Visual Servoing via Navigation Functions

Noah J. Cowan and Joel D. Weingarten and Daniel E. Koditschek

February 6, 2002

Abstract

This technical report[1] presents a framework for visual servoing that guarantees convergence to a visible goal from most initially visible configurations while maintaining full view of all the feature points along the way. The method applies to first and second order fully actuated plant models. The solution entails three components: a model for the "occlusion-free" workspace; a change of coordinates from image to model coordinates; and a navigation function for the model space. We present three example applications of the framework, along with experimental validation of its practical efficacy.

[1] A version of this manuscript will appear in IEEE Transactions on Robotics and Automation. This version contains many details which were omitted from the final paper.

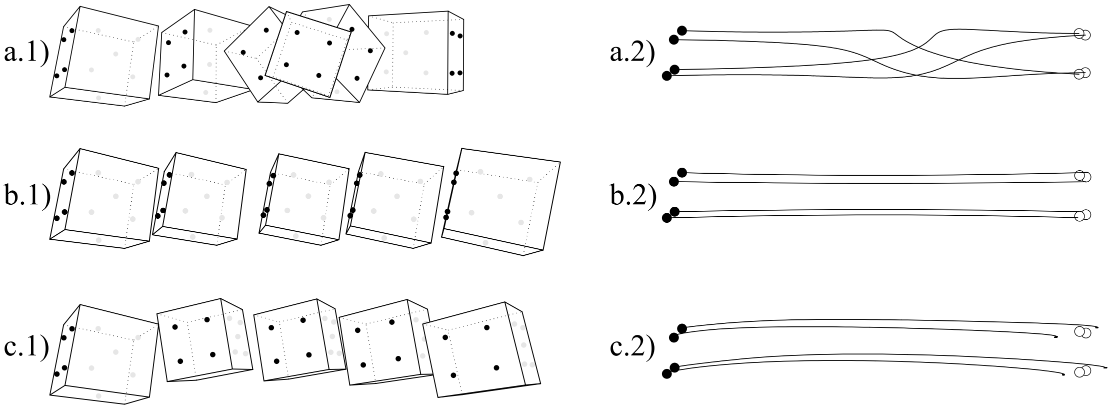

Figure 1: Simulation results for three different artificial potential function based visual servo controllers: **Top:** based upon the navigation function proposed in equation (34); **Middle:** based upon the potential function in equation (3) resulting in failure due to self occlusion; **Bottom:** based upon the potential function in equation (4) resulting in failure due to a spurious local minimum.

1 Introduction

Increasingly, engineers employ computer vision systems as sensors that measure the projection of features from a rigid body at as it moves in some scene. Such an approach presupposes the availability of a reliable machine vision system that supplies a controller with the image plane coordinates of features of the body. The machine vision system must incorporate image processing, feature extraction and correspondence algorithms suitable to the scene in view.[2] For vision-based control, these features are then used to close a servo loop around some desired visual image. Traditionally, visual servoing algorithms impose motion upon the actuated configuration space variables, $q \in \mathcal{Q}$, so as to align the image-plane features, $y \in \mathcal{Y}$, with a previously stored goal image, y^*. When the mapping, $c : \mathcal{Q} \to \mathcal{Y}$, from configuration variables to the image-plane is one-to-one in some vicinity of the goal then traditional visual servoing generally results in a closed loop system with an asymptotically stable equilibrium at the unique pre-image $q^* = c^{-1}(y^*)$. Many algorithms of this nature have been proposed and implemented with the result of large basins of attraction around the equilibrium in the presence of large errors in sensor and robot calibration. Hutchinson *et. al.* [15] provide a general introduction and extensive bibliography to this approach.

Simple visual position regulation, for which a suitable set of visible features contrast well their surrondings, leaves few challenges: high performance, dynamic (albeit local) vision-based controllers which fit neatly into the framework of linear control have existed now for some years [4]. This simple and complete characterization is owed entirely to the local nature of regulation problems. However, the creation of a richer set of behaviors whose *global* (and hence highly nonlinear) properties are well characterized remains a significant challenge to visual servoing. The perspective projection of features to a CCD camera does substantial violence to the scene being viewed. In addition to the loss of depth information of point features, the transient loss of features (due, for example, to occlusions) can cripple any control system which does not explicitly address the possibility. Moreover, there are subtle geometric facts that constrain our designs (whether or not we take them into account!). For

[2] Of course, as is clear from the extensive literature (*e.g.*[10, 12]), implementing these feature extracting “virtual sensors” represents a huge challenge and is beyond the scope of the present paper.

example, given the (monocular) image plane projection of a set of three rigidly connected points, it is well known that there exists up to four poses of a camera which will result in the same projection [35]. On the other hand, simply adding a fourth point over constrains the motion so that the individual feature point trajectories may not be assigned arbitrarily as they are algebraically constrained.

The classical approach to visual servoing attempts to impose straight-line trajectories on image feature point locations. For example, suppose the camera projects four feature point on a rigid body controlled by a 6 DOF robot. The configuration space of the robot – for example, its 6 joint angles – is given by $\mathcal{Q} \subset \mathbb{R}^6$, and image space is the four image plane pairs, namely $\mathcal{Y} \subset \mathbb{R}^8$. Locally, then, the camera map is then just a map from $\mathbb{R}^6$ to $\mathbb{R}^8$, namely

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_4 \end{bmatrix} = c(q), \quad \text{where} \quad y_i \in \mathbb{R}^2,\ i = 1, \ldots, 4 \tag{1}$$

are the image plane feature locations of the four features being observed. The traditional (kinematic) visual servoing law is then

$$\dot{q} = -J^{\dagger}(q)(y - y^*), \tag{2}$$

where $J^{\dagger} = (J^T J)^{-1} J^T$ is the pseudo inverse of the camera Jacobian matrix, $J(q) = Dc(q)$.

Traditional visual servoing systems, such as those based on the above approach, have numerous limitations. First, they result in a *local* basin of attraction whose extent is poorly or not at all characterized. For example, the incursion of spurious (attracting) critical points may arise when $y - y^*$ aligns with the the null space of $J^{\dagger}$ in (2). Consequently, the local basin of attraction around q^* may exclude seemingly reasonable initial conditions [3]. The second failing of the visual servoing algorithms proposed to date, involves their vulnerability to transient loss of features — either through self-occlusions or departure from the field of view (FOV). To the best of our knowledge, no prior work guarantees that these obstacles will be avoided. Usually, the problem of visibility obstacles is ignored in analysis, and when encountered in practice necessitates human intervention. However, as we will show, both of these limitations – local convergence and transient loss of features – can be overcome quite readily.

Another major problem is that most visual servoing algorithms do not specifically address dynamics. Of course, given the multitude of successful inverse dynamics based control strategies, trajectories generated from (2) (or any other kinematic controller) could be tracked very precisely with a high performance robot control system. However, such control techniques require precise parametric knowledge of the robot's kinematics and dynamics, the extra complexity of which seems superfluous given the simple end-point convergence objective of most visual servoing algorithms. In other words, the specific reference trajectory generated by kinematic controllers is merely a means to an end, and tracking that trajectory exactly may not be necessary. By contrast, our approach generates controllers capable of extremely high performance, which exhibit global convergence to the end-point goal, without the added complexity involved in precisely tracking a (clearly somewhat arbitrary) reference trajectory. Even though our algorithms do not prescribe specific reference trajectories, the methodology nevertheless affords certain guarantees on the trajectories that result. For example, features remain visible throughout the trajectory (even in the presence of Newtonian dynamics).

methodology also reduces to first-order settings, in which case the more traditional approach in the robotics literature of tracking reference trajectories applies.

1.1 Image-based navigation

In a naive attempt to improve (2), note that it may be conceived as a gradient law using the potential function

$$\widetilde{\varphi}(y) = \frac{1}{2}\sum_{i=1}^{4} \|y_i - y_i^*\|^2 \tag{3}$$

and letting $\dot{q} = -(J^T J)^{-1}\nabla_q(\widetilde{\varphi} \circ c)$. Suppose the four features are coplanar, e.g. they are all on the same face of a polyhedral body. A self-occlusion occurs when the plane containing the features intersects the camera pinhole – namely the 4 points project onto the same line on the image plane. To avoid this scenario, consider a naive improvement of (3) that avoids self-occlusion by "blowing them up", namely,

$$\widetilde{\varphi}(y) := \frac{\sum_{i=1}^{4} \|y_i - y_i^*\|^2}{\prod_{\{i,j,k\}\in\Gamma} \left|\det \begin{bmatrix} y_i & y_j & y_k \\ 1 & 1 & 1 \end{bmatrix}\right|^{1/2}}, \tag{4}$$

where $\Gamma = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}$. Because the features are coplanar, the denominator will go to zero as the features become collinear, and thus the gradient will point away from the self-occlusion obstacle. However, as can be seen from Figure 1, even though self-occlusions are avoided, convergence is not necessarily achieved. Other similarly naive approaches suffer the same peril.

Although the naive potential function approach above neither adequately addresses occlusions nor guarantees convergence, one suspects that some appropriately designed potential might overcome these serious limitations of traditional visual servoing. Indeed, we will show in this paper that the obstacles presented by self occlusion and a finite FOV can be obviated by addressing in a methodical fashion the relationships between the domain and range of c from which they arise.

Specifically, we introduce a framework for visual servoing, depicted in Figure 2, yielding feedback controllers which are

1. *dynamic:* applicable to second order (Lagrangian) as well as first order (kinematic) actuation models;

2. *global:* guaranteeing a basin of attraction encompassing most initial configurations that maintain feature visibility;

3. *visibility-obstacle free:* avoiding configurations that lose features due to either self occlusion or departure from the camera FOV.

1.2 Related work

A few researchers have incorporated Lagrangian dynamics into the visual servo loop. Zhang and Ostrowski [36] developed a controller for an Unpiloted Aerial Vehicle (UAV) equipped with a camera for which they exhibit a diffeomorphism between the centroid and radius of a circle in the image plane and the position and orientation of the UAV relative to a sphere in the workspace. Formulating the dynamics in generalized image-plane coordinates leads to a feedback linearized controller that accounts for the mechanical system dynamics. This novel contribution does not address the notion of "visibility" which we find central to visual servoing.

Many implementations in the literature offer strong anecdotal evidence that suggests convergence for visual servoing systems is robust with respect to large parametric uncertainty, though researchers rarely establish formally large basins of attraction of visual servoing systems (parametric uncertainty

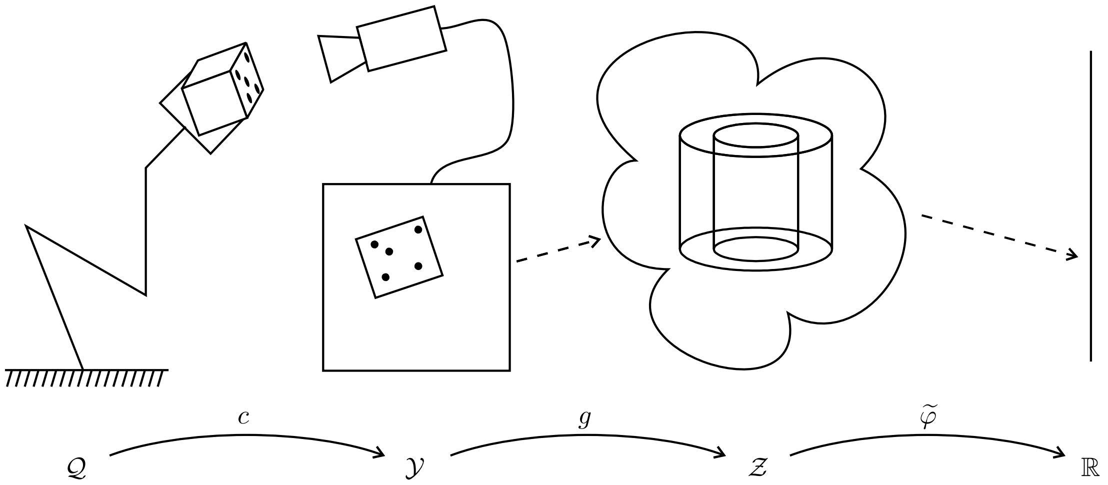

Figure 2: Our approach to visual servoing requires a camera map, $c : \mathcal{Q} \to \mathcal{Y}$, which takes each configuration $q \in \mathcal{Q}$ and produces an image plane measurement $y = c(q) \in \mathcal{Y}$, as described in Section 2. The image plane is then deformed into a model of the "safe" unoccluded scene $\mathcal{Z}$, through a change of coordinates g, as described in Section 4. Finally, a navigation function, $\widetilde{\varphi} : \mathcal{Z} \to \mathbb{R}$, which encodes the goal as the unique global minimum, is used to construct a convergent generalized nonlinear PD controller, as reviewed in Section 3.

aside). Malis *et. al.* [22] and Taylor and Ostrowski [32] introduce visual servo controllers incorporating partial pose reconstruction to guarantee convergence even in the presence of large parametric uncertainty. However, these contributions do not address second order dynamics nor guarantee that transients will avoid visibility obstacles. Rizzi *et. al.* designed a globally convergent nonlinear dynamical observer for a falling point mass viewed by a binocular[3] camera pair [29]. Their observer exploits the property that a pinhole camera projects lines into lines to map the observer feedback into the image plane. This "pinhole property" can also be applied to binocular quasi-static visual servoing [16] to yield a controller with a large basin of attraction. Hashimoto, *et. al.* [13] use potential functions to increase the domain of attraction for their (quasi-static) visual servoing system.

In addition to our preliminary results [6, 7, 8], there have been some recent efforts to address the FOV problem, albeit in a quasi-static setting. Malis *et. al.* [22] guarantee that a single base point remains within the FOV while, as noted above, guaranteeing convergence for a large basin of attraction. Corke and Hutchinson [5] make no formal guarantees, but have designed a clever "partitioned" visual servo strategy for which simulations suggest a large basin of attraction while maintaining all features within the FOV boundary. Both of these contributions are in a quasistatic setting.

