FEEDBACK AUGMENTATION OF PDE-BASED IMAGE SEGMENTATION ALGORITHMS USING APPLICATION-SPECIFIC EXOGENOUS DATA

A Thesis
Presented to
The Academic Faculty

by

Peter A. Karasev

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Electrical and Computer Engineering

Georgia Institute of Technology
December 2013

FEEDBACK AUGMENTATION OF PDE-BASED IMAGE SEGMENTATION ALGORITHMS USING APPLICATION-SPECIFIC EXOGENOUS DATA

Approved by:

Professor Patricio Vela, Advisor
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Allen Tannenbaum (co-advisor)
School of Computer Science
SUNY Stony Brook

Professor Anthony Yezzi
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mark Richards
Electrical and Computer Engineering
Georgia Institute of Technology

Professor John Xerogeanes
School of Medicine
Emory University

Professor Jeff Shamma
Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 2 April 2013

ACKNOWLEDGEMENTS

First, a sincere thanks goes to my advisors Allen Tannenbaum and Patricio Vela for introducing me to computer vision and control theory and guiding me through graduate studies. Allen, your insight into mapping challenging real-world problems into rigorous mathematics has been inspiring. Patricio, learning to emulate your methodical and detailed approach to research has helped me greatly. Next, a word of thanks to my fellow PhD students with whom I have collaborated over the past few years; I will miss the interesting and multilingual discussions we've had over coffee. Finally, I am grateful for the research sponsors that have supported my work over these past few years. I thank the NSF for awarding me the Graduate Student Fellowship, which allowed me to be comfortable enough to focus on research, not where the next meal would come from. Finally, a major thanks to the NA-MIC community for funding medical imaging work; in retrospect, much of my practical knowledge about implementing and debugging algorithms has been gained thanks to Ron Kikinis and the Slicer project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis is composed of 111 pages divided into five chapters. The scope of problems considered is defined in chapter I. Next, chapter II provides background material on image processing with partial differential equations and a review of prior work in the field. Chapter III covers the medical imaging portion of the research; the key contribution is a control-based algorithm for interactive image segmentation. Applications of the feedback-augmented level set method to fracture reconstruction and surgical planning are shown. Problems in vision-based control are considered in Chapters IV and V. A method of improving performance in closed-loop target tracking using level set segmentation is developed, with unmanned aerial vehicle or next-generation missile guidance being the primary applications of interest. Throughout this thesis, the two application types are connected into a unified viewpoint of open-loop systems that are augmented by exogenous data.

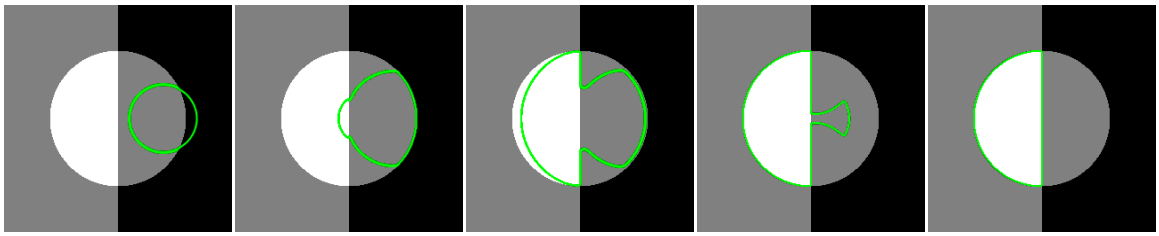
CHAPTER I

INTRODUCTION

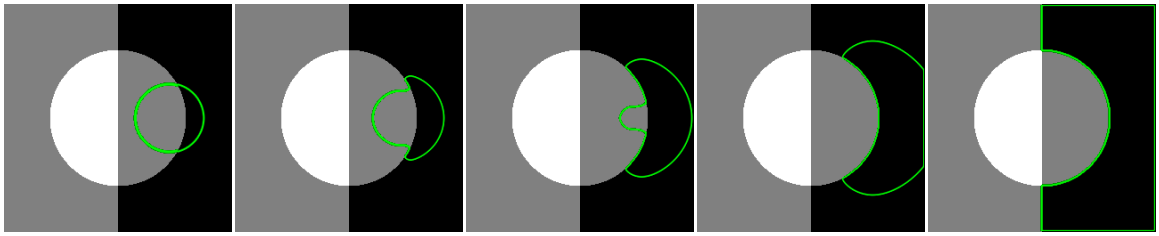
Many problems of practical interest in computer vision are formulated as the the process of finding a function ϕ^* which minimizes a functional $\mathcal{F}[\phi, I(\mathbf{x}, t)]$ dependent on image I defined on spatial domain Ω. I is considered to be either three-dimensional and fixed in time or alternatively two-dimensional and time-varying. Solving for ϕ involves iterative updates via $\partial\phi/\partial t$. This *time* t may correspond either to physical time in a dynamic setting (*e.g.* visual-servoing and vision-based-control) or simply a fictitious time when a static image segmentation is desired.

Modeling error in variational methods for image segmentation and tracking often emerges as a serious obstacle when we attempt to move from theory to practical computer-vision and image-processing applications. The choice of $\mathcal{F}$ is based on a simplified model of the real data. In reality, there are many local minima other than the "correct" ϕ^*, as exemplified by the two local minima in function space in Figure 1. Furthermore, this ideal ϕ^* may not even be a local minimum of $\mathcal{F}$. Nevertheless, variational methods are popular in image segmentation and tracking from both theoretic and pragmatic viewpoints; an appropriately designed $\mathcal{F}$ induces local minima which in practice are "close" (in L^2, for example) to ψ. Modeling error in the functional must simply be taken into consideration as a limiting factor on performance. An important observation to make is that many problems of this type actually contain a source of data other than the raw image I. In this work, exogenous data augments a partial differential equation (PDE) system describing the dynamics of ϕ.

The remainder of this thesis is organized as follows. Chapter 2 first covers the mathematical background of curve evolution and level set methods, then introduces the practical issues motivating the research. A PDE control formulation of interactive segmentation is presented in Chapter 3. Next, pose-invariant target tracking is developed in Chapter 4.



(a) Initialization at $x = 0.40, y = 0.00$



(b) Initialization at $x = 0.45, y = 0.00$

Figure 1: The boundary of the grey right half-circle is not a steady-state $\phi = 0$ level set for the open-loop dynamics of mean-alignment. Curves initially inside the grey half-circle tend to move into either the *(a)* bright or *(b)* dark adjacent regions.

This chapter assumes that region-based measurements are available and presents a control design to exploit our understanding of segmentation. In contrast, Chapter 5 considers techniques that modify the segmentation in order to generate measurements whose characteristics improve the net closed-loop tracking system performance.

CHAPTER II

BACKGROUND

Literature on algorithms that seek an iterative curve evolution solution for image segmentation and tracking are reviewed in Section 2.1. Two classes of applications have driven the preliminary research and serve to motivate the proposed research in segmentation and closed-loop tracking with exogenous data. First, interactive static image segmentation (ISIS) is reviewed in Section 2.2, with emphasis on volumetric medical imagery. Second, vision-based control (VBC) is reviewed in Section 2.3.

2.1 Variational Algorithms for Segmentation and Tracking

2.1.1 Curve Evolution Equations

This section reviews mathematical background on the basic curve evolution equations used in computer vision. [1] Consider the families of embedded closed curves $\mathcal{C} : S^1 \times [0, T) \to \mathbb{R}^2$ evolving according to functions of the curvature. Here, S^1 denotes the unit circle. More precisely, in vision problems, the general deformation of a curve in the plane of interest is given by

$$\frac{\partial \mathcal{C}}{\partial t} = \alpha(p,t)\mathcal{T} + \hat{\beta}(p,t)\mathcal{N} \tag{1}$$

where $\mathcal{N}$ is the unit (inward) normal, $\mathcal{T}$ is unit tangent, and α, $\hat{\beta}$ are smooth functions. Note that positive orientation of a curve is defined such that the interior is to the left when traversing the curve. The *curvature, orientation* and *length* are defined in the standard way. The normal is taken to point inward, where the inward or outward directions are determined by the interior, or equivalently by the orientation of the curve. Notice that, since only shape is interest, it is admissible to take $\alpha = 0$; changing α only changes the curve's parametrization and not its shape. Furthermore, as is typical in this area, the deformations

[1] For complete details, the interested reader is referred to the books [175, 176] that also have an extensive list of references.

are constrained to be determined by the local geometry of the curve, i.e., $\hat{\beta}(p,t) = \beta(\kappa)$ where $\kappa(p,t)$ denotes the Gaussian curvature of the curve $\mathcal{C}(p,t)$. The following basic equations are obtained for curve evolution under curvature driven flow:

$$\frac{\partial C}{\partial t} = \beta(\kappa)\mathcal{N}. \tag{2}$$

A key flow for nonlinear scale-space and active geometric contours is presented. This flow is achieved by setting $\beta(\kappa) = \kappa$ in (2). So, when $\beta(\kappa) = \kappa$, where κ is the curvature, and $\mathcal{N}$ the inward unit normal, a plane curve evolves according to the *geometric heat equation*

$$\frac{\partial \mathcal{C}}{\partial t} = \kappa\mathcal{N}. \tag{3}$$

This equation has a number of properties which make it very useful in image processing, and in particular, the basis of a nonlinear scale-space for shape representation [154]. Indeed, (3) is the Euclidean curve shortening flow in the sense that the Euclidean perimeter shrinks as quickly as possible when the curve evolves according to (3) [166–168]. Since segmentation and tracking depend so fundamentally on evolution of smooth curves and surfaces, the details are presented here. Let $\mathcal{C} = \mathcal{C}(p,t)$ be a smooth family of closed curves where t parameterizes the family and p the given curve, say $0 \leq p \leq 1$. (assume that $\mathcal{C}(0,t) = \mathcal{C}(1,t)$ and similarly for the first derivatives.) Consider the length functional

$$L(t) := \int_0^1 \|\frac{\partial \mathcal{C}}{\partial p}\| dp.$$

After differentiating and using integration by parts,

$$\begin{aligned} L'(t) &= \int_0^1 \frac{\langle \frac{\partial \mathcal{C}}{\partial p}, \frac{\partial^2 \mathcal{C}}{\partial p \partial t} \rangle}{\|\frac{\partial \mathcal{C}}{\partial p}\|} dp \\ &= -\int_0^1 \langle \frac{\partial \mathcal{C}}{\partial t}, \frac{1}{\|\frac{\partial \mathcal{C}}{\partial p}\|} \frac{\partial}{\partial p} \left[\frac{\frac{\partial \mathcal{C}}{\partial p}}{\|\frac{\partial \mathcal{C}}{\partial p}\|} \right] \|\frac{\partial \mathcal{C}}{\partial p}\| \rangle dp. \end{aligned}$$

(multiplying and dividing by $\|\frac{\partial \mathcal{C}}{\partial p}\|$ in the latter integral.) Observing that

$$\|\frac{\partial \mathcal{C}}{\partial p}\| dp =: ds$$

is (Euclidean) arc-length, and using the definition of curvature, the last integral becomes

$$-\int_0^{L(t)} \langle \frac{\partial \mathcal{C}}{\partial t}, \kappa\mathcal{N} \rangle ds$$

and

$$L'(t) = -\int_0^{L(t)} \langle \frac{\partial \mathcal{C}}{\partial t}, \kappa \mathcal{N} \rangle ds.$$

Using the Cauchy-Schwartz inequality,

$$\langle \frac{\partial \mathcal{C}}{\partial t}, \kappa \mathcal{N} \rangle \leq \|C_t\|_{L^2} \|\kappa \mathcal{N}\|_{L^2},$$

therefore the direction in which $L(t)$ is decreasing most rapidly is given by

$$\frac{\partial C}{\partial t} = \kappa \mathcal{N}.$$

Thus (3) is precisely a gradient flow. A much deeper fact is that simple closed curves converge to "round points" when evolving according to (3) without developing singularities [166, 167].

