of Networks and Systems*, Notre Dame, Indiana, August 2002.

[4] Noah J. Cowan and Dong Eui Chang. *Control Problems in Robotics and Automation*, chapter Toward Geometric Visual Servoing. Springer, 2002.

[5] Noah J. Cowan, Joel D. Weingarten, and Daniel E. Koditschek. Visual servoing via navigation functions. *Transactions on Robotics and Automation*, 2002.

[6] Tarek Hamel and Robert Mahony. Visual servong of an under-actuated dynamic rigid-body system: An image-based approach. *IEEE Transactions on Robotics and Automation*, 18(2):187–198, April 2002.

[7] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, pages 651–670, October 1996.

[8] Daniel 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)*, pages 131–157. Amer. Math. Soc., Providence, RI, 1989.

[9] Daniel E. Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442, 1990.

[10] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[11] Chien-Ping Lue, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.

[12] Ezio Malis, Francois Chaumette, and Sylvie Boudet. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transactions on Robotics and Automation*, 18(2):176–186, April 2002.

[13] Jerrold E. Marsden and Tudor S. Ratiu. *Introduction to Mechanics and Symmetry*, chapter 14.7. Springer, 2 edition, 1999.

[14] Elon Rimon and Daniel E. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, Oct 1992.

[15] Camillo J. Taylor and James P. Ostrowski. Robust vision-based pose control. In *International Conference on Robotics and Automation*, volume 3, pages 2734–2740, San Francisco, CA, 2000. IEEE.

[16] Hong Zhang and James P. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *International Conf. on Robotics and Automation*, 1999.

[17] Hong Zhang and Jim Ostrowski. Visual motion planning for mobile robots. *IEEE Transactions on Robotics and Automation*, 18(2):199–208, April 2002.