[3] This paper makes the distinction between stereo, wherein a camera pair has nearly the same perspective of a scene (as is the case for biological systems with stereo vision), and binocular vision in general.

1.3 Visual sensing of rigid motion

If the features are such that the camera map c is injective, each measurement y "pins down" a unique location q. Hence a positioning task may be accomplished with image measurements, even in the presence of parametric uncertainty in the robot and sensor. Injectivity of c implies that the dimension of the output space is greater than or equal to the dimension of the configuration space, but, of course, the converse is not true. For example, the perspective projection of three feature points of a rigid body is a 6-dimensional measurement, from the three image plane (x, y) pairs, but each image 6-tuple generally corresponds to multiple algebraic solutions for the position of the rigid body (for a nice geometric interpretation, see Wolfe *et. al.* [35]). However, the perspective projection of at least four rigidly constrained points in general position on a body uniquely determines its pose (see for example [27]) and hence there exists a unique pose which registers with each such projection.

"Natural" coordinates for visual servoing have not yet emerged in any generality. In the example applications to follow, we identify in each case the underlying model space, construct a solution for the model, and furnish a change of coordinates that pulls the model solution back into terms of the physically measured features. However, we are not yet able to characterize in any physically convenient manner the general properties of the features that would make such constructions possible. The UAV work described above [36] represents a different attempt to develop natural features for visual servoing. Another seemingly promising alternative relies on projective kinematics [31], although it has not been validated empirically and currently lacks the machinery required to lift it to the second order setting.

1.4 Organization

The central contribution of the paper is found in Section 4 where we propose a novel framework for dynamic, occlusion-free global visual servoing. We show how to apply this framework using three illustrative examples that provide insight into the specific geometries of some prototypical visual servoing systems. In Section 5 we present our empirical results for two of the example setups. Finally we provide some concluding remarks in Section 6.

Before we can proceed, however, we must introduce our sensor in Section 2, and then review concepts from robot control in Section 3 to provide a theoretical foundation for the subsequent material.

2 Sensor Model

To sense a robot's configuration with a camera, we seek a map, c, from the robot configuration space $\mathcal{Q}$ to an appropriate output space $\mathcal{Y}$, which depends on several factors. We assume that a camera may be modeled as map, π, from three space to the image plane. The feature space, $\mathcal{FS}$, varies from problem to problem, but often we just choose point features in Euclidean space, $\mathbb{E}^3$, in which case $\mathcal{FS} = \mathbb{E}^3 \times \cdots \times \mathbb{E}^3$ (N times). The features, $p \in \mathcal{FS}$, are rigidly attached to the end effector of the robot. Their position in a body fixed frame, $\mathbf{F}_b$, is known and hence their location in a camera fixed frame, $\mathbf{F}_c$, is given by the kinematics, $h : \mathcal{Q} \times \mathcal{FS} \to \mathcal{FS}$.

Suppose $P = \{p_i\}_{i=1}^N$ is our feature set. Then ${}^c p_i = h(q, {}^b p_i)$ where ${}^b p_i$ is the location of the feature point in the body fixed frame (assuming we are dealing with point features) and ${}^c p_i$ is the same point in the camera fixed coordinate frame. The composition of the camera model with the kinematics as applied to all of the features generates the camera map $c_P : \mathcal{Q} \to \mathcal{Y}$

$$c_P(q) := \begin{bmatrix} \pi \circ h(q, p_1) & \cdots & \pi \circ h(q, p_N) \end{bmatrix} \tag{5}$$

parameterized by the (constant) feature locations in body coordinates.[4]

2.1 Camera models

A well documented family of models mapping elements of projective space, $\mathbb{P}^3$, to the projective plane, $\mathbb{P}^2$, admit of a matrix representation [23]: orthographic projection; weak perspective projection; affine projection; and full perspective projection — in order of increasing accuracy. Because we are interested in potentially large displacements over the camera field of view (FOV), we employ a full perspective camera model in the present work. Of course, more complex models do exist for camera projection, *e.g.* Tsai [33] presents a technique for calibrating a camera to account for radial lens distortion. However, the physical camera systems used for experiments in the present work are well modeled without accounting for additional complexities.

Assume that there is a camera fixed coordinate frame, $\mathbf{F}_c$, and that points are expressed with respect to that frame and that the (x, y)-plane of $\mathbf{F}_c$ is parallel to the image plane and coincident with the optical center or "pinhole." Projective points, $p \in \mathbb{P}^n$, are represented in homogeneous coordinates as $n+1$ vectors where not all elements are zero, and Euclidean points $x \in \mathbb{E}^n$ are represented in homogeneous coordinates as $n+1$ vectors $[\ x^T,\ 1\ ]^T$. Note that while $\mathbb{E}^n$ is a subset of $\mathbb{P}^n$ ([2], Chapter 3), the use of the homogeneous matrix representation plays a very different role in the two cases. Nevertheless, in an abuse of notation, we will use the homogeneous representation for both situations when the specific space of interest is clear from the context.

A simple pinhole camera with unit focal length, $\lambda = 1$, admits a matrix representation relative to the homogeneous matrix representation of $\mathbb{P}^n$ as $\Pi : \mathbb{P}^3 \to \mathbb{P}^2$, where

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6}$$

The restriction of the camera map to $\mathbb{E}^n \subset \mathbb{P}^n$ does not admit of a matrix representation when using the homogeneous matrix representation for points in $\mathbb{E}^n$. Rather, we must divide the last entry to normalize so that the pinole camera map from $\mathbb{E}^3$ to $\mathbb{E}^2$ is given by $p \mapsto \gamma(\Pi p)$, where

$$\gamma(p) = \frac{1}{p_3} p = \begin{bmatrix} \frac{p_1}{p_3} & \frac{p_2}{p_3} & 1 \end{bmatrix}^T. \tag{7}$$

To account for focal length λ, pixel scale α_x, α_y, offset s_x, s_y and a skew factor on the image plane θ (which is usually $\pi/2$), we introduce the projective transformation $A : \mathbb{P}^2 \to \mathbb{P}^2$

$$A := \begin{bmatrix} -\overline{\alpha}_x & \overline{\alpha}_x \cos\theta & s_x \\ 0 & -\overline{\alpha}_y / \sin\theta & s_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{array}{l} \overline{\alpha}_x = \lambda\, \alpha_x \\ \overline{\alpha}_y = \lambda\, \alpha_y \end{array} \tag{8}$$

The 5 parameter matrix A is the so-called *intrinsic parameter* matrix. A complete model for a perspective projection camera, $C : \mathbb{P}^3 \to \mathbb{P}^2$ is

$$C := A\,\Pi = \begin{bmatrix} A & \mathbf{0} \end{bmatrix} \tag{9}$$

or, in Euclidean coordinates, $\pi : \mathbb{E}^3 \to \mathbb{E}^2$

$$\pi(p) = \gamma(A\,\Pi\, p). \tag{10}$$

[4] To simplify the analysis, we assume that the measurements are temporally and spatially continuous, the validity of which depends heavily on the problem. For the implementations discussed in Section 5 we believe they are accurate assumptions.

In the case that the camera frame $\mathbf{F}_c$, is not coincident with a world reference frame $\mathbf{F}_w$, in which the points are expressed, then we have $C := A\,\Pi\,{}^c\!H_w$, where the 6 parameter *extrinsic parameter* matrix ${}^c\!H_w$ is the transformation from the world frame coordinates to camera frame coordinates.

The camera map (5) depends on several factors: the type, number and configuration of features selected; the robot kinematics; the camera model. Section 4.1 considers a planar problem in which the configuration space is SE(2), and the camera map is defined to be the perspective projection of three feature points on the body onto a 1D camera, hence $\mathcal{FS} = \mathbb{E}^2 \times \mathbb{E}^2 \times \mathbb{E}^2$ and $\mathcal{Y} = \mathbb{E}^1 \times \mathbb{E}^1 \times \mathbb{E}^1$. Section 4.2 presents an architecture in which a 3DOF robot moves in a subset of T^3, and the camera map is the projection of the position and orientation of a feature on the end effector, $\mathcal{FS} = T_1\mathbb{E}^3$, so that $\mathcal{Y} \subseteq T_1\mathbb{E}^2 \approx \mathbb{E}^2 \times \mathrm{S}^1$. Finally, in Section 4.3, we present a 6DOF visual servoing algorithm which uses the projection of a set of coplanar points. In this example we compute the inverse of the forward camera map and work in a reconstruction of the 3D space for visual servoing, while still guaranteeing dynamical convergence and occlusion obstacle avoidance.

3 Robot Control via Navigation Functions

Assume we have a holonomically constrained, fully actuated robot with known kinematics, affording a suitable set of local coordinates, $q_i, i = 1, \ldots, n$, and denote the $n-$dimensional free configuration space as $\mathcal{Q}$. The system dynamics

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + F_{ext}(q, \dot{q}),$$

may be found using Lagrange's equations (see, for example, [1, 9, 11, 25]) where F_{ext} are external forces (such as friction) which do not arise from Hamilton's variational principle and τ are the input torques. The camera plays the role of output map, $c : \mathcal{Q} \rightarrow \mathcal{Y}$. We assume exact knowledge of the kinematics and link masses[5] (hence, the gravitational term G is exactly known) as well as the external forces F_{ext}. Letting the input torque be $\tau = u - F_{ext} + G(q)$, where u is our control input, the plant equations are, in state-space form,

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} x_2 \\ M(x_1)^{-1}\left(u - C(x_1, x_2)x_2\right) \end{bmatrix}, \\ y &= c(x_1), \end{aligned} \tag{11}$$

where $x = [q^T, \dot{q}^T]^T$.

3.1 Task specification

The state space is constrained by the presence of forbidden configurations, the *obstacle set* $\mathcal{O} \subset \mathcal{Q}$. The *free space* is defined as the obstacle-free configuration space $\mathcal{V} = \mathcal{Q} - \mathcal{O}$, and *safe configurations* $\mathcal{D} \subseteq \overline{\mathcal{V}}$ form a compact connected differentiable manifold with boundary. The positioning objective is described in terms of a *goal set* $\mathcal{G} \subset \overset{\circ}{\mathcal{D}}$. The task is to drive q to $\mathcal{G}$ asymptotically subject to (11) by an appropriate choice of u while avoiding obstacles. We restrict our attention to point attractors, $\mathcal{G} = \{q^*\}$. Moreover, the basin of attraction $\mathcal{E} \subset T\mathcal{D}$ must include a dense subset of the zero velocity section of $T\mathcal{D}$, so that we may guarantee convergence from the entire configuration space. Obstacle avoidance requires that the trajectories avoid crossing the *boundary set* $\mathcal{B} = \partial\mathcal{D}$, *i.e.* $q(t) \in \mathcal{D}$, for all $t \geq 0$.

[5] Generally these data are supplied by a manufacturer, while the link inertias tensors remain unreported and are generally difficult to measure exactly. Notice that our "generalized PD" approach to control, below, will not require the precise form of the mass-inertia matrix M or the Coriolis term C that would necessitate knowledge of these parameters.

3.2 Navigation functions

The task of moving to a goal while avoiding obstacles along the way can be achieved via a nonlinear generalization of proportional-derivative (PD) control deriving from Lord Kelvin's century old observation that total energy always decreases in damped mechanical systems [19]. Formally, this entails the introduction of a gradient vector field from a "navigation function," a refined notion of an artificial potential function, together with damping to flush out unwanted kinetic energy.[6]

3.2.1 Gradient Vector Fields

Let $\mathcal{D}$ be a compact $n-$dimensional Riemannian manifold with boundary, with metric $\langle \cdot\, , \cdot \rangle$, and $\varphi : \mathcal{D} \to \mathbb{R}$ be a C^2 functional. For Lagrangian control systems of the kind introduced above, the Riemannian metric is specified by the kinetic energy tensor. When the mechanical system is so heavily damped that kinetic energy is negligible, or is explicitly controlled in "velocity mode" then the plant equations may be considered first order, and any Riemannian metric may be used. In these situations, for example as in the case of our RTX robot discussed in 5.3, the convergence properties of the gradient field alone yield the desired result.

Choose a local coordinate chart $\psi : U \subset \mathbb{R}^n \to \mathcal{D}$. The gradient in local coordinates is given by the vector of partial derivatives weighted by the inverse of the metric. Suppose $q \in \mathbb{R}^n$ are the local coordinates induced by ψ. In an abuse of notation, we write $\varphi \circ \psi(q) = \varphi(q)$, and hence, in local coordinates

$$\nabla \varphi(q) = M^{-1}(q)\, D_q^T \varphi(q)$$

where $(D_q \varphi)_i = \frac{\partial \varphi}{\partial q_i}$, and M is the local representation of the Riemannian metric. Gradient descent can now be achieved in local coordinates via

$$\dot{q} = -M^{-1}(q)\, D_q^T \varphi(q) \tag{12}$$

A smooth scalar valued function whose Hessian matrix is non-singular at every critical point is called a *Morse* function [14]. Artificial potential controllers arising from Morse functions impose global steady state properties that are particularly easy to characterize, as summarized in the following proposition.

Proposition 1. *(Koditschek, [19]) Let φ be a twice continuously differentiable Morse function on a compact Riemannian manifold, $\mathcal{D}$. Suppose that $\nabla\varphi$ is transverse and directed away from the interior of $\mathcal{D}$ on any boundary of that set. Then the negative gradient flow has the following properties:*

1. *$\mathcal{D}$ is a positive invariant set;*
2. *the positive limit set of $\mathcal{D}$ consists of the critical points of φ*
3. *there is a dense open set $\widetilde{\mathcal{D}} \subset \mathcal{D}$ whose limit set consists of the local minima of φ.*

For the quasi-static mechanical systems, this observation is sufficient to guide controller design. However, these first order dynamical convergence results do not apply when a Lagrangian system is subject to such a potential field. We now briefly review machinery to "lift" the gradient vector field controller to one appropriate for second order plants of the kind introduced in (11).