2.1.2 Level Set Representation

When applying curvature driven flows, as above, to images, it is necessary to move from curves to (gray-scale) images. Thus, it is essential to be able to relate curve evolution theory to operations on 2D images. Fortunately, the powerful algorithms developed by Osher and Sethian for curve evolution allow precisely this. In this section, a sketch is given of the beautiful work of Osher and Sethian on level curve evolutions. For more details, see [173, 174, 178].

Let $\mathcal{C}(p,t) : S^1 \times [0,T) \to \mathbb{R}^2$ be a family of curves satisfying the following evolution equation:

$$\frac{\partial \mathcal{C}}{\partial t} = \beta(\kappa)\mathcal{N}. \tag{4}$$

There are a number of problems which must be resolved when implementing curve evolution equations such as (4) numerically. For example, singularities may develop. Note for $\beta \equiv 1$ in (4) that even a smooth initial curve can develop singularities. The question is how to continue the evolution after the singularities appear. A natural way is to choose the solution which agrees with the Huygens principle [177, 179], or as Sethian observed, if the front is viewed as a burning flame, this solution is based on the principle that *once*

a particle is burnt, it stays burnt [177, 179]. As indicated above that, from all the *weak* solutions corresponding to (4), the one derived from the Huygens principle is *unique* and can be obtained via the entropy condition constraint.

In any numerical algorithm, the key requirements of accuracy and stability must be addressed. The numerical algorithm must approximate the evolution equation, and it must be robust. Osher and Sethian [173,174,177,178] showed that a simple, Lagrangian, difference approximation requires an impractically small time step in order to achieve stability. The basic problem with Lagrangian formulations is that the marker particles on the evolving curve can become unevenly distributed along the curve during the evolution. The algorithm proposed by Osher and Sethian provides a reliable numerical solution for curve (and hypersurface) evolution. It is based on the Hamilton-Jacobi equation and viscosity theory. First the curve is embedded in a two dimensional surface, and then, the equations of motion are solved using a combination of straightforward discretization and numerical techniques derived from hyperbolic conservation laws [181].

The embedding step is done in the following manner: the curve $\mathcal{C}(p,t)$ is represented by the zero level set of a smooth and Lipschitz continuous function $\Phi : \mathbb{R}^2 \times [0,\tau) \to \mathbf{R}$. Assume that Φ is negative in the interior and positive in the exterior of the zero level set. The zero level set defined by

$$\{X(t) \in \mathbb{R}^2 \,:\, \Phi(X,t) = 0\}\,. \tag{5}$$

It is necessary to find an evolution equation of Φ such that the evolving curve $\mathcal{C}(t)$ is given by the evolving zero level $X(t)$, i.e.,

$$\mathcal{C}(t) \equiv X(t)\,. \tag{6}$$

By differentiating $\Phi(X,t)$ with respect to t:

$$\nabla\Phi(X,t)\cdot X_t + \Phi_t(X,t) = 0\,. \tag{7}$$

Note that at the zero level, the following relation holds:

$$\frac{\nabla\Phi}{\parallel \nabla\Phi \parallel} = -\mathcal{N}\,. \tag{8}$$

In this equation, the left side uses terms of the surface Φ while the right side is related to the curve $\mathcal{C}$. Using equations (4) to (8),

$$\Phi_t = \beta(\kappa) \parallel \nabla\Phi \parallel \tag{9}$$

and the curve $\mathcal{C}$, evolving according to (4), remains equal to the zero level set of the function Φ, which evolves according to (9), throughout the evolution. Osher and Sethian [174] called this scheme an *Eulerian formulation* for front propagation because it is written in terms of a fixed coordinate system.

The second step of the algorithm consists of discretizing equation (9). If singularities cannot develop during the evolution, as in the geometric heat equation flow, a straightforward discretization can be performed [174]. If singularities can develop, as in the case of $\beta = 1$, a special discretization must be implemented. In this case, the implementation of the evolution of Φ is based on a *monotone* and *conservative* numerical algorithm derived from the theory of hyperbolic conservation laws [172, 174, 181]. For a large class of functions β of this type, this numerical scheme automatically obeys the entropy condition, i.e., the condition derived from Huygens principle [181]. For velocity functions of the form $\beta = a\kappa + 1$, a combination of both methods is used [173, 174, 178].

Of particular interest for handling hyperbolic systems is the *upwinding* discretization [80]. For example, with a 2nd-order upwinding scheme, a flux term $\frac{\partial}{\partial x}F(u) = f(u)u_x$ is discretized at a grid-coordinate x_i as

$$\begin{aligned}
\left.\frac{\partial}{\partial x}F(u)\right|_{x_i} &:= f^+ u_x^- + f^- u_x^+ \ , \\
f^+ &:= \max(f(u_i), 0) \ , \\
f^- &:= \min(f(u_i), 0) \ , \\
u_x^- &:= \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} \ , \\
u_x^+ &:= \frac{-u_{i+2} + 4u_{i+1} - 3u_i}{2\Delta x} \ .
\end{aligned}$$

Such an approximation for spatial derivatives enables sharp features in the curve to be

preserved while satisfying stability conditions; similar considerations arise in computational fluid dynamics [80]. It is important to note that the discretization of the evolution equations is performed on a fixed *rectangular grid*. This rectangular grid can be associated with the *pixel* grid of digital images making this discretization method natural for image processing.

2.1.3 Re-distancing and Regularization

Analytical computations and numerical implementations are easier to work with when ϕ is a signed distance function that satisfies $\|\nabla\phi\| = 1$ (almost everywhere). However, it is pointed out in [9] that a signed distance function cannot, in general, be a solution to Hamilton-Jacobi equations of the form Eq. 9. Given the practical usefulness of keeping $|\nabla\phi|$ bounded and nonzero, a number of methods have been proposed to reconcile level-sets and distance functions. [49] propose the addition of a constraint to Eq. 11 to make $|\nabla\phi| = 1$ almost everywhere. With similar goals, [81] do not constrain the PDE directly but instead augment the cost functional $E(\phi)$ with a regularization term:

$$\mathcal{E}(\phi) = E(\phi) + \mathcal{P}(\phi) \ , \quad \mathcal{P}(\phi) \doteq \frac{1}{2}\int_{\Omega} \left(|\nabla\phi| - 1\right)^2 d\mathbf{x} \ . \tag{10}$$

Computing the gradient flow for Eq. 10 yields a modified ϕ_t PDE. A subsequent work [82] by the same authors proposes an improved formulation of $\mathcal{P}(\phi)$ that, unlike Eq. 10, has negligible effect on the $\phi = 0$ level-set and regions where $|\phi|$ is much greater than 0.

2.1.4 Topics in the Recent Literature

The level set methodology of Osher and Sethian [109, 131] has become a standard in the field. This approach and its variants (fast-marching, narrow-banding, etc.) are ubiquitious in the literature and practice in computer vision and image processing. Several recent developments, by no means exhaustive, are as follows.

- **Shape Information** is typically available for target objects in a particular tracking or segmentation task. Suitable definitions of *shape* include a set of kernel-principal-components-analysis (KPCA) coefficients [35, 128], a three-dimensional mesh model [36], or a measure of silhouette deformation between frames [31, 93].

- **Statistical Models** are of paramount importance for practical image processing tasks, as an appropriate data representation is prerequisite for achieving any useful outcome [32]. A vast number of statistical models exist in the literature, wherein functionals are proposed that either maximize statistical differences between an object and its background or maximize similarity of the object to a template. Several examples of statistical quantities used are region-based feature means [23], feature covariance [122], and n-dimensional non-parametric density estimates [70, 97].

- **Coupled Level Sets** seek to use individual level set curves (or surfaces) to bound distinct regions in an image [19, 47, 98, 112]. Naturally, it is necessary to use a cost functional which prevents curves from overlapping and to ensure that no pixels are in none of the curves.

- **Dynamic Active Contours** [140] are a means of addressing the static nature of earlier snakes and geodesic active contours. Changes between image frames are kept bounded by incorporating a *kinetic energy* term. In [105,106], a geometric formulation using partial level sets attaches optical-flow data a curve that maintains image-state information in addition to penalizing data-independent $\|\phi_t\|_2$.

2.2 Interactive Static Segmentation of Volumetric Images

2.2.1 Background

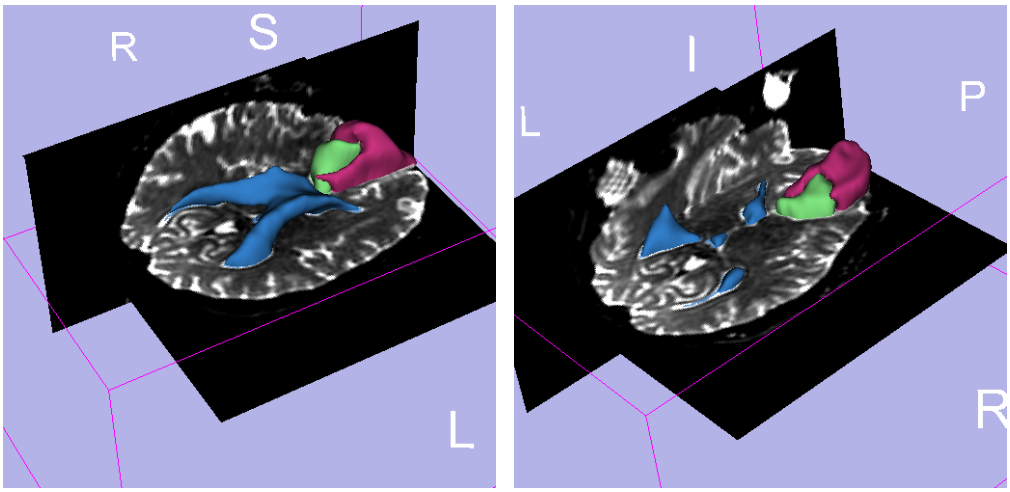
While statistical descriptors of anatomic regions and prior shape knowledge are powerful concepts towards increased automation of segmentation, they have not significantly displaced the expert human as the primary generator of volume segmentations. In addition to reasons of insufficient data for unique juvenile and trauma cases, there is a great deal of mistrust both from patients and doctors towards fully automatic medical analysis. Instead, there has been a recent drive towards semi-automatic image processing [52, 61, 149]. Ideally, such an interactive system should be designed as to enable a user to create excellent segmentation results with a minimal amount of time and effort.

Interactive segmentation as presented here is motivated by experiences with existing software tools, such as 3D-Slicer [119, 120], Seg3D [1], Spotlight [2], and ITK-SNAP [151]. To understand user behavior and design interaction accordingly, [101] simulate user interaction when evaluating segmentation performance. [91] compares a variety of interactive segmentation algorithms. In doing so, [91] notes that many semi-automatic segmentation tools demand that a user specify various parameters, whose name and purpose only makes sense in terms of the algorithm's mathematical derivation. Since intended users (medical students, technicians, residents, etc) do not have an in-depth mathematics background, the cryptic parameters tend to make medical practice wary of employing the latest image-processing algorithms. Consequently, the *impact* of image processing research is hindered.

Figure 2 depicts a typical application in medical volume segmentation. Typically, a user will first attempt to use automatic segmentation algorithm with partial success. They will then find a combination of manual editing and re-processing one small at a time to get a satisfactory result. The novel contribution in this paper is a modelling formulation that represents interactive segmentation as a feedback system, thus enabling a principled merging of automated methods and user input. Having this framework in place allows the tools of control theory to be invoked for system analysis and design.

2.2.2 Visualization Feedback

Visualization feedback to the human user is a vital link in interactive segmentation. Before any updates to $U(\mathbf{x}, t)$ can occur, the current state of $\phi(\mathbf{x}, t)$ must be displayed along with the image data. Visualization and interaction occur through a two-dimensional computer screen, limiting a human user's ability to see and interact with the evolving $\phi(\mathbf{x}, t)$. Besides increasing computational burden, the domain $\mathbf{x}$ being three-dimensional restricts the observability and controllability of an evolving PDE in interactive segmentation. Typically, editing is restricted to a particular 2D slice selected by the user. This domain restriction leads to the desirability of two qualitative behaviors in a segmentation system. First, the user would like to be able to see and easily interact with the largest possible sub-domain $D \subset \Omega$, as long as no significant delays are introduced in the visualization response to his



(a) Top View (b) Bottom View

Figure 2: There are two types of brain tumor tissue tightly interwoven (*green* and *magenta*). Segmentation enables us to accurately measure the volume and morphology of the tumor over time.

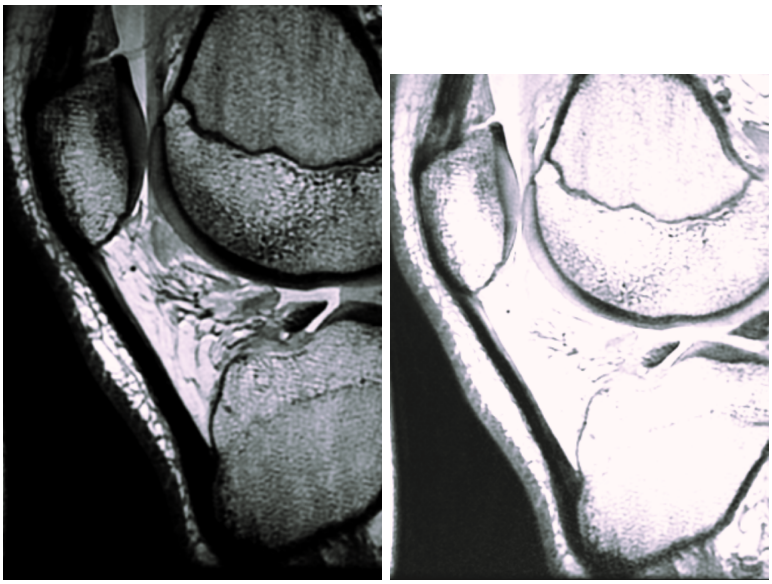
actions. Second, it is helpful if a PDE observer is able to form accurate estimates of the user input signal outside of D. Both of these design goals enable control of ϕ in a large domain with minimal effort from the human expert.

Even for an expert human user, it can be challenging to ascertain the "correct" segmentation. Enabling a user to determine the ideal boundary is achieved by a graphical user interface allowing contrast adjustment and rotation/translation of the volume to best see anatomic landmarks Figure 3. In this example, the user first wants to clearly see the patellar tendon in a saggital view (Figure 3a). Second, he traverses through the axial view to find the transition between trabecular and cortical bone structure (Figure 3b). He expects the patellar tendon to insert into the tibia and patella in the plane containing these transitions.

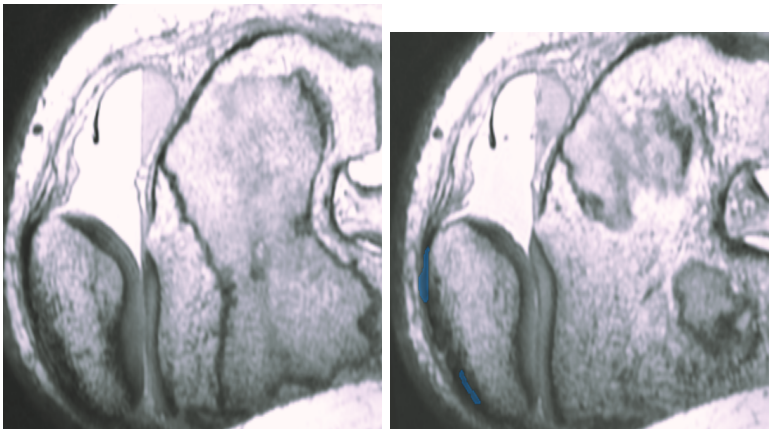
2.3 Tracking in Vision-Based Control

2.3.1 Overview

A variety of practical systems seek a segmentation which *tracks* a particular target while the underlying image data is changing dynamically. Furthermore, the system is to take



(a) Adjusting Contrast



(b) Transforming View

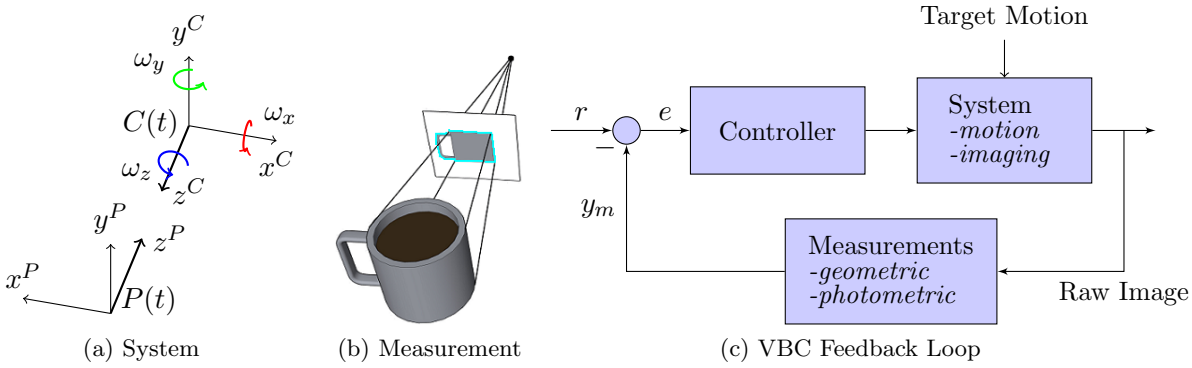
Figure 3: Users require visualization feedback to determine an ideal segmentation. A low-delay user interface with geometric and photometric transforms is required for effective interaction.

some action in response to the measurements of tracked regions: a closed-loop control is sought. Several such applications include:

- **Manufacturing Robotics** are a popular application of segmentation and tracking [24,29,30,152]. The nature medium to large-scale manufacturing enables fixed lighting and backgrounds, along with an excellent geometric and photometric understanding of how targets appear in imaging.

- **Autonomous Ground Vehicles** must deal with an open and uncontrolled environment. A major motivating push for progress has been the *Darpa Grand Challenge* [141], where teams of engineers and researchers modify a standard automobile to use algorithms and sensors to achieve some navigation task. Particularly difficult sub-problems that arise are how to predict and avoid collisions [147] and more generally how to formulate algorithms to operate in harsh urban conditions [71].

- **Unmanned Aerial Vehicles** require estimation of quantities such as relative range [144] and relative position amongst a group of UAV's [95]. Applications integrating computer vision results with flight control include analysis of flight stability [44] and autonomous landing [48, 133].



(a) System (b) Measurement (c) VBC Feedback Loop

Figure 4: Camera and target reference frames are denoted by $C(t)$ and $P(t)$ in Figure 21a, respectively. Projection of 3D to 2D is illustrated in Figure 21b. Vision-Based-Control feedback (Figure 4c) seeks to extract features from raw images and regulate state in response to unknown target motion.

Constructing a closed-loop VBC system requires an analysis of three related concepts which describe the system's dynamics and image-based measurements.

- Independent Euclidean motion of the camera and target. Interaction between these two motions describes the dynamics of 3D points in the camera frame.

- Pinhole camera projective image-formation, which projects 3D point coordinates to 2D image points. Disturbances due to unknown target motion become dependent on depth.

- Region-based segmentation, which provides target area and centroid measurements. Projective image-formation means that these quantities are biased by target pose, particularly in-place rotation.

Later, in Chapter 4, the above items are reviewed in detail.

2.3.2 Separation of Control and Estimation

Formulation of VBC systems requires both estimation of state from a large volume of data (image stream) and the selection of a control law which generates a twist ζ_C^C. This control may occur at discrete time intervals (such as incremental pan-tilt commands) or smoothly over time (such as thruster commands generating linear and angular acceleration). It is emphasized that these *estimation* and *control* sub-problems are extensively studied individually, but the manner in which they are combined is far less studied yet has a major impact on net system performance.