[6] A good survey of potential field methods for robot navigation is given by ([21], Chapter 7). The refinement to *navigation functions*, first articulated by Koditschek and Rimon [19, 20, 28], is only very briefly sketched here.

3.2.2 Second order, damped gradient systems

Introducing a linear damping term, yields a simple "PD" style feedback, in local coordinates,

$$u = -D_q\varphi(q)^T - K_d\,\dot{q}, \tag{13}$$

that is appropriate for second order plants. Lord Kelvin's observation is now relevant and it follows that the total energy,

$$\eta = \varphi + \kappa \quad \text{where} \quad \kappa = \tfrac{1}{2}\dot{q}^T M(q)\dot{q}, \tag{14}$$

is non-increasing.

Unfortunately, if the total initial energy is higher than the energy at some point on the boundary $\partial\mathcal{D}$, trajectories may intersect the boundary. Fortunately, further refining the class of potential functions will enable us to construct controllers for which the basin of attraction contains a dense subset of the zero velocity section of $\mathcal{D}$. The following definition has been adapted from [19].

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

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

This notion, together with Lord Kelvin's observation, now yield the desired convergence result for the Lagrangian system (11).

Proposition 2. *(Koditschek [19]) Given the system described by (11) subject to the control (13), almost every initial condition q_0 within the set*

$$\mathcal{E} = \{(q, \dot{q}) \in T\mathcal{D} : \eta(q, \dot{q}) \leq 1\} \tag{15}$$

converges to q^* asymptotically. Furthermore, transients remain within $\mathcal{D}$ such that $q(t) \in \mathcal{D}$ for all $t \geq 0$.

Proposition 2 generalizes the kinematic global convergence of Proposition 1. Note that for the second order system $\mathcal{E}$ imposes a "speed limit" as well as a positional limit, since the total energy must be initially bounded.

3.2.3 Invariance under diffeomorphism

One last key ingredient in the mix of geometry and dynamics underlying the results we present revolves around the realization that a navigation function in one coordinate system is a navigation function in another coordinate system, if the two coordinate systems are diffeomorphic [19]. This affords the introduction of geometrically simple model spaces and their correspondingly simple model navigation functions.

4 Navigation Function Based Visual Servoing

We wish to create visual servoing algorithms that are high performance, global and yet safe with respect to occlusion obstacles $\mathcal{O}$ that arise due to the finite FOV and self-occlusions. To achieve our objective we compute the *visible set* for a particular problem. This is the set of all configurations $\mathcal{V} := \mathcal{Q} - \mathcal{O}$ in which all features are visible to the camera and on which c is well defined. We then design a safe, possibly conservative, subset $\mathcal{D} \subseteq \mathcal{V}$ in which the body is permitted to move. The set $\mathcal{D}$ provides additional safety with respect to obstacles and possibly simplifies the topology. We then define the *image space* $\mathcal{I} = c(\mathcal{D}) \subset \mathcal{Y}$. The camera map must be a diffeomorphism $c : \mathcal{D} \approx \mathcal{I}$. For each problem, $\mathcal{D}$ is analyzed to construct a model space $\mathcal{Z}$ and a diffeomorphism $g : \mathcal{I} \approx \mathcal{Z}$. For simplicity, we restrict our attention to point attractors $\mathcal{G} = \{q^*\}$, $q^* \in \overset{\circ}{\mathcal{D}}$ from which we define the *goal image* $y^* = c(q^*)$.

We propose a new framework for visual servoing that incorporates three ingredients:

1. a *model space*, $\mathcal{Z}$, for the "safe" configurations, $\mathcal{D}$;

2. a *navigation function* $\widetilde{\varphi} : \mathcal{Z} \to [0, 1]$, for the model space;

3. a *diffeomorphism*, $g : \mathcal{I} \to \mathcal{Z}$, from the image space to the model space.

These three ingredients are assembled with the feedback control strategy (13), which guarantees that *all* initial configurations within $\mathcal{D}$ *dynamically converge* to the goal while ensuring *occlusion-free* transients with respect to the obstacle set $\mathcal{O}$ and hence overcome all of the traditional limitations of visual servoing systems listed in Section **??**.

In 2D NF-based visual servoing, the function g does not contain a copy of the inverse of the camera map. In 3D NF-based visual servoing, however, g does contain a copy of the inverse camera map, c^{-1}. In that sense, 2D algorithms are "simpler." On the other hand, 2D algorithms require the Jacobian of the camera map c, because the navigation function in the configuration variables is given by $\varphi(q) = \widetilde{\varphi} \circ g \circ c(q)$ and hence the gradient is given by

$$D_q \varphi^T = Dc^T \, Dg^T \, D\widetilde{\varphi}^T.$$

By recourse to the general framework outlined above we develop controllers for several specific configurations of a robot and monocular camera in the subsections that follow. In Section 4.1 and Section 4.2 we present two example systems which incorporate 2D visual servoing, and in Section 4.3 we present a 3D visual servoing algorithm for a 6DOF body. Interestingly, the very different visual servoing problems of Section 4.2 and Section 4.3 share a common model space $\mathcal{Z} = [-1, 1]^n \times \mathrm{S}^1$ for $n = 2$ and $n = 5$ respectively, so Appendix A presents a new navigation function for the more general model space $\mathcal{Z} = [-1, 1]^n \times \mathrm{T}^m$ for all n and m in $\mathbb{N}$.

4.1 Example 1: Planar body, planar camera

In this section, we discuss the problem of visually servoing a planar rigid body, viewed by a planar pinhole camera, as depicted in Figure 3, first presented in [6].

Suppose we have a planar rigid body, $\mathcal{Q} = \mathrm{SE}(2)$, with three distinguishable collinear feature points. In this case, we will adopt a representation via homogeneous matrices parameterized in local coordinates by a position and orientation $q = [\mathbf{T}_x, \mathbf{T}_y, \theta]^T$, *i.e.*

$$H = \begin{bmatrix} R & \mathbf{T} \\ 0^T & 1 \end{bmatrix} = \psi(q) := \begin{bmatrix} \cos\theta & -\sin\theta & \mathbf{T}_x \\ \sin\theta & \cos\theta & \mathbf{T}_y \\ 0 & 0 & 1 \end{bmatrix}$$

and interpret $H = {}^C H_B \in \mathrm{SE}(2)$ as the rigid change of coordinates from the body to the camera frame. We conveniently co-locate the x-axis of the body with the edge containing the feature points, P, so that in body coordinates

$${}^b p_i = \begin{bmatrix} l_i & 0 & 1 \end{bmatrix}^T \quad \text{for } i = 1, 2, 3$$

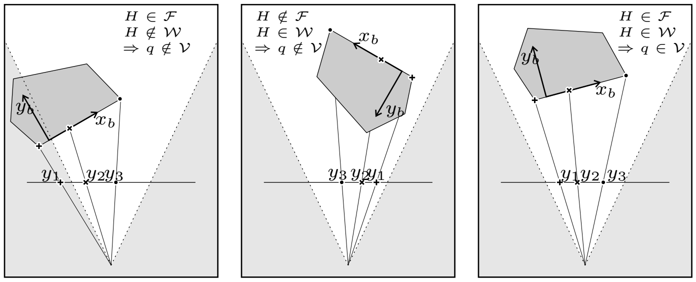

Figure 3: The setup for a planar pinhole camera and planar rigid body with collinear feature points showing three typical configurations of a rigid body with respect to the camera. **Left:** The features face the camera, but the leftmost point lies out of view. **Center:** Although within the camera workspace, the body occludes the features. **Right:** The features are all visible.

where $l_1 < l_2 < l_3$, and in camera coordinates

$$ {}^c p_i = H\,{}^b p_i = \begin{bmatrix} \mathbf{T}_x + l_i \cos\theta & \mathbf{T}_y + l_i \sin\theta & 1 \end{bmatrix}^T $$

The body y-axis is oriented "into" the body, as depicted in Figure 3.

The camera is assumed to have a finite FOV, represented by two angles $\alpha_1, \alpha_2 \in (-\pi/2, \pi/2)$ measured with respect to the camera optical axis (y-axis). The FOV angles define a FOV cone which is the workspace of the camera, $\mathcal{W}_c \subset \mathbb{E}^2$

$$ \mathcal{W}_c = \left\{ p \in \mathbb{E}^2 : p_2 > 0, \arctan(p_1/p_2) \in (\alpha_1, \alpha_2) \right\}. $$

For simplicity, consider a pinhole camera (10) with unit focal length $\pi : \mathcal{W}_c \to \mathbb{E}^1$ is simply

$$ \pi(p) := \begin{bmatrix} \frac{p_1}{p_2} & 1 \end{bmatrix}^T \qquad (16) $$

where p is a point expressed with respect to the camera frame. The "edge" of the image plane corresponds to the two points $y^{\min} = \tan\alpha_1$ and $y^{\max} = \tan\alpha_2$.

4.1.1 The camera map and visible set for planar servoing

The total workspace $\mathcal{W} \subset \mathrm{SE}(2)$ is

$$ \mathcal{W} := \left\{ H \in \mathrm{SE}(2) : H\,{}^b p_i \in \mathcal{W}_c, i = 1 \ldots 3 \right\}. $$

In other words, to be in the workspace, all features must project to the interval $\pi(\mathcal{W}_c) = (y^{\min}, y^{\max}) \subset \mathbb{E}^1$, as depicted in Figure 3.

The set of configurations "facing" the camera is simply

$$ \mathcal{F} := \{ H \in \mathrm{SE}(2) : \mathbf{v}(H) > 0 \} \quad \text{where} $$

$$ \mathbf{v}(H) = (\mathbf{T}_y \cos\theta - \mathbf{T}_x \sin\theta), $$

so the visible set is defined $\mathcal{V} := \mathcal{F} \cap \mathcal{W}$.

We now define the camera map, $c : \mathcal{V} \to \mathbb{E}^1 \times \mathbb{E}^1 \times \mathbb{E}^1$

$$ c(H) := \begin{bmatrix} \pi(H\,{}^b p_1) & \pi(H\,{}^b p_2) & \pi(H\,{}^b p_3) \end{bmatrix}. \qquad (17) $$

In an abuse of notation, we will often write $c(q)$ when using local coordinates, as a stand in for $c \circ \psi(q)$.

The points $y = [y_1, y_2, y_3] = c(H)$ are related to the feature positions along the face of the body by a "homography" (a member of the group of projective transformations of the line, PL(1) [10]), *i.e.*

$$\alpha_i y_i = Z \begin{bmatrix} l_i & 1 \end{bmatrix}^T, \quad i = 1, 2, 3, \quad \text{where} \qquad (18)$$
$$Z = \begin{bmatrix} \cos\theta & \mathbf{T}_x \\ \sin\theta & \mathbf{T}_y \end{bmatrix},$$

4.1.2 Diffeomorphism to the image plane for planar servoing

We identify the output space $\mathcal{Y} = \mathbb{E}^1 \times \mathbb{E}^1 \times \mathbb{E}^1$ with $\mathbb{E}^3$ in the natural way. Define

$$\mathcal{I}_\rho = \Big\{ \; y \in \mathbb{E}^3 : y_1 - y^{\min} \geq \rho_0, \; y_2 - y_1 \geq \rho_1, \qquad y_3 - y_2 \geq \rho_2, \; y^{\max} - y_3 \geq \rho_3 \; \Big\}$$

where $\rho_i \geq 0$, $i = 0, \ldots, n$. When $\rho = [\rho_0, \ldots, \rho_3]^T = 0$, we write $\mathcal{I}_0$, and define

$$\mathcal{J} := \overset{\circ}{\mathcal{I}_0} = \left\{ y \in \mathbb{E}^3 : y^{\min} < y_1 < y_2 < y_3 < y^{\max} \right\}.$$

The map c (17) is a diffeomorphism from $\mathcal{V}$ to its image $c(\mathcal{V}) = \mathcal{J}$, the proof of which follows from the following three facts: (i) c is onto, *i.e.* $c(\mathcal{V}) = \mathcal{J}$, (ii) c is one-to-one and (iii) c is a local diffeomorphism – *i.e.* it is smooth and smoothly invertible – at each point in $\mathcal{V}$.

(i) One readily verifies that for each $H \in \mathcal{V}$, $y = c(H) \in \mathcal{J}$, *i.e.* $c(\mathcal{V}) \subseteq \mathcal{J}$. Furthermore, for each three vector $y \in \mathcal{J}$, there is a unique (up to scale) nondegenerate homography, Z, which relates the correspondences via[7]

$$\alpha_i y_i = Z \begin{bmatrix} l_i & 1 \end{bmatrix}^T, \quad i = 1, 2, 3$$

in homogeneous coordinates. Since Z is full rank, the columns are linearly independent and hence Z may be parameterized as in (18) for $H \in \mathcal{V}$ (since if $H \notin \mathcal{V}$ then $y \notin \mathcal{J}$).

(ii) Suppose that c is not one-to-one, *i.e.* $y = c(H_1) = c(H_2)$, $H_1, H_2 \in \mathcal{V}$, then the matrices

$$Z_1 = \begin{bmatrix} \cos\theta_1 & \mathbf{T}_1 \\ \sin\theta_1 & \end{bmatrix} \quad \text{and} \quad Z_2 = \begin{bmatrix} \cos\theta_2 & \mathbf{T}_2 \\ \sin\theta_2 & \end{bmatrix}$$

represent the same homography, namely $Z_1 = \alpha Z_2$. Since the first column of each matrix has unit length, then $Z_1 = \pm Z_2$, but if both $\mathbf{T}_1$ and $\mathbf{T}_2$ are in front of the camera, then $(\mathbf{T}_i)_y > 0$, and hence $H_1 = H_2$ which is a contradiction.