Unlike point-sensors (accelerometer, gyroscope, *etc*), an image stream's statistical characteristics can be dramatically affected by the control being applied, not merely the values of constituent pixels. For example, rapid motion will cause blurring and a change in the image gradient distribution. In the optics and low-level image processing literature, preprocessing or otherwise warping nominally acquired images to adjust for camera motion and other known phenomena are known as *image-stabilization* [62, 139]. Improved open-loop tracking and camera-calibration using fusion of accelerometer and gyroscope measurements has been explored [14, 55, 67, 94]. All of these methods are fundamentally *open-loop*; they either modify the measurement data or augment the imagery with sensor data for state-estimation.

In contrast, biological systems perform *closed-loop* fusion of physical sensors and imagery. For example, the *vestibulo-ocular reflex* [6, 15]. uses structures in the inner-ear to sense acceleration of the head and thus warp images and compensate gaze-tracking to enable higher-level vision processing. If one holds a driving license at arm's length, it will still be readable if the reader's head shakes rapidly, but not if the license moves with the same

relative velocity. As illustrated by this powerful feature of humans' integrated signal process and control, biological systems are an excellent source of motivation for vision-based-control. Indeed, our survival depends on vision-based control during the daily commute!

Studying insects' vision [104], mechanics [22], and neural system [117] has given some insight into how these "primitive" creatures dramatically outperform man-made systems in maneuverability, robustness, and low power-consumption. As biologists gain more understanding into insect anatomy, engineers seek to apply this knowledge to design bio-inspired robots [153]. There are also efforts to directly exploit insects' powerful low-level visual feedback systems. A radio receiver is integrated into the insect's nervous system in [129,130,142]. Radio control forces a navigation course while vision and flight stability are handled by the bug's sensory and neural systems.

Unlike in biological systems, current man-made VBC systems are designed in a way that separates "computer vision" from "feedback control". Image processing engineers seek to design open-loop visual tracking algorithms and test their efficacy primarily with pre-recorded video sequences. In turn, control engineers assume a black-box source of image-plane quantites like centroid, area, and possibly shape parameters. *Open-loop* visual tracking ignores the feedback strategy and considers only the visual processing. Tracking itself is an estimation procedure that generates a spatial description of the target within the image. This description is typically either a point or a closed region in the image.

The use of underlying computer vision techniques directly in *closed-loop* design is much less explored as compared to the multitude of open-loop tracking algorithms. Closed-loop visual tracking is inherently nonlinear, and thus necessitates design procedures from nonlinear control [69,73] when closing the loop. Visual-servoing is a *closed-loop* strategy that uses the image stream dynamics directly to perform a control task. Image formation, rigid body motion, and projective geometry are now well-understood [51,85,102], as are the significant difficulties in computing solutions to the related *inverse* problems.

A number of papers in the control field work towards the goal of linking control-theoretic

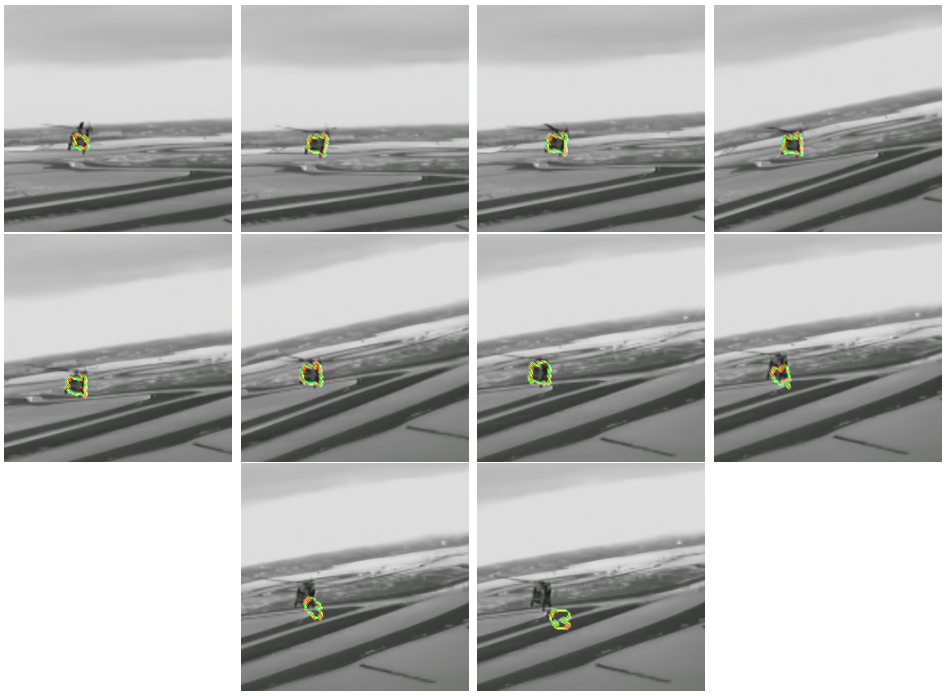


Figure 5: Tracking without using the available exogenous data in VBC feedback can cause loss of tracking during maneuvers.

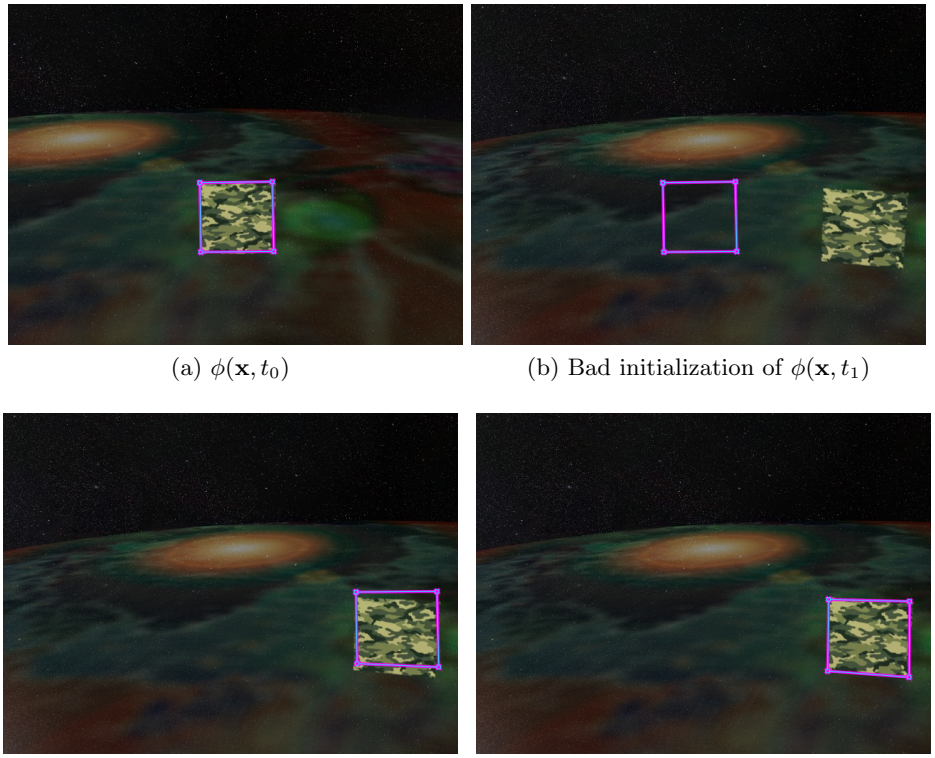
design to an understanding of geometric image formation, with a particular focus on homography decomposition and depth estimation from correspondence of projected points [25,37,38,57,58,77,87,99]. Feedback terms used in these works require as input a stream of solutions to inverse problems in computer vision, such as reliable feature-correspondence and homography-decomposition for rigid pose estimation. Linear solutions to these problems come from computer vision literature, where they are derived under assumptions of zero or highly structured measurement error. Accurate, numerically-stable geometric estimation requires nonlinear optimization and outlier-rejection schemes [51, 85, 138].

A motivating example for considering the use of exogenous data together with feedback VBC systems is illustrated in Figure 5. The rapid roll and yaw of the follower (to which the camera is attached) causes initialization of a tracking-contour to be erroneously placed on a background object of similar photometric properties; tracking subsequently fails as a result. While the previous work of [105, 106] addressed several *estimation* issues using dynamics of the segmenting contour, actually considering the *control* problem in conjunction with tracking is a direction that has not yet been rigorously analyzed. It is conjectured that the

control and estimation continue to remain separate in the vision-based-control literature due to the way in which computer-vision and control engineers interact. Tracking algorithms have been to-date almost exclusively tested and formulated based on pre-recorded video sequences, without access to control signals or other state measurements.

Figure 6 illustrates a beneficial way to exploit known control signals for augmented image processing. Suppose that a current segmentation is available, as in Figure 6a. A rapid camera motion can occur between the reception of images at times t_0 and t_1 and it is unlikely that a local gradient-based algorithm evolving ϕ would succeed at tracking given the initialization of Figure 6b. However, if the motion is purely rotational, we can simply integrate the nonlinear ODE describing motion of points in the image plane. That is, the domain[2] of $\phi(\mathbf{x}, t)$ is integrated for the duration $t_1 - t_0$ through the equations for $\dot{x}_c$ and $\dot{y}_c$. Doing so is straightforward since *we know the the control signals* $\{\omega_x, \omega_y, \omega_z\}$ applied during this time interval. Shown here are the results of considering only the effect of yaw in Figure 6c and of yaw, pitch, and roll as known in the simulation environment Figure 6d.

[2]or, for performance reasons, only the zero level set



(a) $\phi(\mathbf{x}, t_0)$ (b) Bad initialization of $\phi(\mathbf{x}, t_1)$

(c) Compensate $\phi(\mathbf{x}, t_1)$ for ω_y (d) Compensate $\phi(\mathbf{x}, t_1)$ for $\omega_x, \omega_y, \omega_z$

Figure 6: Integrating the twist-induced image coordinate derivatives using available angular velocity measurements can compensate for the platform's rapid motion. Image processing is then simplified and/or accelerated.

CHAPTER III

INTERACTIVE STATIC IMAGE SEGMENTATION

3.1 Introduction

Microwave-frequency Magnetic-Resonance-Imaging (MRI) [34] and X-Ray Computed Tomography (CT) [56] yield three-dimensional volumetric images which are viewed by a medical professional for diagnosis, treatment planning, or population studies [43,114]. Typically, only a particular anatomic region or organ is of interest and must be *segmented*. Segmentation is the task of identifying and localizing salient structures in the image volume. Since there is an abundance of raw image data that is not analyzed due to the infeasible time and cost of manual segmentation, automated methods for segmentation are the subject of much recent medical computing literature [8,39,116]. However, a human expert's ability to combine observed image data with prior anatomical knowledge to accurately perform segmentation is unmatched by computer algorithms. There is substantial mistrust both from patients and doctors towards fully automatic algorithms. Recognizing this, there has been a recent drive towards interactive segmentation [91, 92, 108]. Interactive approaches use a data-driven automatic algorithm to process a majority of the volume. As the automatic segmentation runs and displays the current state, a human user can influence the algorithm's behavior to more closely align with an expected result. Figure 7 shows an example of the segmentation corrections a human would make. Ideally, an interactive system should enable a user to create excellent segmentation results with a minimal amount of time and effort.

Interactive medical image segmentation employs software tools such as SEG3D [1] and 3D-Slicer [119] for applying algorithms and visualizing the results; a human expert uses the algorithms to achieve a segmentation that is as close as possible to the ideal region boundary. Available algorithms include iterative methods with a concept of *time*; a partial

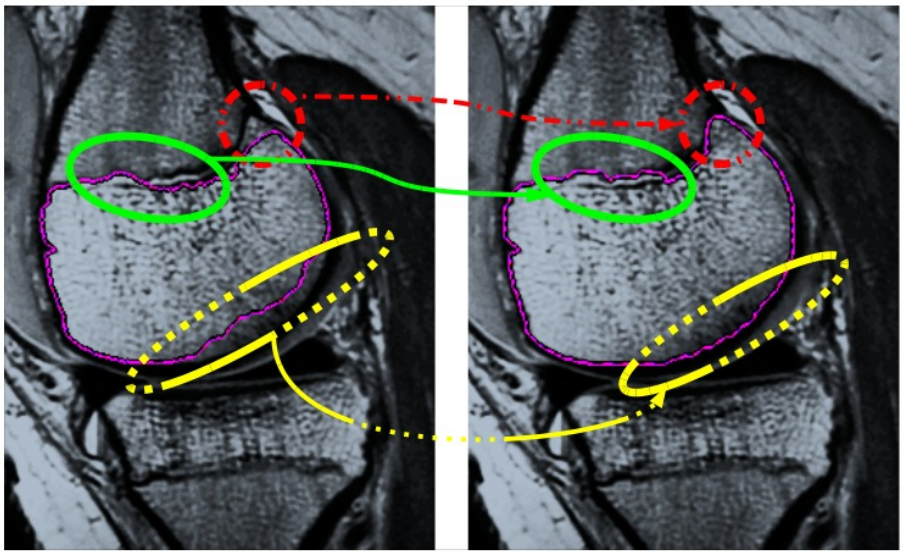


Figure 7: Segmentation by minimizing a meaningful image-dependent functional is not sufficient when the desired anatomic boundary is not actually a minimizer (*left*). An expert user would typically desire to make some corrections (*right*) that contradict the functional's minimizer.

differential equation (PDE) is used to model the space-time relationship between image data and segmentation boundaries [23, 31, 79, 112]. Given user-specified parameters and initial state, incremental modifications are made to the segmentation and shown to the user [12,110,125,148]. It is not clear to a user whether it is possible for their ideal region boundary to be a steady-state. In practice, they must stop the algorithm at some time t_f when the segmentation is reasonably accurate, then apply smoothing and manual corrections to the boundary. An alternative approach is to formulate segmentation as a time-independent problem; in [52, 100, 149], user input acts as a constraint in finite-dimensional nonlinear optimization problems. It is not known a-priori whether the user's ideal region boundary is a feasible solution for some collection of constraints, while a changing number of user input constraints can affect the computational complexity. Furthermore, the time-independent formulation can lead to large changes in the segmentation output when new user input is received. Other classes of algorithms such as graph-cuts [16, 17, 125] are also effective for automated segmentation; we consider only level-set PDE algorithms due to their theoretical compatibility with methods in the PDE control literature.

Level-set methods define a region boundary implicitly as the zero level-set of a function

$\phi(\mathbf{x}, t)$ with domain[1] $\Omega \subset \mathbb{R}^n$. Temporal changes of the region boundary are described by a partial differential equation (PDE) as a consequence of the implicit representation. Typically $\frac{\partial \phi}{\partial t}$ arises as the gradient flow that minimizes some meaningful functional of ϕ and the image data $I(\mathbf{x})$. From a controls perspective, the image-dependent PDE is an open-loop system. We present a framework for interactive segmentation using feedback augmentation of a level set PDE system; the results and theory are a substantial extension of the preliminary version [64]. This paper is motivated by the following observation: when human users influence the level-set evolution, *they have in mind a desired reference state and are trying to apply control to an image-dependent PDE system.*

In the literature on control of PDE systems, two characteristics of problems are the domain of input actuation and the available measurement (pointwise throughout a domain, boundary-value only, or as an integral over space). Control through region boundaries is of paramount interest; [42] characterizes the stabilizing controls and admissible boundary conditions for a class of unstable reaction-diffusion (R-D) systems. Similarly, [74] explicitly computes invariant regions for coupled R-D systems. Stabilization of the viscous Burgers equation $u_t + \frac{d}{dx}(\frac{1}{2}u^2) = \epsilon u_{xx}$ using boundary actuation is achieved in [86, 135] by first designing a feedback law using $u(x, t)$ over all x, then deriving a u-observer that uses only boundary measurements. In [13], the inviscid $u_t + \frac{d}{dx}(\frac{1}{2}u^2) = 0$ is stabilized with a boundary input $u(0, t)$ from an admissible set of controls that admits a weak solution to the initial-boundary-value problem. A common theme in PDE-control for setting input and gain values are scalar functions $w(t)$ defined as functionals of the state $u(x, t)$ [75, 76, 143]; *e.g.*, $w(t) = \int_\Omega k(x, t)u(x, t)dx$. Such inputs appear in this paper as well, with image dependence entering via a term analogous to $k(x, t)$. The model used in this paper uses actuation and measurement within a neighborhood of the time-varying segmentation boundary.

Contribution: Using a PDE formulation guarantees that the computational complexity is fixed and that the segmentation result changes continuously over time. By incorporating control-theoretic tools, it is shown that the steady state segmentation can be driven

[1]The dimension is $n = 3$ or $n = 2$, for volumetric images and planar images, respectively.

to an ideal reference boundary. Input from the human user indicating locations where the current state does not match the desired reference state is processed and used as feedback in the PDE. An approach similar to backstepping [69, 75] is used; first, stability of a labeling error functional is shown under the assumption of a known reference state. Second, an auxiliary observer-like system that reacts to user input is formulated. The net coupled system is shown to have bounded error when a sufficient amount of user input has been accumulated. To the best of our knowledge, this is the first approach to interactive level set segmentation with input from the user used in feedback to guarantee stabilization about a reference boundary.

Organization: The remainder of this paper is as follows. Image segmentation based on the narrow-band level-set method is reviewed in Section 3.2. Following an approach similar to back-stepping, a controller is proposed in Section 3.3 that stabilizes a labeling-error term, assuming exact knowledge of a reference state. An auxiliary observer-like system is designed in Section 3.4 to process a user input signal and estimate the reference state. Application of the technique to interactive segmentation of CT and MRI volumes is demonstrated in Section 3.5, using images that are difficult to segment with existing methods. Section 4.7 summarizes the results and applicability of the approach. Key components of the final closed-loop system and corresponding paper sections are visualized in Figure 8.

3.2 Level Sets and Automated Segmentation

The class of open-loop systems considered in this work are PDE-based segmentation algorithms using the popular *level set method*, reviewed in §3.2.1. §3.2.2 describes the limitations of open-loop segmentation, thus motivating the feedback control model of interactive image segmentation in § 3.3.1.

3.2.1 Review of Level Set Methods

Level set methods represent time-varying region boundaries in a computationally straightforward manner [109, 131]. Define $\Omega \subset \mathbb{R}^n$ to be the spatial domain and $\mathbf{x}$ a coordinate in Ω. Labeling assignments are represented with an implicit function $\phi(\mathbf{x}, t) : \Omega \times [0, t) \rightarrow \mathbb{R}$.

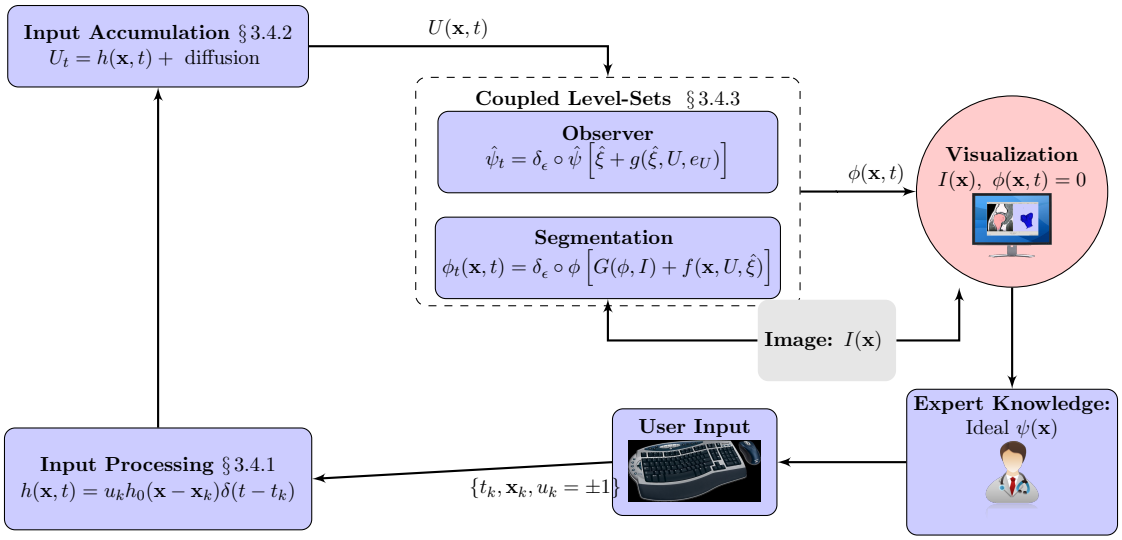


Figure 8: Block diagram of the proposed control formulation. Feedback compensates for deficiencies in automatic segmentation by exploiting the human expert's interpretation of complex imagery.