(iii) The pinhole camera π is differentiable everywhere in front of the camera, and hence c is differentiable on $\mathcal{V}$. Direct computation reveals that

$$|Dc(q)| = \frac{(l_1 - l_2)(l_2 - l_3)(l_3 - l_1)(\mathbf{T}_y \cos\theta - \mathbf{T}_x \sin\theta)}{(\mathbf{T}_y + l_1 \sin\theta)^2 (\mathbf{T}_y + l_2 \sin\theta)^2 (\mathbf{T}_y + l_3 \sin\theta)^2}$$
$$= \mathbf{v}(H) \frac{(l_1 - l_2)(l_2 - l_3)(l_3 - l_1)}{(\mathbf{T}_y + l_1 \sin\theta)^2 (\mathbf{T}_y + l_2 \sin\theta)^2 (\mathbf{T}_y + l_3 \sin\theta)^2}$$

which is different from zero at every point in $\mathcal{V}$, and hence c is a local diffeomorphism at every point in $\mathcal{V}$ (inverse function theorem).

[7] A projective transformation, $Z \in$ PL(1) is uniquely determined by the correspondence of three distinct points [26].

4.1.3 Model space for planar servoing

We seek a compact manifold with boundary on which to impose a navigation function. Note that $\mathcal{V}$ is an open set. Moreover, $\overline{\mathcal{V}}$ is not compact since there is no bound on the magnitude of translational component, and hence it is impossible that $\overline{\mathcal{V}} \approx \mathcal{I}_0$. However it is practical to impose impose a "collar" around the points on the image plane by enforcing $\rho_i > 0$ and hence require that they maintain a minimum distance from one another, as well as maintaining their distance from the edge of the image plane. Note that $\mathcal{I}_\rho \subset \mathcal{J}$ if $\rho_i > 0$, and hence

$$\mathcal{D} = c^{-1}(\mathcal{I}_\rho) \subset \mathcal{V} \tag{19}$$

is compact (Munkres [24], Theorem 5.5).

So, we identify a general model space of the form

$$\mathcal{Z} = \{z \in \mathbb{R}^n : z_{i+1} - z_i \geq \rho_i,\ i = 0, \ldots, n\} \tag{20}$$

where z_0, z_{n+1} and $\rho_i > 0$, $i = 0, \ldots, n$ are suitable constants. For planar visual servoing, $n = 3$.

4.1.4 Navigation function for planar visual servoing

We require a change of coordinates from the image plane to a model space $\mathcal{Z}$ for $\mathcal{D}$. In our case, we choose $\mathcal{Z} = \mathcal{I}_\rho$, and hence g is the identity mapping. For the model space (20) we refine a class of navigation functions designed in a separate context by Koditschek [18].

Proposition 3. *(Adapted directly [18]) The objective function*

$$\overline{\varphi}(z) = \frac{\|z - z^*\|^{2k}}{\prod_{i=0}^{n}(z_{i+1} - z_i)^2 - \rho_i^2}$$

is convex on

$$\mathcal{Z} = \{z \in \mathbb{R}^n : z_{i+1} - z_i \geq \rho_i,\ i = 0, \ldots, n\}$$

where z_0, z_{n+1} and ρ_i, $i = 0, \ldots, n$ are suitable constants and $k > (2n+3)/2$ in which case

$$\widetilde{\varphi} := \frac{\overline{\varphi}^{1/k}}{(1 + \overline{\varphi})^{1/k}} \tag{21}$$

is a navigation function on $\mathcal{Z}$.

For planar servoing, $n = 3$ and so we require $k > 9/2$, hence

$$\overline{\varphi}(y) = \|y - y^*\|^{2k} / (\left[(y_1 - y^{\min})^2 - \rho_0^2\right] \left[(y_2 - y_1)^2 - \rho_1^2\right] \\ \left[(y_3 - y_2)^2 - \rho_2^2\right] \left[(y^{\max} - y_3)^2 - \rho_3^2\right])$$

is convex. Moreover, $\widetilde{\varphi}$ given by (21) is a navigation function on $\mathcal{I}_\rho$ and $\varphi := \widetilde{\varphi} \circ c$ is a navigation function on $\mathcal{D}$.

4.2 Example 2: Buehgler arm, spatial camera

This example serves two purposes. From a practical standpoint, it has allowed us to explore visual servoing with a high performance experimental apparatus called the Buehgler arm built previously in our laboratory [34, 29] which we augmented with two high-speed digital cameras. The experimental results are shown in Section 5. From a theoretical standpoint, it will allow us to explore servoing on a different space $\mathcal{Z} = [-1, 1]^n \times \mathrm{T}^m$ with $n = 2$ and $m = 1$, with a high performance platform that demands a "tunable" navigation function. Moreover, it will turn out that 6DOF visual servoing (Section 4.3) can be cast in the same model space (with $n = 5$ and $m = 1$).

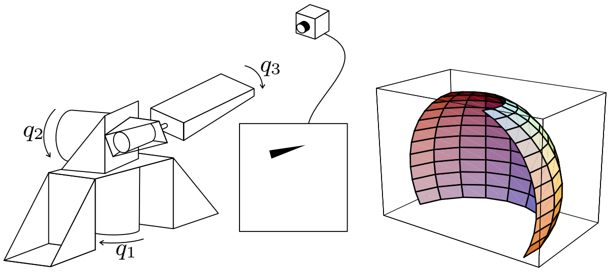

Figure 4: **Left:** The Buehgler arm has been modified with an "arrow" feature on the tip of the arm, which is observed by a perspective projection camera. The camera image is segmented to extract the arrow, as depicted. **Right:** The surface swept out by the tip of the arm for $q_1 \in (-\pi/2, \pi/2)$, $q_2 \in (-2\pi/3, -0.1\pi)$.

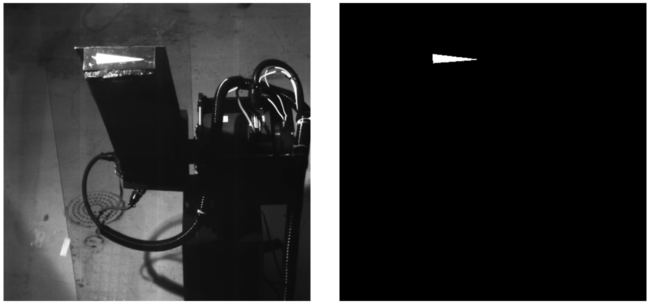

Figure 5: **Left:** An image of the Buehgler arm taken from the servo camera. The tip of the arm has been marked with a "pointer." **Right** The segmented image showing the extracted feature.

4.2.1 Buehgler arm kinematics

The Buehgler arm, depicted in Figure 4, has three actuated revolute degrees of freedom parameterized in local coordinates by angles $q = [\ q_1,\ q_2,\ q_3\ ]^T$ and its configuration space is $\mathcal{Q} = \mathrm{S}^1 \times \mathrm{S}^1 \times \mathrm{S}^1 = \mathrm{T}^3$. Denote the homogeneous transformation from the gripper frame to the world base frame, given in [30], as ${}^{w}H_b(q)$. The transformation from the world frame to camera frame is ${}^{c}H_w$, and hence $H = {}^{c}H_b = {}^{c}H_w {}^{w}H_b$. We let $R = [r_1, r_2, r_3]$ and $\mathbf{T}$ denote the rotation and translation, respectively, effected by H.

We affix a "pointer" to the tip of the arm, which we model as the position and unit orientation of a vector in three space, namely[8] $({}^{b}r, {}^{b}v) \in T_1\mathbb{E}^3 \approx \mathbb{E}^3 \times \mathrm{S}^2 = \mathcal{FS}$. The total forward kinematic map is then defined $h_\delta : \mathcal{Q} \times \mathcal{FS} \to \mathcal{FS}$

$$h_\delta(q) := \begin{bmatrix} H(q){}^{b}r \\ R(q){}^{b}v \end{bmatrix} =: \begin{bmatrix} r_\delta(q) \\ v_\delta(q) \end{bmatrix}$$

where (r_δ, v_δ) are in the camera frame, and the subscript denotes dependence on an important kinematic parameter, the "shoulder" offset, δ.

We consider a feature point at the tip of the arm (which has length ϱ), oriented in the y-direction of the gripper frame, namely

$$({}^{b}r, {}^{b}v) = ([\ 0,\ 0,\ \varrho,\ 1\ ]^T, [\ 0,\ 1,\ 0\ ]^T) \in \mathcal{FS}$$

[8] $T_1\mathcal{X} \subset T\mathcal{X}$ is the unit tangent bundle of $\mathcal{X}$ [14].

4.2.2 The camera map and visible set for the Buehgler

A perspective projection camera (10), located at the frame $\mathbf{F}_c$, is positioned to view the robot end effector as depicted in Figure 4. The FOV angles, $(\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}) \in (-\pi/2, \pi/2)$, determine the workspace of the camera, $\mathcal{W}_c \subset \mathbb{E}^3$, in a fashion analogous to the planar camera, namely

$$\begin{aligned}\mathcal{W}_c := \big\{ \; p \in \mathbb{E}^3 : p_3 > 0, \; \arctan(p_1/p_3) \in (\alpha_{11}, \alpha_{12}), \\ \arctan(p_2/p_3) \in (\alpha_{21}, \alpha_{22}) \; \big\} \; .\end{aligned}$$

The edge of the image plane is defined in terms of the "lower left corner" $y^{\min} \in \mathbb{E}^2$ and the "upper right corner" $y^{\max} \in \mathbb{E}^2$, the image plane projections of the FOV edges.

We define the total workspace as

$$\mathcal{W} := \{q \in \mathcal{Q} : r_\delta(q) \in \mathcal{W}_c\} \tag{22}$$

The set of configurations facing the camera, those for which the feature is not occluded by the arm, is defined very similarly as for planar visual servoing. Let ${}^b n = [\; 0, \; 0, \; -1 \;]^T$ denote the normal vector to the tip of the arm, facing "into" the arm. Then

$$\mathcal{F} := \left\{ q \in \mathcal{Q} : r_\delta(q)^T R(q) {}^b n > 0 \right\} \tag{23}$$

And finally, we have $\mathcal{V} := \mathcal{F} \cap \mathcal{W}$.

The map c, introduced formally below, can be intuitively described as follows. A "pointer" is attached to the tip of a arm, and a camera sees its position and orientation on the image plane. Roughly speaking, the waist (q_1) moves the feature in the image plane x direction, the shoulder (q_2) moves the feature in the image plane y direction and the wrist (q_3) rotates the feature on the image plane. The camera is positioned so the feature may move freely over the entire image plane and may be rotated to any orientation on the image plane at any base point.

It turns out [30] that r_δ and $R^b n$ depend only on the "base" (first two) configuration variables. Hence, according to (22) and (23), the visible set decomposes as $\mathcal{V} = \mathcal{R} \times \mathrm{S}^1$, *i.e.* the third configuration variable, q_3, is unrestricted at each base point $(q_1, q_2) \in \mathcal{R} \subset \mathrm{T}^2$.

The camera map is given by the projection of the feature, and its orientation on the image plane, namely

$$c(q) := \begin{bmatrix} \pi \circ r_\delta(q) \\ \angle \left\{ [D\pi \circ r_\delta(q)] \; v_\delta(q) \right\} \end{bmatrix} \tag{24}$$

where $\angle$ normalizes the vector on the image plane. Hence, the function c yields the position and orientation of our projected feature on the image plane, *i.e.* $c : \mathcal{V} \to \mathcal{Y}$, where $\mathcal{Y} = T_1\mathbb{E}^2 \approx \mathbb{E}^2 \times \mathrm{S}^1$. We used the symbolic algebra package Mathematica to generate a relatively simple closed-form expression for c, and its Jacobian $D_q c(q)$.

In our experiments, we position the camera so that all the rays within the workspace of the camera intersect the set $\mathcal{M}_\delta = r_\delta(\mathcal{R}) \subset \mathbb{E}^3$ transversely once. This requires the camera be sufficiently close to enable the robot tip to position the pointer anywhere on the image plane. Morever, we assume the camer is positioned so that $\mathcal{R}$ does not contain the kinematic singularity that occurs when the second axis is pointing straight up [30]. As a direct consequence we have i) $r_\delta : \mathcal{R} \approx \mathcal{M}_\delta$ is an analytic diffeomorphism and ii) the restriction of π defined by composition with the kinematics, $\pi \mid_{r_\delta(\mathcal{R})} : \mathcal{M}_\delta \to \mathbb{E}^2$, is an analytic diffeomorphism as well, $\pi \circ r_\delta : \mathcal{R} \approx \mathbb{E}^2$.

4.2.3 Diffeomorphism to the image plane for the Buehgler

The image space is defined

$$\mathcal{I}_\rho := \{\, (y,\theta) \in T_1\mathbb{E}^2 : y_i^{\min} \le y_i \pm \rho_i \le y_i^{\max},\ i = 1,2 \,\}$$

where $\rho_i,\ i = 1,2$ are positive constants which impose a "safe" border around the image plane. When $\rho_1 = \rho_2 = 0$, we write $\mathcal{I}_0$. Define

$$\mathcal{J} := \left\{ y \in \mathbb{E}^2 : y_i^{\min} < y_i < y_i^{\max},\ i = 1,2 \right\}.$$

and note that $T_1\mathcal{J} = \overset{\circ}{\mathcal{I}}_0$. In Appendix B, we show that c (24) is a diffeomorphism $c : \mathcal{V} \approx T_1\mathcal{J}$ in the case that δ is sufficiently small, which we now simply assume to be true (an assumption validated by the experiments).

4.2.4 Model space for the Buehgler

As in the planar example (see Section 4.1.3), we seek a compact manifold with boundary $\mathcal{D}$ on which to impose a navigation function. As before, this is done by taking the inverse image under c of a compact subset of the image plane, namely

$$\mathcal{D} = c^{-1}(\mathcal{I}_\rho) \subset \mathcal{V} \tag{25}$$

which again as before is a compact manifold with boundary.

4.2.5 Navigation function for Buehgler servoing

We require a change of coordinates from the image plane to a model space $\mathcal{Z}$ for $\mathcal{D}$. By letting g be a simple (affine) scaling and translation of the coordinates, the set $\mathcal{I}_\rho$ is diffeomorphic to the model space $\mathcal{Z}$ in (35) with $n = 2$ and $m = 1$. Hence, according to Appendix A,

$$\varphi = \widetilde{\varphi} \circ c \tag{26}$$

where $\widetilde{\varphi}$ is given by (37), is a navigation function on $\mathcal{D}$.

4.3 Example 3: Spatial body, spatial camera

Consider a free convex polygonal rigid body, with configuration space $\mathcal{Q} = \mathrm{SE}(3)$, and let $P = [\, p_1,\ \cdots,\ p_N \,]$, $p_i \in \mathbb{E}^3$, be a set of coplanar distinguishable points on a face of the body. Because the points are coplanar, they are in a 2-dimensional subspace of $\mathbb{E}^3$. Attach the body-fixed frame $\mathbf{F}_b$ such that the (x, y)-plane contains the feature points, and the z-axis is normal and oriented toward the interior of body. Hence the z-component of the points in body coordinates is zero, so for convenience let

$${}^bB = \Lambda\, {}^bP = \begin{bmatrix} {}^bb_1 & \cdots & {}^bb_1 \end{bmatrix}, \quad {}^bb_i = \begin{bmatrix} {}^bp_{i1} \\ {}^bp_{i2} \\ 1 \end{bmatrix} \in \mathbb{E}^2$$

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

denote the "reduced" homogeneous body coordinates, where ${}^bp_{i1}$ and ${}^bp_{i2}$ are the x and y components, respectively, of ${}^bP$. Note that ${}^bP = \Lambda^T \, {}^bB$. Let

$$p_c = \arg\min_{p \in \mathbb{E}^2} \max_{i \in \Gamma} \|p_i - p\|, \quad \rho = \max_{i \in \Gamma} \|p_i - p_c\|,$$
$$\Gamma = \{1, \ldots, N\}$$

denote the center and radius of the smallest sphere containing all the points.[9] Now, assume without loss of generality that $\mathbf{F}_b$ is located at p_c, *i.e.* ${}^bp_c = [\ 0,\ 0,\ 0,\ 1\ ]^T$. Finally, we let the homogenous matrix $H = {}^CH_B \in \mathrm{SE}(2)$ denote the rigid change of coordinates from the body to the camera frame. Once again $R = [r_1, r_2, r_3]$ and $\mathbf{T}$ denote the rotation and translation, respectively, effected by H.

4.3.1 The camera map and visible set for spatial servoing

The visible set is defined analogously for the spatial case as it was in the planar case in Section 4.1. The visible set is the set of all poses in SE(3) such that the feature points are within the FOV, and the body is facing the camera. The set of configurations facing the camera is defined

$$\mathcal{F} := \{H \in \mathrm{SE}(3) : \mathbf{v}(H) > 0\} \quad \text{where} \quad \mathbf{v}(H) := r_3^T \mathbf{T}.$$

The FOV angles $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}$ determine the workspace of the camera $\mathcal{W}_c \subset \mathbb{E}^3$

$$\begin{aligned} \mathcal{W}_c \quad := \quad \{\, & p \in \mathbb{E}^3 : p_3 > 0, \arctan(p_1/p_3) \in (\alpha_{11}, \alpha_{12}), \\ & \arctan(p_2/p_3) \in (\alpha_{21}, \alpha_{22}) \,\}. \end{aligned}$$

depicted in Figure 6. The total workspace $\mathcal{W} \subset \mathrm{SE}(3)$ is then

$$\mathcal{W} := \left\{H \in \mathrm{SE}(3) : H \, {}^bp_i \in \mathcal{W}_c, i \in \Gamma\right\}$$

and the visible set is defined as for the planar case $\mathcal{V} := \mathcal{F} \cap \mathcal{W}$.

To simplify the geometry, we restrict the visible set to a conservative subset $\mathcal{V}' \subset \mathcal{V}$ with respect to the FOV. In particular, we consider all translations which keep the point p_c within the so-called "admissible cone," $\mathcal{W}_\rho$

$$\mathcal{W}_\rho := \left\{p \in \mathbb{E}^3 : \mathcal{B}_\rho(p) \subset \mathcal{W}_c\right\}$$

depicted in Figure 6. In other words, $\mathcal{W}_\rho$ is all translations that keep a sphere of radius ρ completely within the FOV. If we restrict the translation of our rigid body to be within this set, then all the feature points will stay in the FOV regardless of the body orientation. Hence, the restricted workspace

$$\mathcal{W}' := \{H \in \mathrm{SE}(3) : \mathbf{T} \in \mathcal{W}_\rho\} \tag{27}$$

is combined with the facing set, to yield the conservative visible set $\mathcal{V}' := \mathcal{F} \cap \mathcal{W}' \subset \mathrm{SE}(3)$

The centroid is confined to move in an open solid cylinder, *i.e.* $\mathbb{R}^+ \times \mathbf{D}^2$ where $\mathbf{D}^n$ is an n-dimensional open disk. Recall that the body coordinate frame, $\mathbf{F}_b$ is attached such that the z-axis is orthogonal to the face (facing into the body) and the (x, y)-plane contains the feature points. Consider the fact that SO(3) is an SO(2) bundle over S^2, and identify the orientation of the z-axis with the base point in S^2. The requirement that the body faces the camera is a constraint on the the z-axis, namely that it always has a positive projection of onto the line-of-site. This yields an

[9] Note that simply computing the centroid of all the points does not necessarily give the center of the smallest sphere containing all the points, and so this somewhat more awkward definition is needed.

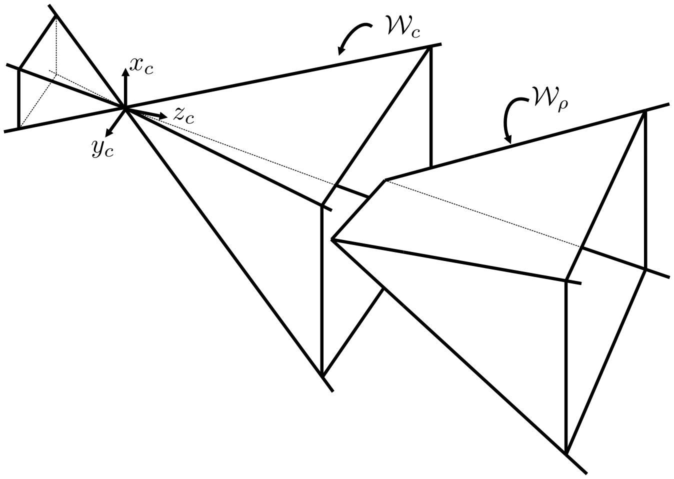

Figure 6: The camera workspace, $\mathcal{W}_c \subset \mathbb{E}^3$, is the set of all positions that are within the field of view. It is parameterized relative the camera frame by the angles $\{\alpha_i\}_{i=1}^4$ of the four FOV bounding planes. The admissible cone, $\mathcal{W}_\rho \subset \mathcal{W}_c$, refers to all positions that keep a sphere of radius ρ completely within the camera workspace.

open hemisphere; *i.e.* a diffeomorphic copy of $\mathbb{R}^2$. An SO(2) bundle over $\mathbb{R}^2$ is diffeomorphic to $\mathbb{R}^2 \times \mathrm{SO}(2)$. Therefore

$$\mathcal{V}' \approx \mathbb{R}^+ \times \mathbf{D}^2 \times \mathrm{SO}(2) \times \mathbb{R}^2 \approx \mathbb{R}^5 \times \mathrm{S}^1.$$

A perspective projection camera (9), located at the frame $\mathbf{F}_c$, observes the rigid body feature points, $\mathcal{Y} = \mathbb{P}^2 \times \cdots \times \mathbb{P}^2$ (N times). In homogeneous coordinates, we may write

$$Y = c(H) = C\,H\,{}^bP = A\,\Pi\,H\,\Lambda^T\,{}^bB$$

where C, A, Π are given in (9) and note that

$$Z(H) := \Pi\,H\,\Lambda^T = \Pi \begin{bmatrix} r_1 & r_2 & r_3 & \mathbf{T} \\ 0 & 0 & 0 & 1 \end{bmatrix} \Lambda^T = \begin{bmatrix} r_1 & r_2 & \mathbf{T} \end{bmatrix}$$

Denote the columns $Y = [\ y_1,\ \cdots,\ y_N\ ]$ where $y_i = C\,H\,{}^bb_i$ and note that Y and ${}^bB$ are related by the matrix $A\,Z(H)$, namely

$$Y = A\,Z(H)\,{}^bB \tag{28}$$

This fact will provide a convenient way to compute c^{-1}

4.3.2 Computing c^{-1} for spatial servoing

Because we will work in a reconstructed three dimensional space to perform 6DOF visual servoing, we need an inverse for the camera map. Computing the inverse requires that a map be injective onto its image which is true of c as we show so long as P contains at least four points in general position (i.e., having the property that no three are collinear).

We must first show that c is well defined on $\mathcal{V}$. Assume $H \in \mathcal{V} \subset \mathcal{F}$, *i.e.* that $\mathbf{v}(H) \neq 0$, which implies that $Z(H)$ is nonsingular. Since $Z(H)$ is nonsingular, $A\,Z(H)$ is a homography, and hence $c(H)$ is well defined.

Second, we must show that each H yields a unique Y. Assume $H_1, H_2 \in \mathcal{V}$ and $H_1 \neq H_2$. Using a contrapositive argument, assume that $Y = c(H_1) = c(H_2)$. Since ${}^b\!B$ has the property there are four points such that no three of the four points are collinear then we know that Y has the same property.[10] Hence $c(H_1) = c(H_2)$ iff $A\,Z(H_1)$ and $A\,Z(H_2)$ the are the same up to a scale. This implies that $Z(H_1)$ and $Z(H_2)$ are the same up to a scale, *i.e.*

$$\begin{bmatrix} r_{11} & r_{12} & \mathbf{T}_1 \end{bmatrix} = \beta \begin{bmatrix} r_{21} & r_{22} & \mathbf{T}_2 \end{bmatrix}$$

where r_{ij} denotes the j^{th} column of the i^{th} rotation matrix. But $\beta = 1$ since $\|r_{ij}\| = 1$ and $(\mathbf{T}_i)_3 > 0$. Hence $Z(H_1) = Z(H_2)$, which is true iff $H_1 = H_2$. This is a contradiction, so c is injective.

In practice, computing the inverse of c is very straight forward. There must be at least four point correspondences such that no three points are collinear as discussed above. Using the possibly redundant and noisy data, Y, one simply computes the "best" (in a linear sense) homography, M, between ${}^b\!B$ and Y [10]. Subsequently, using an approximation for the camera parameter matrix, compute $\hat{G} = \hat{A}^{-1}M$. With noiseless measurements and perfect calibration

$$H = Z^{-1}(\hat{G}) = \begin{bmatrix} \frac{\hat{G}_1}{\|\hat{G}_1\|} & \frac{\hat{G}_2}{\|\hat{G}_1\|} & \frac{\hat{G}_1 \times \hat{G}_2}{\|\hat{G}_1\|^2} & \frac{\hat{G}_3}{\|\hat{G}_1\|} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

however, due to calibration errors in $\hat{A}$ and noise in Y, $\hat{G}$ will not be exactly in the form of $Z(H)$. In our experiments we use a simple heuristic pseudo-inverse $\hat{H} = Z^{\dagger}(\hat{G})$ which is computed by projecting the above expression, conceived as a matrix in $\mathbb{R}^{4\times 4}$, down to the space of homogeneous transformation matrices, so that $Z(\hat{H})^{-1}\hat{G} - I$ is "small."

4.3.3 Model space for spatial servoing

To apply the machinery in Section 3, we restrict $\mathcal{D}$ to be a compact subset of $\mathcal{V}'$. Consider the two planes parallel to the image plane given by $z = \delta_{\min}$ and $z = \delta_{\max}$ such that they each intersect $\mathcal{W}_\rho$ as depicted in Figure 7. The *bounded workspace* is defined simply to be the compact set

$$\mathcal{W}_{\delta,\rho} := \left\{ p \in \mathbb{E}^3 : p \subset \overline{\mathcal{W}}_\rho,\ \delta_{\min} \leq p_3 \leq \delta_{\max} \right\} \tag{29}$$

We will limit the translation to this compact subset. Selection of $\delta_{\max,\min}$ is arbitrary provided that $\mathcal{W}_{\delta,\rho}$ is well defined. Practically, one chooses them to limit the range of depth of the robot: too far away and it becomes more difficult to sense the features; too close and we risk hitting the camera with the body. The safe workspace is now

$$\mathcal{W}'' := \{ H \in \mathrm{SE}(3) : \mathbf{T} \in \mathcal{W}_{\delta,\rho} \} \tag{30}$$

To parameterize the safe set, $\mathcal{D}$, we use the translation $\mathbf{T}$ and Euler angles (ϕ, ψ, θ), in a manner which avoids singularities. In particular the Euler angles $(0, 0, \theta)$ have the z-axis of the body parallel to the translation, *i.e.*

$$h(\mathbf{T}, \phi, \psi, \theta) = h_1(\mathbf{T}) h_2(\phi, \psi, \theta) \tag{31}$$

where

$$\begin{aligned} h_1(\mathbf{T}) &:= \begin{bmatrix} \frac{\mathrm{e}_2 \times \mathbf{T}}{\|\mathrm{e}_2 \times \mathbf{T}\|} & \frac{\mathbf{T} \times (\mathrm{e}_2 \times \mathbf{T})}{\|\mathbf{T} \times (\mathrm{e}_2 \times \mathbf{T})\|} & \frac{\mathbf{T}}{\|\mathbf{T}\|} & \mathbf{T} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ h_2(\phi, \psi, \theta) &:= \begin{bmatrix} R_x(\phi) R_y(\psi) R_z(\theta) & 0 \\ 0^T & 1 \end{bmatrix} \end{aligned}$$

[10] A projective transformation, $A \in \mathrm{PL}(2)$ is uniquely determined by the correspondence of four points in general position – that is, no three points may be collinear. Furthermore, given four points with the property that no three are collinear, their image under A has the same property [26].