Boundaries between regions of interest are represented as level sets where $\phi(\mathbf{x}, t) = C$. Propagation of ϕ over time is defined by $\phi_t = -\langle \nabla\phi, \mathbf{f} \rangle$ for some vector field $\mathbf{f}$ that is a function of image data, of ϕ, and spatial derivatives of ϕ. In this paper, $\phi(\mathbf{x}, t) > 0$ denotes the interior of a segmented region and the (outward) normal vector $\mathbf{N}$ along a level set of ϕ is $\mathbf{N} = -\frac{\nabla\phi}{|\nabla\phi|}$. Assuming the gradient and normal of ϕ exist, the general form of a level set PDE is

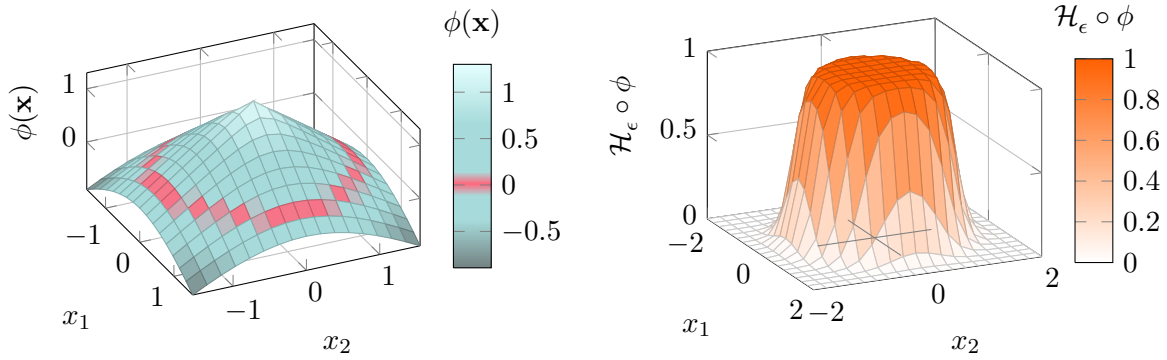
$$\phi_t = |\nabla\phi| \langle \mathbf{N}, \mathbf{f} \rangle \doteq F|\nabla\phi| \ . \tag{11}$$

In "variational level set methods" [11,32,126], the F in Eq. 11 is constructed to regulate some quantity of the form

$$E(t) = \int_\Omega g\big(\mathbf{x}, \phi, \nabla\phi\big) d\mathbf{x} \ . \tag{12}$$

As pointed out in [105], many image segmentation applications use an *artificial* time parameter t, which arises solely due to an iterative minimization of Eq. 12. In this paper, however, t corresponds to a physical time since the human user watches a time-varying visualization of $\phi(\mathbf{x}, t)$ to decide where and when to provide input.

It is usually desirable to have $|\phi_t| > 0$ only in a neighborhood of the moving zero level set [49, 81, 82]. This *narrowband* restriction is used in the image processing community for



(a) $\phi(\mathbf{x})$ with 0 level set in red.

(b) Regularized Heaviside function $\mathcal{H}_\epsilon \circ \phi(\mathbf{x})$.

Figure 9: The interior of a segmented region satisfies $\phi(\mathbf{x}) > 0$ while its exterior has $\phi(\mathbf{x}) < 0$.

several reasons. First, efficient numerical techniques such as the "sparse field method" [146] update ϕ only within the narrow-band region. Dimensions of 3D medical image volumes are on the order of 512^3; real-time performance on desktop computers is attainable only by restricting computations to a subset of Ω. Second, algorithms typically seek to separate Ω into statistically different regions within the $I(\mathbf{x})$ image. It is sufficient to know only $\text{sign}(\phi)$ when labeling regions as interior and exterior; the magnitude of ϕ has little meaning in segmentation applications. Finally, if ϕ_t is nonzero for arbitrary values of $|\phi|$, zero crossings can develop far from the initial $\phi = 0$ level-set; a visualization of ϕ would show new "boundaries" spontaneously appearing. Users will be confused by such behavior; they expect to initialize $\phi(\mathbf{x}, 0)$ and watch a moving level set.

Ω is divided into *exterior* and *interior* regions by a regularized version of the Heaviside step function denoted by $\mathcal{H}_\epsilon$, illustrated in Figure 9. This regularized step function and its derivative δ_ϵ are defined as

$$\mathcal{H}_\epsilon \circ \phi = \begin{cases} 1 & \text{if} \quad \phi > \epsilon \\ 0 & \text{if} \quad \phi < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\epsilon})\right) & \text{otherwise} \end{cases} \tag{13}$$

$$\delta_\epsilon \circ \phi = \begin{cases} 1/\epsilon & \text{if} \quad \phi = 0 \\ 0 & \text{if} \quad |\phi| > \epsilon \\ \frac{1}{2\epsilon}\left(1 + \cos(\frac{\pi\phi}{\epsilon})\right) & \text{otherwise} \end{cases} \tag{14}$$

This paper considers systems described by narrowband level-set PDEs with the general form

$$\begin{aligned} \phi_t =& \delta_\epsilon \circ \phi \left[G(\mathcal{H}_\epsilon \circ \phi, I) + \lambda \nabla \bullet \frac{\nabla \phi}{\|\nabla \phi\|_2} \right] , \\ \phi(\mathbf{x}, 0) =& \phi_0(\mathbf{x}) \ . \end{aligned} \tag{15}$$

Application-specific goals, such as minimization of a functional, dictate the choice of image-dependent $G(\bullet)$ in Eq. 15; an example is given in the next section. A concise notation for the elliptic operator $\nabla \bullet \frac{\nabla F}{\|\nabla F\|_2}$ is $\kappa(F)$, where F is a smooth function with non-vanishing gradient defined on $\mathbf{x} \in \Omega$.

In practice, the nonlinear PDEs encountered in level-set segmentation will tend to develop discontinuities. Periodic reinitialization of ϕ to a signed distance function [109, 127] mitigates these effects while preserving the $\phi = 0$ level set. Re-distancing enforces the properties

$$0 < p_1 < |\nabla \phi| < p_2 \ , \quad -\kappa_0 < \kappa(\phi) < \kappa_0 \ , \tag{16}$$

which are helpful for control synthesis in Section 3.3.

3.2.2 Segmentation as an Open-Loop System

Automated image segmentation systems are designed under the assumption that a particular discriminative model captures distinguishing features that separate regions of interest from the rest of the image volume. Consequently, the systems often lead to erroneous segmentation results when their underlying assumptions do not hold. The term "open-loop" in the context of automatic segmentation means that the system evolves without any external input that might indicate failure of model assumptions and that the boundary is not moving towards a desired steady-state. Such systems arise from discriminative statistical models in the literature, wherein functionals are proposed that either maximize statistical differences between an object and its background or maximize similarity of the object to a template [32]. Several examples of statistical quantities used are region-based feature means [23], feature covariance [122], and n-dimensional non-parametric density estimates [70, 97].

Nevertheless, a recurring problem is that many objects of interest do not coincide with

minima of a proposed functional. As an example, consider segmentation via the "mean-alignment" system [23]. The weighted means of image $I(\mathbf{x})$ in the interior ($\phi > 0$) and exterior ($\phi < 0$) regions are

$$\mu_{in} \doteq \frac{\int_\Omega [\mathcal{H}_\epsilon \circ \phi] I \; d\mathbf{x}}{\int_\Omega [\mathcal{H}_\epsilon \circ \phi] \; d\mathbf{x}} \; , \quad \mu_{out} \doteq \frac{\int_\Omega [1 - \mathcal{H}_\epsilon \circ \phi] I \; d\mathbf{x}}{\int_\Omega [1 - \mathcal{H}_\epsilon \circ \phi] \; d\mathbf{x}} \; . \tag{17}$$

A gradient flow for the functional

$$E(t) = \frac{1}{2} \int_\Omega \mathcal{H}_\epsilon \circ \phi (I - \mu_{\text{in}})^2 + (1 - \mathcal{H}_\epsilon \circ \phi)(I - \mu_{\text{out}})^2 d\mathbf{x} + \lambda \int_\Omega |\nabla \mathcal{H}_\epsilon \circ \phi| d\mathbf{x} \tag{18}$$

gives the following narrowband open-loop system:

$$\begin{aligned} \phi_t =& \delta_\epsilon \circ \phi \left[-(I - \mu_{\text{in}})^2 + (I - \mu_{\text{out}})^2 + \lambda \kappa(\phi) \right] \; , \\ \phi(\mathbf{x}, 0) =& \phi_0(\mathbf{x}) \; . \end{aligned} \tag{19}$$

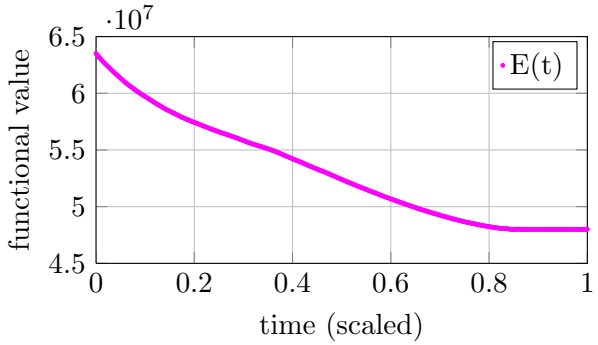
Figure 10 illustrates the behavior of the open-loop system Eq. 19 on a synthetic image that resembles a noisy image of the left and right ventricles in the brain. The desired segmentation boundary is drawn in dashed red, while the moving zero level set is solid green. Although the open-loop system correctly stops along most of the reference boundary, it creeps into an adjacent region of $I(\mathbf{x})$ in the course of minimizing $E(t)$.

3.3 Feedback Augmentation of a Narrowband Levelset PDE

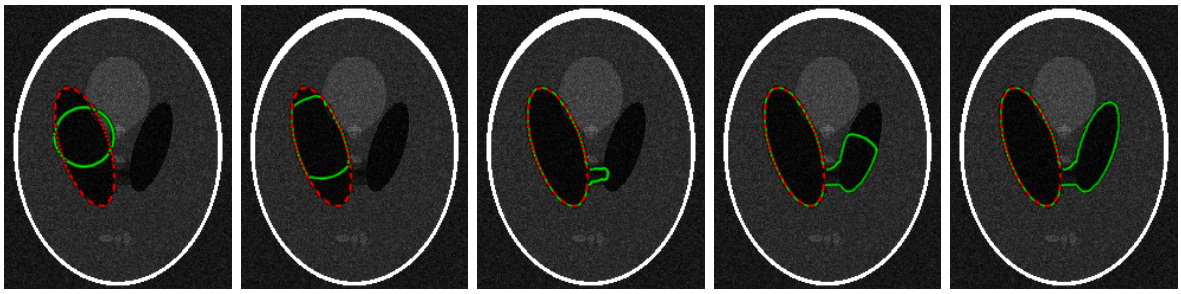
3.3.1 Reference State and Input Structure

Rather than having the human user give up on the PDE system and manually outline the desired region of interest, the PDE can be augmented with a user-driven control input. A control solution is sought due to limitations in the efficacy of open-loop system Eq. 15 for real images. Necessary human effort in segmentation can then be kept low; the user need not apply input in locations where the open-loop system keeps ϕ in agreement with a desired segmentation.