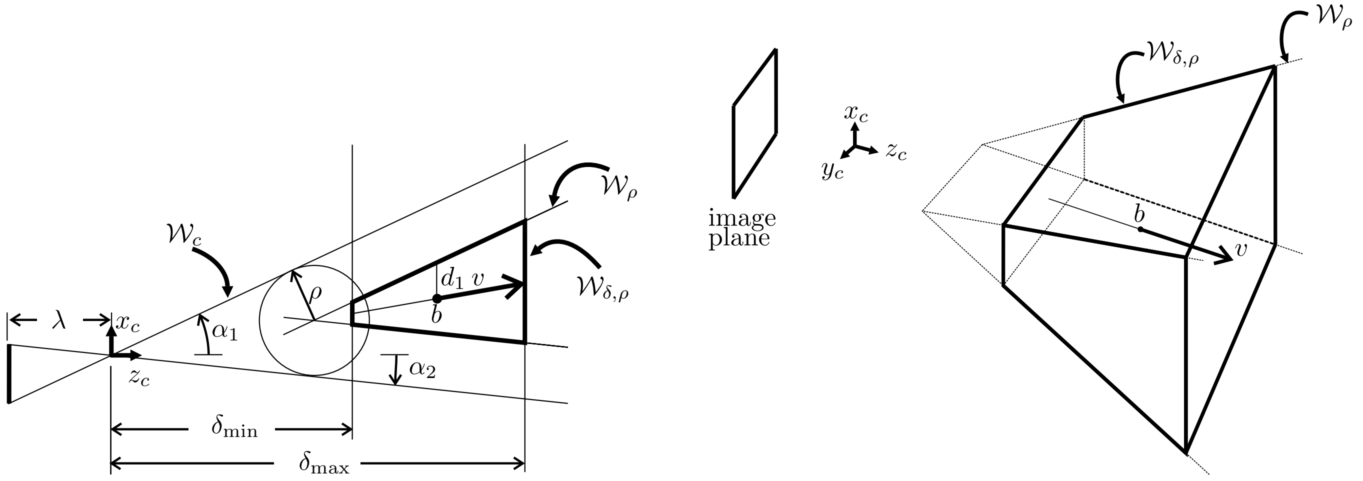

Figure 7: The *workspace* $\mathcal{W}_c$ is constrained by the FOV angles $\{\alpha_i\}_{i=1}^4$ **Left:** An orthographic projection of the workspace, illustrating the *admissible cone*, $\mathcal{W}_\rho$, and the *bounded workspace,* $\mathcal{W}_{\delta,\rho}$. **Right:** A perspective view of the admissible cone and the bounded workspace.

where R_x, R_y and R_z are the standard $x-y-z$ Euler angle rotation matrices. Consider a compact subset $\mathcal{F}' \subset \mathcal{F}$

$$\mathcal{F}' := \{H \in \mathrm{SE}(3) : -\vartheta_{\max} \leq \phi, \psi \leq \vartheta_{\max}\} \tag{32}$$

where H is parameterized by (31) and $\vartheta_{\max} < \pi/2$. We then have $\mathcal{D} := \mathcal{F}' \cap \mathcal{W}''$. Note that $\mathcal{D} \subset \mathcal{V}'$ is compact.

We need a mapping from $\mathcal{D}$ to $\mathcal{Z}$. We will now build up the inverse of that mapping $f^{-1} : [-1,1]^5 \times \mathrm{S}^1 \to \mathcal{D}$. The first piece is given by $f_1^{-1} : [-1,1]^3 \to \mathcal{W}_{\delta,\rho}$

$$f_1^{-1}(x) = b + x_3 v + x_1 \mathrm{e}_1(m_1 x_3 + d_1) + x_2 \mathrm{e}_2(m_2 x_3 + d_2)$$

where we choose the parameters $b, v \in \mathbb{R}^3$, $m_1, m_2, d_1, d_2 \in \mathbb{R}$ so that f_1^{-1} is a diffeomorphism from $[-1,1]^3$ to $\mathcal{W}_{\delta,\rho}$, as follows. The point b is on the midpoint of the line segment down the center of $\mathcal{W}_{\delta,\rho}$, as shown in Figure 7, and v is a vector which points along this centerline. To find b, intersect the four planes of $\mathcal{W}_\rho$ with a plane parallel to the image plane at $z = (\delta_{\min} + \delta_{\max})/2$, and take the midpoint of the rectangle that results. The vector v is found by subtracting b from the center of the rectangle which results from the intersection of the plane $z = \delta_{\max}$ and $\mathcal{W}_\rho$. Note that $b \pm v$ corresponds to the center points of the rectangles at the intersection of $z = \delta_{\max,\min}$, respectively, and $\mathcal{W}_\rho$. The constants m_1, m_2, d_1 and d_2 are all selected so that for each $x_3 \in [-1,1]$, then $(x_1, x_2) \in [-1,1]^2$ parameterizes a planar cross section of $\mathcal{W}_{\delta,\rho}$ parallel to the image plane. To select these constants, observe the (x,z)-orthographic projection of the workspace in Figure 7. The value of d_1 is shown and is computed by intersecting the plane $z = (\delta_{\min} + \delta_{\max})/2$ with $\mathcal{W}_\rho$ and taking $1/2$ of the x-dimension of the resulting rectangle, whereas d_2 is $1/2$ the y-dimension. The value of m_1 is found by observing that $x_1 = 1$ and $x_2 = 0$ corresponds to the top edge of $\mathcal{W}_\rho$. Likewise for m_2. We then have

$$f^{-1}(x,\theta) = h_1(f_1^{-1}(x_1, x_2, x_3))h_2(x_4\vartheta_{\max}, x_5\vartheta_{\max}, \theta) \tag{33}$$

Note that computing f is a very straightforward, albeit somewhat tedious, computation.

The topology of $\mathcal{D}$ is borne out by this change of coordinates: the translation, $\mathbf{T}$, is free to move in a diffeomorphic copy of $[-1,1]^3$, and the first two Euler angles are constrained to a copy of $[-1,1]^2$, while the last is free to move in the entire circle, S^1. Moreover, the parameterization was carefully designed to have no singularities. In essence, we have shown that the model space (35) for $n = 5$ and $m = 1$, namely $\mathcal{Z} = [-1,1]^5 \times \mathrm{S}^1$ is diffeomorphic to $\mathcal{D}$ via the mapping $f : \mathcal{D} \approx \mathcal{Z}$ (33).

4.3.4 Navigation function for spatial visual servoing

Let $g = f \circ c^{-1} : \mathcal{I} \to \mathcal{Z}$. Then

$$\varphi = \widetilde{\varphi} \circ g \circ c \tag{34}$$

where $\widetilde{\varphi}$ is given by (37) for $n = 5$, $m = 1$, is a navigation function for spatial visual servoing.

5 Empirical Validation

In order to test the framework proposed in Section 4 we experimented with two robotic systems that implement the latter two imaging models introduced in the previous section. The first system is the custom 3DOF direct drive Buehgler Arm described in Section 4.2 to test a fully dynamical controller (13) based on the NF given by (26). Our second set of experiments employ an industrial 6DOF RTX robot from Universal Machine Intelligence to test a kinematic controller (12) using the spatial 6DOF NF (34).

5.1 Calibration

Although the visual servoing methodolgy confers robustness against parameter mismatch in practice, all such methods, including the one presented in the paper, require at least coarse calibration of the robot and the camera. For the RTX, we used the manufacturer specified Denavit-Hartenberg parameters and a linear method ([10], Section 3) that requires a set of point correspondences between points in space and their respective image to simultaneously estimate both intrinsic and extrinsic camera parameters. To obtain the correspondences, a feature affixed to the robot end effector was moved to a grid of positions in view of the vision system which extracted an image plane location for each feature position in space. For the Buehgler setup we measured the paddle length ϱ and the shoulder offset δ by hand and proceeded as for the RTX to obtain a rough estimate of the camera parameters. A gradient algorithm based on a simple pixel disparity cost function refined our parameter estimates for the 11 camera and two robot parameters.

5.2 The Buehgler Arm

The Buehgler Arm is controlled by a network of two Pentium II class PCs running LynxOS (http://www.-lynx.com/), a commercial real-time operating system. The two nodes communicate on a private ethernet using the User Datagram Protocol (UDP). The first captures 8bit 528x512 pixel images at 100Hz using an Epix (http://www.epixinc.com/) Pixci D frame grabber connected to a DALSA (http://www.dalsa.com/) CAD6 high-speed digital camera. The images are processed to extract the location (position and orientation) of the feature point at the end of the paddle. The second node implements servo control using the Trellis (http://www.trellissoftware.com/) motion controller with a servo rate of 1kHz, based on the dynamical controller in (13), wherein the damping term is computed from encoder data using finite differencing to estimate the joint velocities.

Two sets of experiments were conducted using the appropriate NF (26), implemented with two different gain settings (*i.e.*, assignments for the parameter array K in (36) and K_d in (13)) chosen to contrast performance resulting from a locally well tuned critically damped closed loop using relatively high gains, as against a "detuned" low gain and underdamped circumstance. Each trial consisted of driving the feature position and orientation to a goal (z^*, ζ^*) from some initial condition in joint space $(q_0, \dot{q}_0)$. For the "tuned" gain experiments, a set of 8 goal locations with and 40 initial conditions were chosen in an effort to "defeat" the controller. In particular, initial configurations were chosen near the edge of the FOV, with initial velocity vectors chosen so as to drive the robot out of the FOV. The initial conditions were prepared with a simple joint-space trajectory planner and joint-space PD controller that drove the robot to the starting state at which time the control switched to the NF based controller. In other words, we forced the robot to literally "fling" itself toward an obstacle before turning on our visual servoing controller. Both the goal positions and initial conditions where chosen to span the visible robot workspace. The control law gains were hand-tuned to provide nearly critically damped performance, and settling times on the order of a second.[11] For the "detuned" gain experiments, a smaller set of more aggressive initial conditions and goal locations was used, and the damping gain was reduced to provide "underdamped" performance. There were 4 goals and 8 initial conditions. Figure 8 shows the the error coordinates of a typical run for both "tuned" and "detuned" gains.

Table 1: Summary of results for Buehgler arm.

		Normalized Path Length		
Goal #	Succ. Rate	Jnt. Space Mean (dev)	Pix. Space Mean (dev)	Gains Tuned?
1	40/40	1.75 (0.12)	1.63 (0.19)	Yes
2	40/40	2.38 (0.34)	1.85 (0.35)	Yes
3	36/40	2.02 (0.87)	1.65 (0.22)	Yes
4	40/40	2.07 (0.36)	1.94 (0.50)	Yes
5	40/40	1.79 (0.36)	1.64 (0.23)	Yes
6	37/40	2.15 (0.41)	2.00 (0.62)	Yes
7	36/40	1.96 (0.85)	1.63 (0.19)	Yes
8	40/40	2.05 (0.65)	2.02 (0.75)	Yes
1	6/8	3.01 (1.93)	3.59 (1.94)	No
2	6/8	1.64 (0.40)	2.41 (0.53)	No
3	5/8	2.54 (1.86)	3.91 (2.40)	No
4	7/8	2.97 (0.46)	3.30 (0.65)	No

To quantify the results we examine similar measures as for the RTX. Table 1 summarizes the results of the our experiments. With well tuned gains the controller consistently drove the feature to the goal location with a rate of success of 97%. Of the 11 errors one was due to exceeding the robot's maximum velocity, one to a software driver error, and one to a feature leaving the FOV of the camera during initialization. The remaining 8 failures were caused by not allowing enough time for convergence as each experiment lasted 6 seconds. These errors generally arose when the robot was close to a saddle of the NF so the controller was slow to overcome the robot's unmodeled friction. However, with "detuned" gains and high initial velocity the feature left the FOV 25% of the time. These failures are due to the fact that the initial energy of the robot arm caused the arm to escape the

[11] Of course, the allusion to linear notions of damping is merely an intuitive designer's convenience. We chose gains to ensure the local linearized system was critically damped at the eight equilibrium states, and then tuned up the "boundary" gains to force reasonably snappy descent into the domain wherein the linearized approximation was dominant.

potential well – by using a lower "detuned" gain on the potential energy feedback term, the potential barrier is reduced. (It would not be difficult to compute the invariant domain, as in [17] - these experiments give some sense of the relatively graceful performance degradation consequent upon imperfectly tuned gains.) Figure 8 shows image-based error plots and the image-plane trajectory for two typical runs.

To determine the accuracy with which the feature reached the goal, the mean pixel error is given by

$$256\sqrt{(z_{\text{final}}-z^*)^T(z_{\text{final}}-z^*)+(\tfrac{1}{\pi})^2(\zeta_{\text{final}}-\zeta^*)^2}$$

so the model coordinates are scaled to be commensurate with pixel error. As can by seen in Figure 8 (bottom right) the mean errors are in the neighborhood of 1 to 2 pixels over each of the eight goal positions.