Let ψ denote the ideal reference segmentation. A human user could manually trace the level set $\psi(\mathbf{x}) = 0$ if given unlimited time. We seek to drive ϕ towards an explicit estimate of ψ, while maintaining closed-loop stability and minimizing burden placed on the user.



(a) $E(t)$ functional (Eq. 18) regulated by the open-loop system.



(b) Desired segmentation and $\{\phi(\mathbf{x}, t) = 0\}$ level-set overlayed on image $I(\mathbf{x})$.

Figure 10: Open-loop segmentation example using image means. A user would like to segment the "left ventricle" in this synthetic image that resembles an MRI scan of the brain. Evolving ϕ according to Eq. 19 shrinks $E(t)$ successfully (*top*). However, the open-loop system fails to segment the desired region (*bottom*).

User-driven control effort should preserve the advantages of the narrowband formulation noted in § 3.2.1; therefore, an admissible control signal $f(\mathbf{x}, t)$ will act only on the $|\phi| \leq \epsilon$ subdomain of Ω. The closed-loop system then becomes

$$\phi_t = \delta_\epsilon \circ \phi \left[G(\phi, I) + \lambda\kappa(\phi) + f(\mathbf{x}, t) \right] \ . \tag{20}$$

In this section, $\psi(\mathbf{x})$ is assumed known; a control is synthesized to drive ϕ such that it matches ψ. Later in Section 3.4, a coupled system that estimates ψ from available user input is formulated.

3.3.2 Existence of a Regulatory Control

Define the pointwise and total labeling error as ξ and $\mathcal{D}$, respectively:

$$\xi(\mathbf{x}, t) \doteq \left(\mathcal{H}_\epsilon \circ \phi - \mathcal{H}_\epsilon \circ \psi \right) \ , \quad \mathcal{D}(t) \doteq \frac{1}{2} \int_\Omega \xi^2 d\mathbf{x} \ . \tag{21}$$

If ψ is known and available, regulation of $\mathcal{D}(t)$ is straightforward with the following two theorems in this section. The regulatory control uses known bounds on the image-dependent G term of Eq. 20; define $G_{\text{M}}(\mathbf{x}), \bar{G_{\text{M}}}$ to be upper bounds such that for any segmentation state ϕ,

$$|G\left(\phi(\mathbf{x}), I(\mathbf{x})\right)| \leq G_{\text{M}}(\mathbf{x}) \leq \bar{G_{\text{M}}} \ . \tag{22}$$

Theorem 3.3.1 *Using a spatially-varying $U(\mathbf{x})$, a control for Eq. 20 that stabilizes functional Eq. 113 is*

$$f(\mathbf{x}, t) = -\xi U^2 + \lambda(t)\left[\kappa\left(\delta_\epsilon^2 \circ \phi \cdot \xi\right) - \kappa\left(\phi\right)\right] \ . \tag{23}$$

Given constants $\lambda_0 > 0,\ \lambda_1 > 0,\ \rho \in (0, 1)$, a sufficient condition for boundedness of $\mathcal{D}$ is

$$\lambda(t) = \lambda_0 + \lambda_1 \mathcal{D} \ , \ \textit{and} \ \ \left|U(\mathbf{x})\right| \geq \left(\frac{G_{\text{M}}(\mathbf{x})}{\rho} + 1\right)^{1/2} \doteq U_{\text{M}} \ . \tag{24}$$

Furthermore, when the error ξ is large in the sense of

$$\rho \int_\Omega \delta_\epsilon^2 \circ \phi \ |\xi| dx \leq \int_\Omega \delta_\epsilon^2 \circ \phi \ \xi^2 dx \ , \tag{25}$$

the rate of convergence is bounded by

$$\frac{d}{dt}\mathcal{D} \leq -\int_\Omega \delta_\epsilon^2 \circ \phi \ \xi^2 d\mathbf{x} \leq 0 \ . \tag{26}$$

Proof Re-arranging terms in $\mathcal{D}'$ and integrating by parts making use of $\delta_\epsilon \circ \phi = 0$ on $\partial\Omega$:

$$\frac{d}{dt}\mathcal{D} = \int_\Omega \xi\dot{\xi} dx \ = \int_\Omega \xi \delta_\epsilon \circ \phi \cdot \phi_t dx \tag{27}$$

$$= \int_\Omega \delta_\epsilon^2 \circ \phi \left[\xi G(I, \phi) - (\xi U)^2\right] dx + \lambda \int_\Omega \delta_\epsilon^2 \circ \phi \ \xi \ \nabla \cdot \frac{\nabla \delta_\epsilon^2 \circ \phi \ \xi}{\|\nabla \delta_\epsilon^2 \circ \phi \ \xi\|_2} dx \tag{28}$$

$$= \int_\Omega \delta_\epsilon^2 \circ \phi \left[\xi G(I, \phi) - (\xi U)^2\right] dx - \lambda \int_\Omega \nabla(\delta_\epsilon^2 \circ \phi \ \xi) \cdot \frac{\nabla(\delta_\epsilon^2 \circ \phi \ \xi)}{\|\nabla(\delta_\epsilon^2 \circ \phi \ \xi)\|_2} dx \tag{29}$$

$$= \int_\Omega \delta_\epsilon^2 \circ \phi \ \xi G(I, \phi) dx - \int_\Omega \delta_\epsilon^2 \circ \phi \ (\xi U)^2 dx - \lambda \int_\Omega |\nabla(\delta_\epsilon^2 \circ \phi \ \xi)| dx \ . \tag{30}$$

The Poincaré inequality in L^1 guarantees the existence of a constant r such that

$$\int_\Omega |\delta_\epsilon^2 \circ \phi \ \xi| dx \leq r \int_\Omega |\nabla(\delta_\epsilon^2 \circ \phi \ \xi)| dx, \tag{31}$$

$$-\int_\Omega |\nabla(\delta_\epsilon^2 \circ \phi \ \xi)| dx \leq -\frac{1}{r} \int_\Omega \delta_\epsilon^2 \circ \phi |\xi| dx \ , \tag{32}$$

where r is at most half the diameter of Ω [3]. Substituting Eq. 32 into Eq. 30 bounds the Lyapunov functional's time derivative:

$$\frac{d}{dt}\mathcal{D} \leq \int_\Omega \delta_\epsilon^2 \circ \phi \left[\xi G(I,\phi) - (\xi U)^2\right] dx - \frac{\lambda}{r} \int_\Omega \delta_\epsilon^2 \circ \phi |\xi| dx \ . \tag{33}$$

The case of $|\xi|$ being large relative to ρ in an integral sense (Eq. 25) also implies

$$\int_\Omega \delta_\epsilon^2 \circ \phi |\xi G| dx \leq \int_\Omega \delta_\epsilon^2 \circ \phi |\xi| \frac{|\xi|}{\rho} G_{\text{M}} dx \ . \tag{34}$$

Substituting the condition on $|U(\mathbf{x})|$ magnitude from Theorem 3.3.1 gives

$$\int_\Omega \delta_\epsilon^2 \circ \phi |\xi| \frac{|\xi|}{\rho} G_{\text{M}} dx \ < \ \int_\Omega \delta_\epsilon^2 \circ \phi \ \xi^2 \left(\frac{G_{\text{M}}}{\rho} + 1\right) dx \ \leq \ \int_\Omega \delta_\epsilon^2 \circ \phi \ (\xi U)^2 dx \ , \tag{35}$$

and the error rate $\mathcal{D}'$ is negative semidefinite with a bound

$$\frac{d}{dt}\mathcal{D} \leq -\int_\Omega \delta_\epsilon^2 \circ \phi \ \xi^2 dx \leq 0 \ . \tag{36}$$

Boundedness of $\mathcal{D}$ is established after substituting the λ proposed in Theorem 3.3.1:

$$\mathcal{D}' \leq \int_\Omega \delta_\epsilon^2 \circ \phi \ \xi G dx - \frac{\lambda}{r} \int_\Omega \delta_\epsilon^2 \circ \phi \ |\xi| dx \tag{37}$$

$$\leq \bar{G}_{\text{M}} \int_\Omega \delta_\epsilon^2 \circ \phi \ |\xi| dx - \frac{1}{r} \left(\lambda_0 + \lambda_1 \mathcal{D}\right) \int_\Omega \delta_\epsilon^2 \circ \phi \ |\xi| dx \tag{38}$$

$$\leq \left(\bar{G}_{\text{M}} - (\lambda_0 + \lambda_1 \mathcal{D})/r\right) \int_\Omega \delta_\epsilon^2 \circ \phi \ |\xi| dx \ . \tag{39}$$

$\mathcal{D}' > 0$ is only possible for $\mathcal{D} < \left(r\bar{G}_{\text{M}} - \lambda_0\right)/\lambda_1$, with $\mathcal{D}' \leq 0$ otherwise. Thus, $\mathcal{D}$ is bounded.

Remark A near-optimal (i.e. low) value for r can be obtained by substituting the definitions of $\delta_\epsilon, \mathcal{H}_\epsilon$ (eqns. 14,13) into $|\nabla(\delta_\epsilon^2 \circ \phi \ \xi)|$, applying the chain rule, and comparing to $|\delta_\epsilon^2 \circ \phi \ \xi|$ (omitted for space). In practice, r can be directly estimated via numerical evaluation of the integrals in Eq. 31 and is on the order of the $|\nabla\phi|$ due to re-distancing.

3.4 Auxiliary System Design

In the previous section, a stabilizing controller was developed that drives the segmentation towards a reference state relying on fixed quantities ψ and U, which are not known in practice. The user will be employed to provide the missing information by occasionally applying discrete corrections to the segmentation; these corrections are accumulated over time. From the current segmentation and user input, an estimate of the ideal ψ must be inferred. These considerations lead to the coupled dynamical system presented in this section. A method for processing discrete input from a mouse or stylus to generate a distributed U is proposed in § 3.4.1, while the accumulation of input is addressed in § 3.4.2. An observer-like system is formulated in § 3.4.3 to compute $\hat{\psi}$, the explicit estimate of ideal state ψ.