Table 1 shows the path length taken by the robot compared to the straight line path length in both joint and pixel space. The results indicate that in joint space our navigation function produced results within a factor of 2.5 of the straight line distance. We attribute a large portion of this additional path length to the fact that the large initial velocities caused "curviness" to the trajectories.

Finally, we designed our controller to have a very rapid and dextrous response. The Buehgler arm has a mass in excess of 100Kg making precise, quick and efficient movement quite challenging. Figure 8 (top right) shows our navigation based controller produced a one second or less five percent settle time for seven of the eight primary goal positions.

5.3 The RTX Arm

The RTX is commanded through the serial port of a single Pentium PC running a Linux 2.0 kernel (hard real-time is not required, hence the standard Linux kernel was adequate). The PC is equipped with a Data Translations[12] DT3155 frame grabber connected to a standard 30Hz NTSC video camera. Using MATLAB's *C*-language API, we created a simple interface to the camera and robot accessible from within the MATLAB programming language.

The theory presented in Section 4.3 presumes the configuration space to be $\mathcal{Q} = \mathrm{SE}(3)$. However, $\mathcal{Q}$ is parameterized (locally) by the robot joint angles $q \in \mathbb{R}^6$ through the forward kinematics, namely $h : \mathbb{R}^6 \to \mathcal{Q}$. Of course, inevitably, all such kinematic parameterizations introduce singularities that may, in turn, inject spurious critical points to the gradient fields, necessarily actuated in the robot's joint space rather than in the task space, as our theory presumes. Similarly, since our formal theory "knows" only about visibility bounds, the robot's unmodeled joint space angles limits are not in principle protected against.[13] However, the weight of experimental evidence we present below suggests that these discrepancies between presumed model and physical reality do not seriously imperil the practicability of this scheme. Regarding the first discrepancy, the absence of stalled initial conditions suggests that any critical points so introduced were not attractors. Regarding the second, we found that choosing initial and goal locations away from the joint space boundaries was sufficient to avoid running into the end-stops.

The RTX controller employs first order gradient descent on the navigation function in (34). Because the RTX arm accepts only position commands, given goal and current images with feature points extracted, the gradient update was implemented iteratively, as follows:

[12] `http://www.datax.com/`

[13] Addressing the further practical realities of kinematic singularities and robot joint space limitations falls outside the scope of the present paper (and, indeed, is not even addressed at all in the traditional visual servoing literature). In principle, the NF framework would be relevant to these problems as well: joint space limits are analogous to the FOV obstacles, while the kinematic singularities are akin to self-occlusion.

$u_k \Leftarrow -D_q^T \varphi = -D_q^T (f \circ h)(q_k)\, D_z^T \widetilde{\varphi}_{z^*}(z_k)$,
$q_{k+1} \Leftarrow q_k + \alpha u_k$ (where α is the step size).

Note that the Jacobian matrix $D_q(f \circ h)$ can be decomposed into the product of Df, which maps from the body screw axis to the model space, and the manipulator Jacobian Dh, which maps from the robot joint space to the body screw axis. Indeed, such a decomposition implies the extrinsic parameters are not needed and hence one may move the camera without recalibrating.

To explore our algorithm, we conducted a set of experiments in which 58 initial conditions were tested for four goal locations, giving 232 candidate experiments. Both the initial conditions and goal locations were chosen randomly from a grid of 4096 points in model space (configurations near kinematic singularities and not within the robot workspace were removed). We chose many initial and goal configurations near the boundaries of the workspace, hence of the 232 candidate experiments, an additional 29 experiments where removed as the features began outside of the robot's workspace due to "sloppiness" in the joints of the robot *i.e.* there is a lot of play in the joints so a specific initial condition in the joint space may not be the same place in SE(3) in subsequent trials, and therefore some initial conditions were out of the visible workspace at the start. Initially, the robot was moved to each goal location to capture an image of the robot, respecting which the vision system stored the desired location of feature points, y^*. Figure 9 shows the pixel errors feature trajectories of two typical runs. As shown, we used four coplanar feature points for the camera map, $c : \mathcal{Q} \to \mathcal{Y}$.

Table 2: Summary of results for RTX arm

		Normalized Path Length	
Goal \#	Success Rate	Jnt. Space Mean (dev)	Pix. Space Mean (dev)
1	49/51	1.35 (0.39)	1.33 (0.49)
2	47/47	1.36 (0.29)	1.27 (0.37)
3	55/57	1.32 (0.23)	1.27 (0.32)
4	47/48	1.42 (0.36)	1.32 (0.25)

To ascertain the performance of our controller, we employed several metrics described below: success and failures, efficiency of motion, mean pixel error and setting time.

Table 2 shows the success rate of the various goal positions. Of 203 trial runs, 5 were found to have failed. All 5 failures are due to the robot not converging in our limit of 30 iterations (though after inspecting the data by hand, it appears that the robot would have converged if given a few more iterations).

We also measured the "efficiency" of motion relative to the straight line distance in both image and cartesian space. The metric used for measuring distances in the configuration space was the sum of the angular displacement, scaled by body length, and the translational displacements. For all of the runs, both image and cartesian measures indicated that the path length was around 1.4 times that of straight line distance. See Table 2.

Using the root mean squared average pixel error measurement given by $\sqrt{\frac{1}{4}\sum_{i=1}^{4}(y_i - y_i^*)^2}$ we found an average final pixel error on the order of 1-2 pixels upon convergence. Figure 9 (upper right) shows the mean pixel error and standard deviation for each of the four unique goal positions.

The average five percent setting time, shown in Figure 9 (lower right), was approximately 10-14 iterations for each of the four goal locations, averaged over all successful runs.

6 Conclusions

This paper addresses the problem of driving image plane features to some goal constellation while guaranteeing their visibility at all times along the way. We cast the problem as an instance of generalized dynamical obstacle avoidance, thereby affording the use of navigation functions in a nonlinear PD-style feedback controller.

We demonstrate the applicability of this framework in three different examples, including extensive empirical results in the two more complex cases. The two experimental systems confirmed the practicability of theoretical framework: the custom 3DOF Buehgler Arm and the 6DOF commercial RTX arm. For the Buehgler, our experiments suggest that the navigation function based controller indeed drives the feature to within a few pixels of the goal in all cases where the initial energy did not overwhelm the potential energy well. The kinematic experiments with the RTX validated our 6DOF task-space servo architecture. In both cases our results show systems with large basins of attraction that both avoid self occlusion and respect FOV constraints.

The efficacy of this approach depends upon the explicit construction of a model space together with a coordinate transformation back to the image plane respecting which the visibility obstacles such as self occlusions and the FOV can be represented as the boundary of a compact manifold. Finding a good general image-based coordinate system seems challenging, hence we have had to resort to similar but distinct constructions in the examples discussed. Ideally, one creates an image based coordinate system using direct feature information as in the planar setting of Section 4.1 and the dynamical setting of Section 4.2. In many cases, it may be helpful to "enlarge" the obstacles to simplify the topology as in the spatial setting of Section 4.3. Conceivably, the homography matrix (25) computed directly from image-plane correspondences, could provide the basis for a more general visual coordinate system, and this remains an interesting avenue for future investigation.

7 Acknowledgments

This work was supported in part by the National Science Foundation (IRI-9510673), the Rackham School of Graduate Studies (Predoctoral Fellowship for the first author) and DARPA/ONR (N00014-98-1-0747). Thanks to Rob Ghrist for his insight into the topology of SO(n). Also, thanks to Gabriel Lopes, Clark Haynes, Eric Carlson and Eric Westervelt for their development efforts, and to Trellis Software for their help in our controller integration effort.

A A navigation function for $[-1,1]^n \times \mathrm{T}^m$

Let

$$\mathcal{Z} = [-1,1]^n \times \mathrm{T}^m \tag{35}$$

for some $m, n \in \mathbb{N}$ and let $(z^*, \zeta^*) \in \mathring{\mathcal{Z}}$ denote a goal. Consider the function $f : (-1,1)^n \to \mathbb{R}^n$

$$f(z) = \begin{bmatrix} \frac{z_1 - z_1^*}{(1-z_1^2)^{\frac{1}{2}}} & \cdots & \frac{z_n - z_n^*}{(1-z_n^2)^{\frac{1}{2}}} \end{bmatrix}^T .$$

Let $K \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix and $\kappa_i > 0$, $i = 1, \ldots, m$. Define[14]

$$\overline{\varphi}(z,\zeta) := \tfrac{1}{2} f(z)^T K f(z) + \sum_{i=1}^{m} \kappa_i (1 - \cos(\zeta_i - \zeta_i^*)). \tag{36}$$

Proposition 4. *The objective function*

$$\widetilde{\varphi} := \frac{\overline{\varphi}}{1+\overline{\varphi}} \tag{37}$$

is a navigation function on $\mathcal{Z}$, where $\overline{\varphi}$ is given in (36).

Proof. According to Definition 1, $\widetilde{\varphi}$ must be a smooth Morse function which evaluates uniformly to unity on the boundary of $\mathcal{Z}$, and has (z^*, ζ^*) as the unique minimum.

The boundary of $\mathcal{Z}$ is given by

$$\partial\mathcal{Z} = \left\{(z,\zeta) \in \mathcal{Z} : z_i = \pm 1, i \in \{1,\ldots,n\}\right\}.$$

Clearly, $\widetilde{\varphi}$ evaluates to 1 on the boundary, *i.e.* as $z_i \to \pm 1$ then $\widetilde{\varphi} \to 1$. Furthermore, $\forall (z,\zeta) \in \mathcal{Z}$, $\overline{\varphi} >= 0$. Moreover, $\overline{\varphi} = 0$ iff $(z,\zeta) = (z^*, \zeta^* + \sum_{i\in\Gamma} 2\pi \mathrm{e}_i) = (z^*, \zeta^*)$, *i.e.* the (z^*, ζ^*) is the global minimum.

To study the critical points of $\widetilde{\varphi}$, we need only study those of $\overline{\varphi}$, because the function $\sigma : [0,\infty) \to [0,1)$ given by $\sigma(x) = x/(1+x)$ has derivative $\sigma'(x) = 1/(1+x)^2$, which does not introduce any spurious critical points. The critical points of $\overline{\varphi}$ are found by solving

$$0 = D\overline{\varphi} = \begin{bmatrix} f^T K Df, & \kappa_1 \sin(\zeta_1 - \zeta_1^*), \cdots, \kappa_m \sin(\zeta_m - \zeta_m^*) \end{bmatrix} \tag{38}$$

noting that

$$Df = \operatorname{diag}\{f'\}, \quad \text{where} \quad f_i'(z) := \frac{1 - z_i z_i^*}{(1 - z_i^2)^{3/2}}, \quad i = 1, \ldots, n.$$

Since Df is nonsingular on $(-1,1)^n$, $D\overline{\varphi} = 0$ iff $f = 0$ and $\sin(\zeta_i - \zeta_i^*) = 0$, $i = 1, \ldots, n$ which is true iff $(z,\zeta) = (z^*, \zeta^* + \sum_{i\in\Gamma} \pi \mathrm{e}_i)$, $\Gamma \in \text{powerset}\{1,\ldots,m\}$. There are 2^m index sets which enumerate all possible critical points. One readily verifies that the Hessian is nonsingular at every critical point and (z^*, ζ^*) is the only minimum. Hence $\widetilde{\varphi}$ is a Morse function which evaluates uniformly to unity on the the boundary, has $2^m - 1$ saddles and the goal is the unique minimum. □

B Buehgler map diffeomorphism

As described in Section 4.2, the physical Buehgler arm is a three degree of freedom (all revolute) kinematic chain whose free joint space, $\mathcal{V} \subset \mathrm{T}^3$, is the cross product of the (bounded) angles allowed its "base" (first two) joints, $\mathcal{R}$ together with with the (unbounded) circular motion allowed its last

[14] Since use local coordinates to define our cost function (36), we require $\overline{\varphi}(z,\zeta) = \overline{\varphi}(z, \zeta + \sum_{i\in\Gamma} 2\pi \mathrm{e}_i)$ for all possible index sets $\Gamma \in \text{powerset}\{1,\ldots,m\}$, where e_i is the unit vector whose i^{th} element is 1. This ensures that $\overline{\varphi}$ is continuous on $\mathcal{Z}$.

joint — that is, $\mathcal{V} := \mathcal{R} \times \mathrm{S}^1$. In a similar vein as the image space, we identify the joint space with the unit tangent bundle, $\mathcal{V} \approx T_1\mathcal{R} \subset T\mathcal{R}$.

If there were no "shoulder" offset ($\delta = 0$) then the workspace image of the kinematic base map would describe a metrically spherical surface in the codomain (three space), *i.e.* $r_0(\mathcal{R}) = \mathcal{M}_0 \subset \mathrm{S}^2_\varrho$, where ϱ is the length of the arm [30], and the final joint axis would permit motion of the pointer on the tip of the arm along a perfect copy of the unit circle tangent to the surface at each "base point", $s \in \mathcal{M}_0$. For this simplified version of the kinematics, then, we are able to model the range space of the full kinematic map as the unit tangent bundle over the base image, $T_1\mathcal{M}_0 \subset T_1\mathbb{E}^3$. It follows that the full kinematic map, h_0, may be modeled as the restriction of the tangent map, Tr_0, to the unit tangent bundle over the base joint space, $h_0 = Tr_0 \mid_{T_1\mathcal{R}}$. Specifically, h_0 is a fiber map [14], taking its domain in the unit tangent bundle over the base joints, $T_1\mathcal{R}$ and its image in the unit tangent bundle over the surface, $T_1\mathcal{M}_0$.

As described in the body of the text, we have so arranged the camera relative to the Buehgler that $\pi \circ r_\delta$ is an analytic diffeomorphism between the robot base joints, $\mathcal{R}$, and the camera image plane, $\mathcal{J} \subset \mathbb{E}^2$. In general, the tangent lift of a diffeomorphism between two spaces is itself a diffeomorphism (of one lower grade of differentibility) between their tangent spaces, and it follows that $T(\pi \circ r_\delta)$ is an analytic diffeomorphism between $T\mathcal{R}$ and $T\mathbb{E}^2$.

Now, for $\delta = 0$ we have

$$\begin{tikzcd} T_1\mathcal{R} \arrow[r, "h_0"] & T_1\mathcal{M}_0 \arrow[r, "T\pi|_{T_1\mathcal{M}_0}"] & T_1\mathbb{E}^2, \end{tikzcd}$$

hence, we may view the camera map as the composition $c = T\pi \mid_{T_1\mathcal{M}_0} \circ h_0$, which, in turn may be rewritten,

$$\begin{aligned} c &= T\pi \mid_{T_1\mathcal{M}_0} \circ h_0 \\ &= T\pi \mid_{T_1\mathcal{M}_0} \circ Tr_0 \mid_{T_1\mathcal{R}} \\ &= (T\pi \circ Tr_0) \mid_{T_1\mathcal{R}} \\ &= T(\pi \circ r_0) \mid_{T_1\mathcal{R}}, \end{aligned}$$

as the restriction of the diffeomorphism $T(\pi \circ r_0)$ to the unit tangent bundles over the domain and range of the base map composition, $(\pi \circ r_0)$. In general, the restriction of a diffeomorphism to a smooth submanifold in the domain must remain a diffeomorphism onto its image, hence, when $\delta = 0$, we have shown that c is a diffeomorphism between the configuration space, $T_1\mathcal{R}$ and the image space, $T_1\mathcal{J}$.

Since the property of being a diffeomorphism is an open property, it follows that c remains a diffeomorphism for small enough perturbations, $\delta \neq 0$.

References

[1] V. I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, 1989.

[2] Louis Auslander and Robert E. MacKenzie. *Introduction to Differentiable Manifolds.* Dover, 1977.

[3] Francois Chaumette. *The Confluence of Vision and Control*, chapter Potential problems of stability and convergence in image-based and position-based visual servoing. Springer-Verlag, 1999.

[4] P. I. Corke and M. C. Good. Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[5] 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.

[6] Noah J. Cowan and Daniel E. Koditschek. Planar image based visual servoing as a navigation problem. In *International Conference on Robotics and Automation*, volume 1, pages 611–617, Detroit, MI, 1999. IEEE.

[7] Noah J. Cowan, Gabriel A. D. Lopes, and Daniel E. Koditschek. Rigid body visual servoing using navigation functions. In *Conference on Decision and Control*, pages 3920–3926, Sydney, Australia, 2000. IEEE.

[8] Noah J. Cowan, Joel D. Weingarten, and Daniel E. Koditschek. Empirical validation of a new visual servoing stragegy. In *Conference on Control Applications*, Mexico City, September 2001. IEEE, Omnipress.

[9] J. Craig. *Introduction to Robotics*. Addison-Wesley, Reading, Mass., 1986.

[10] Olivier Faugeras. *Three-Dimensional Computer Vision*. MIT Press, London, England, 1993.

[11] H. Goldstein. *Classical Mechanics*. Addison-Wesley, Reading Mass., 1950.

[12] Robert M. Haralick and Linda G. Shapiro. *Computer and robot vision*, volume I and II. Addison-Wesley, 1992.

[13] Koichi Hashimoto and Toshiro Noritsugu. Enlargement of stable region in visual servo. In *Conference on Decision and Control*, volume 4, pages 3927 – 3932, Sydney, Australia, 2000. IEEE.

[14] Morris W. Hirsch. *Differential Topology*. Springer-Verlag, 1976.

[15] 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.

[16] D. Kim, A. A. Rizzi, G. D. Hager, and D. E. Koditschek. A “robust” convergent visual servoing system. In *International Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, 1995. IEEE/RSJ.

[17] D. E. Koditschek. The control of natural motion in mechanical systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 113(4):547–551, Dec 1991.

[18] D. E. Koditschek. An approach to autonomous robot assembly. *Robotica*, 12:137–155, 1994.

[19] 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.

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

[21] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[22] Ezio Malis, Francois Chaumette, and Sylvie Boudet. 2-1/2-d visual servoing. *IEEE Transactions on Robotics and Automation*, pages 238–250, 1999.

[23] Joseph L. Mundy and Anderw Zisserman, editors. *Geometric Invariance in Computer Vision*, chapter 23, pages 463–519. MIT, 1992.

[24] James R. Munkres. *Topology, a first course*. Prentice Hill, 1975.

[25] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Reading, Mass., 1994.

[26] Dan Pedoe. *Geometry, a comprehensive course*. Dover Publications, Inc., New York, 1970.

[27] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.

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

[29] A. A. Rizzi and D. E. Koditschek. An active visual estimator for dexterous manipulation. *IEEE Transactions on Robotics and Automation*, pages 697–713, October 1996.

[30] Alfred A. Rizzi, Louis L. Whitcomb, and Daniel E. Koditschek. Distributed real-time control of a spatial robot juggler. *IEEE Computer*, 25(5):12–24, May 1992.

[31] Andreas Ruf and Radu Horaud. Visual servoing of robot manipulators, part "i" : Projective kinematics. *International Journal on Robotics Research*, 18(11):1101 – 1118, November 1999.

[32] Camillo J. Taylor and James P. Ostrowski. Robust visual servoing based on relative orientation. In *International Conf. on Robotics and Automation*, 1998.

[33] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-theshelf tv cameras and lenses. *IEEE Transactions on Robotics and Automation*, 3(4):323–344, August 1987.

[34] Louis L. Whitcomb, Alfred A. Rizzi, and Daniel E. Koditschek. Comparative experiments with a new adaptive contoller for robot arms. *IEEE Transactions on Robotics and Automation*, 9(1):59–70, Feb 1993.

[35] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee. The perspective view of three points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):66–73, 1991.

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

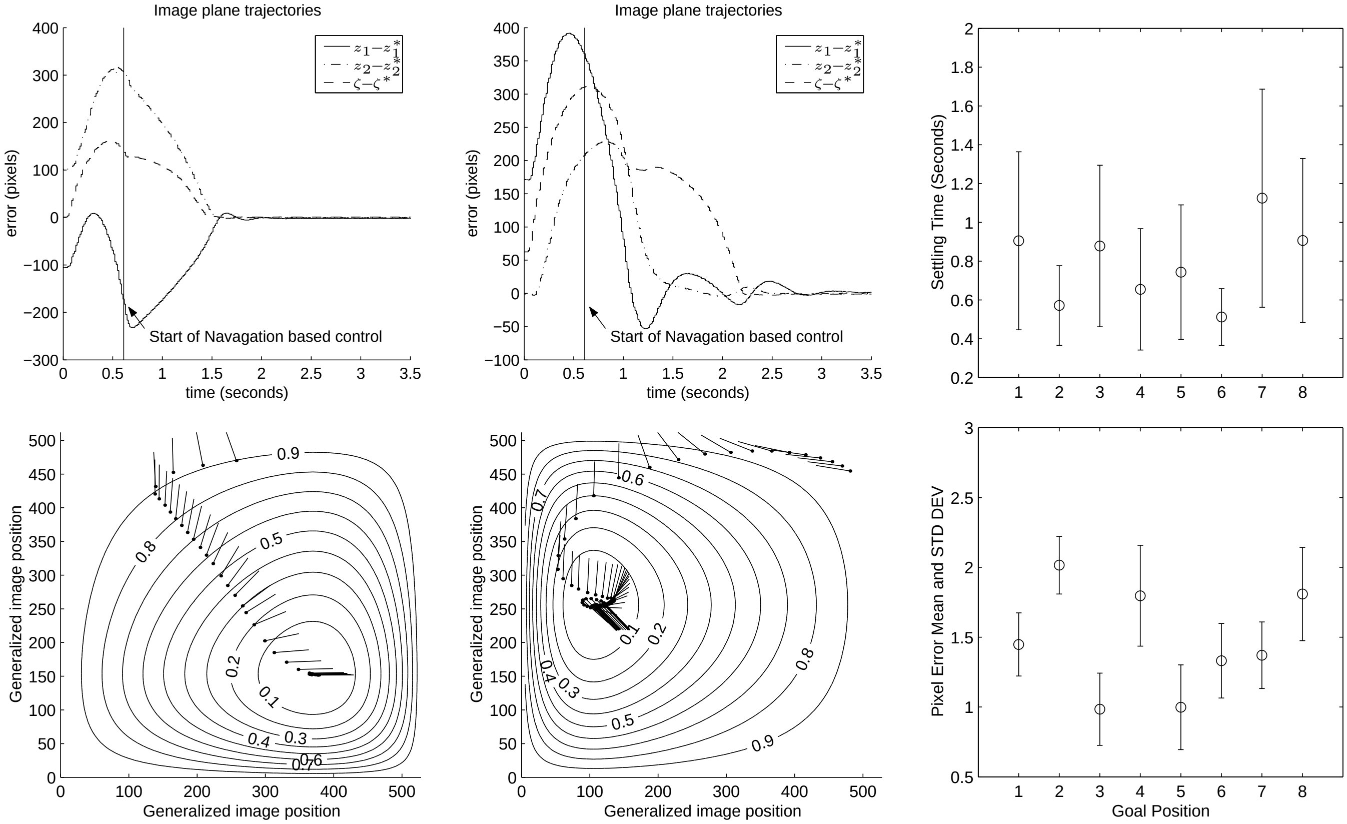

Figure 8: **Left:** The pixel error and corresponding image plane feature trajectory for a typical high gain trial on the Buehgler robot. **Middle:** A typical low gain trial, with a different initial and goal locations. A two-dimensional cross section (with $\theta = \theta^*$) of the levels sets of the NF is superimposed on the image plane trajectory. **Right:** Buehgler convergence results. *Top:* Five percent settling time for each of the eight high-gain goal positions. *Bottom:* The mean pixel error for each of the eight goal positions.

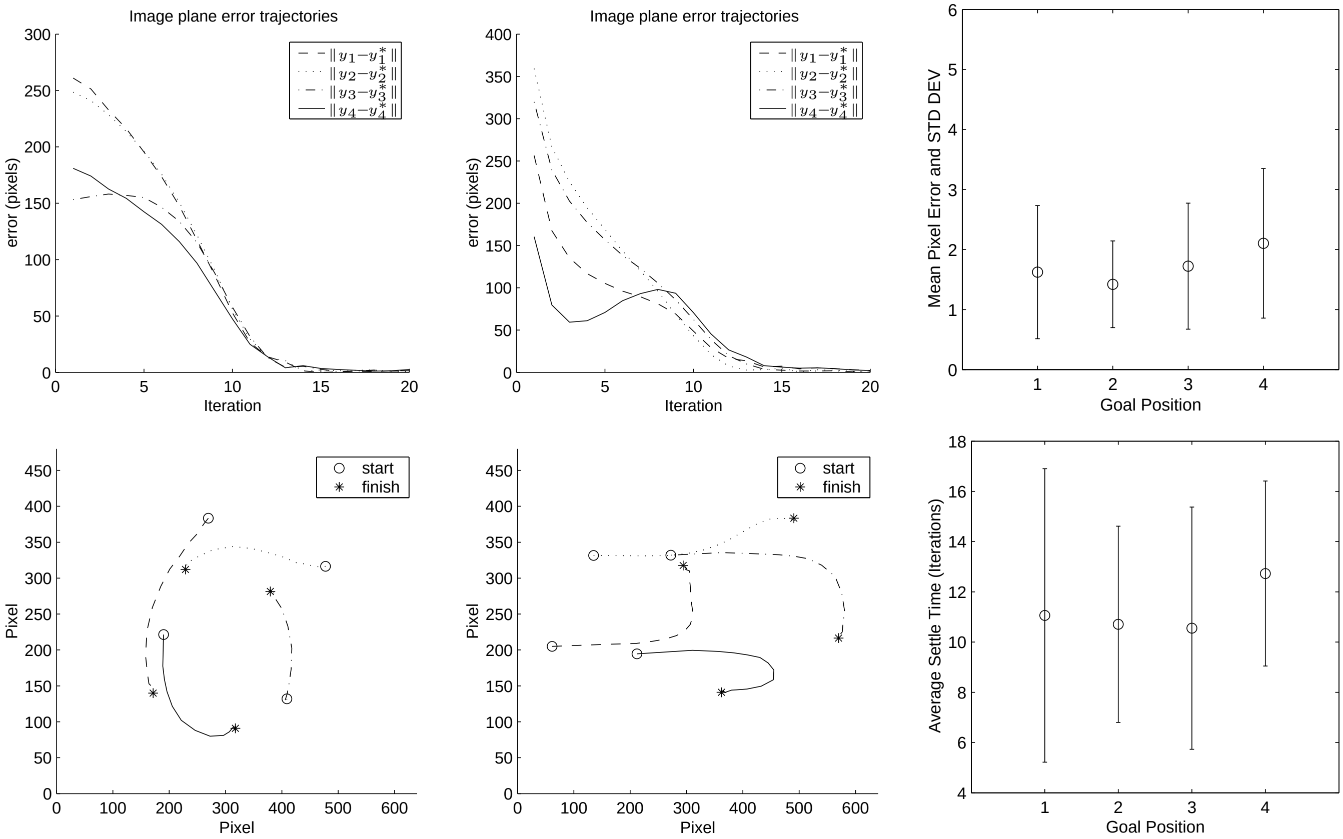

Figure 9: **Left:** The pixel error and corresponding image plane feature trajectory for a typical trial. **Middle:** Another typical trial, with a different initial condition and goal location. **Right:** RTX convergence results. *Top:* The mean pixel error for each of the four goal positions. *Bottom:* Five percent settling time for each of the four goal positions.