3.4.1 User Input Processing

Raw input from the user arrives in the form of binary decisions as to whether a given location in space is correctly labeled as inside or outside the segmentation boundary. The user clicks with a mouse or stylus at discrete points in Ω and time, as illustrated in Figure 11. Define $t_k,\ k \in \mathbb{N}$ to be the sequence of times at which the user sees a visualization of ϕ and has an opportunity to apply input. At time t_k, they look at the labeling of $\phi(\mathbf{x}_k, t_k)$ and either *(a)* apply a signed impulse denoting a "vote" for setting the label there or *(b)* do nothing because they agree with the current labeling of $\phi(\mathbf{x}_k, t_k)$. Denote these sequential actions as

$$u_k = \begin{cases} +1 & \text{if} \quad \psi(\mathbf{x}_k) > 0 > \phi(\mathbf{x}_k, t_k) \ , \\ -1 & \text{if} \quad \psi(\mathbf{x}_k) < 0 < \phi(\mathbf{x}_k, t_k) \ , \\ 0 & \quad \text{otherwise} \ . \end{cases} \tag{40}$$

Before these inputs can be accumulated into U, they must be mapped into the space-time domain with some fixed support. Define the function $h(\mathbf{x}, t)$ as

$$h(\mathbf{x}, t) \doteq \sum_k u_k h_0(\mathbf{x} - \mathbf{x}_k)\delta(t - t_k) \ , \tag{41}$$

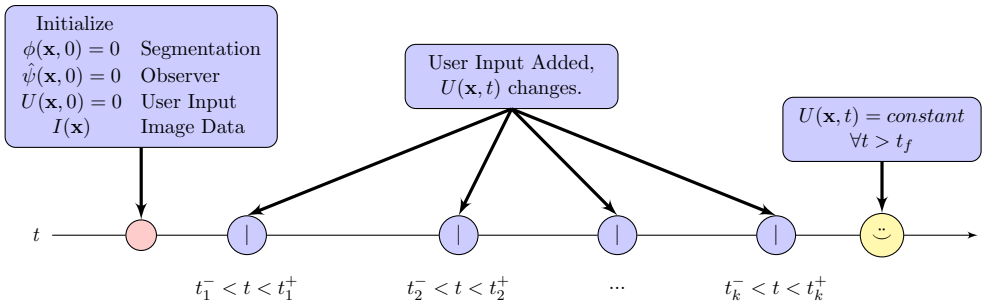


Figure 11: After initialization, the inner loop of Figure 8 updates ϕ and ψ. Input from a human user applies impulses at times t_k that accumulate as $U(\mathbf{x},t)$. Between times t_k, the inner loop changes steady-state in response to updated $U(\mathbf{x},t)$. The user stops applying input when the visualization of $\phi(\mathbf{x},t)$ is satisfactory.

where $h_0(\cdot)$ is a weight function and $\delta(t-t_k)$ is the Dirac delta. As noted in [100], using an image-dependent metric for $h_0(\cdot)$ is a useful way to weight spatial distances. The examples in this paper use

$$h_0(\cdot) \doteq \left(\frac{\sigma_I^2}{\sigma_I^2 + |I(\mathbf{x}) - I(\mathbf{x}_k)|}\right) \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}_k\|_2^2}{\sigma_{\mathbf{x}}^2}\right), \tag{42}$$

which incorporates both Euclidean distance from $\mathbf{x}$ to $\mathbf{x}_k$ and similarity between image values at $I(\mathbf{x})$ and $I(\mathbf{x}_k)$.

3.4.2 Accumulation of User Input

The label error impulse inputs accumulate over time to define the control U. However, U must be regulated to prevent excessive input magnitudes while ensuring spatial smoothness and enabling $|U| \geq U_{\text{M}}$ to satisfy the conditions of §3.3.2. An undesirable excess U and $|\nabla U|$ can occur in $U_t = h(\mathbf{x},t)$ because the human user causes $h(\mathbf{x},t)$ without understanding how their "vote" inputs influence the segmentation dynamics. Furthermore, when the label-error is shrinking at a consistent rate but over a large area, it is expected that the human user will be impatient and apply excess input magnitude in an attempt to speed up the moving $\phi = 0$ boundary.

Regulation of U is achieved using a nonlinear diffusion process together with accumulation of $h(\mathbf{x},t)$:

$$\begin{aligned} \frac{\partial U}{\partial t} &= h(\mathbf{x},t) + \nabla \cdot \left[\mathcal{H}_\epsilon \circ \Big((U/U_{\mathrm{M}})^2 - 1 \Big) \nabla U \right] \ , \\ U(\mathbf{x},0) &= 0 \ . \end{aligned} \tag{43}$$

Changes in U are dominated by $h(\mathbf{x},t)$ for $|U| \ll U_{\mathrm{M}}$. As $|U|$ grows, the diffusion coefficient $\mathcal{H}_\epsilon \circ \Big((U/U_{\mathrm{M}})^2 - 1 \Big)$ gains influence. The following example illustrates qualitative behavior of the U system.

Example Consider the two-dimensional image slice shown in Figure 12a. A simulated "user" chooses locations $\mathbf{x}_k$ at which an update $h(\mathbf{x},t)$ is applied according to Eq. 41. Blue '$\times$' and red '+' denote places where $h(\mathbf{x},t)$ is negative and positive, respectively. Figure 12b shows what U would look like without nonlinear diffusion. Figure 12c shows the response of the regulated U-system from Eq. 43. Comparing 12b and 12c, it is clear that the latter is smoother and satisfies $U \leq U_{\mathrm{M}}$.

3.4.3 Label-Error Estimate Dynamics

Dynamically estimating the reference state necessitates a coupled system; ϕ and the estimate of ψ evolve simultaneously. Let $\hat{\psi}(\mathbf{x},t)$ be an estimate for $\psi(\mathbf{x})$ and define the error terms

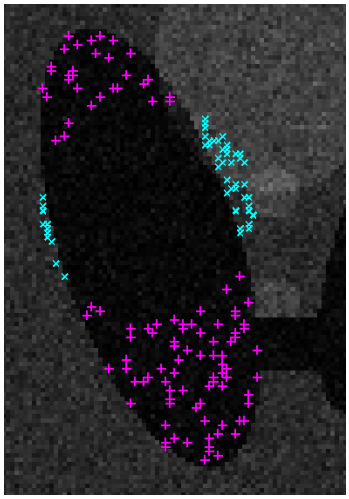
$$\hat{\xi} \doteq \mathcal{H}_\epsilon \circ \phi - \mathcal{H}_\epsilon \circ \hat{\psi} \ , \quad e_U \doteq \mathcal{H}_\epsilon \circ \hat{\psi} - \mathcal{H}_\epsilon \circ U \ . \tag{44}$$

Feedback in the ϕ system (Eq. 20) will now use $\hat{\xi}$,

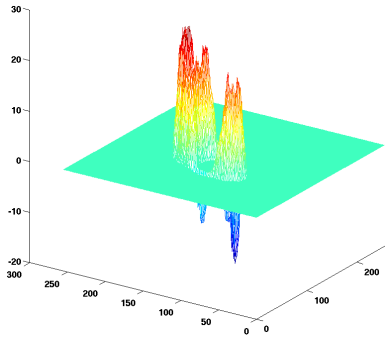
$$\begin{aligned} \phi_t &= \delta_\epsilon \circ \phi \left[G(\phi, I) + \lambda \kappa(\phi) + f(\mathbf{x}, U, \hat{\xi}) \right] \\ \phi(\mathbf{x},0) &= \phi_0 \ , \end{aligned} \tag{45}$$

where the initial ϕ_0 is specified by the user. The auxiliary $\hat{\psi}(\mathbf{x},t)$ observer-like system is driven by accumulated user input U together with error terms $\hat{\xi}$ and e_U:

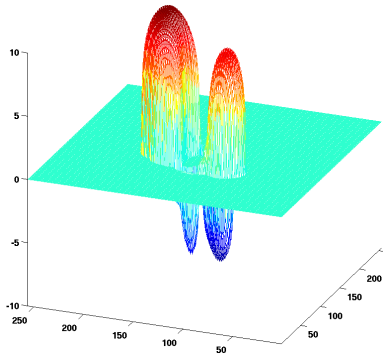
$$\begin{aligned} \hat{\psi}_t &= \delta_\epsilon \circ \hat{\psi} \left[\hat{\xi} + g(\hat{\xi}, U, e_U) \right] \\ \hat{\psi}(\mathbf{x},0) &= \phi(\mathbf{x},0) \ . \end{aligned} \tag{46}$$



(a) Simulated inputs u_k.



(b) Nominal $U_t = h(\mathbf{x}, t)$.



(c) Nonlinear diffusion Eq. 43.

Figure 12: Regulating the input-integration with nonlinear diffusion keeps U smooth and bounded. Diffusion occurs when inputs u_k occur in excess of U_{M}; here, $U_{\text{M}} = 10$.

Total labeling error is defined by the following functionals of $\hat{\xi}$ and e_U, where α is a constant parameter:

$$\text{(observer vs. received user corrections)} \qquad \mathcal{F}(t) \doteq \frac{1}{2} \int_{\Omega} (\alpha U)^2 \, e_U^2 d\mathbf{x} \; , \tag{47}$$

$$\text{(observer vs. visualized segmentation)} \qquad \hat{\mathcal{D}}(t) \doteq \frac{1}{2} \int_{\Omega} \hat{\xi}^2 d\mathbf{x} \; . \tag{48}$$

In addition to stabilizing $\mathcal{F}+\hat{\mathcal{D}}$, the control proposed in Theorem 3.4.1 is designed to achieve a useful qualitative behavior. When the user is satisfied with the agreement between $\phi(\mathbf{x}, t)$ and their ideal $\psi(\mathbf{x})$, it is assumed that $U(\mathbf{x})$ remains constant; either the user never needed to apply a correction near $\mathbf{x}$ or has otherwise stopped adding more inputs. In this case, $\hat{\psi}$ should follow ϕ. Conversely, when $U(\mathbf{x})$ grows due to persistent human input, $\hat{\psi}$ is to become increasingly driven towards U irrespective of agreement between $\hat{\psi}$ and ϕ. Subsequently, $\hat{\psi}$

should pull ϕ along due to the coupling term $-\hat{\xi}U^2$ (Eq. 23) in the closed-loop dynamics of ϕ .

Theorem 3.4.1 *Let* $g(\hat{\xi}, U, e_U) = -e_U(\alpha U)^2$ *and consequently*

$$\hat{\psi}_t = \delta_\epsilon \circ \hat{\psi} \left[\hat{\xi} - e_U(\alpha U)^2 \right] \ . \tag{49}$$

Assume that user input has stopped (U remains constant) and Theorem 3.3.1 is satisfied. Then, the sum $V(t) \doteq \hat{\mathcal{D}} + \mathcal{F}$ *has a negative semidefinite time derivative:*

$$V'(t) \leq -\int_\Omega \delta_\epsilon^2 \circ \phi \ \hat{\xi}^2 dx - \int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \left[(\alpha U)^2 e_U - \hat{\xi} \right]^2 dx \ . \tag{50}$$

Proof:

Computing the time derivative $V'(t) = \hat{\mathcal{D}}' + \mathcal{F}'$,

$$\mathcal{F}' = \int_\Omega (\alpha U)^2 e_U \dot{e_U} d\mathbf{x} = \int_\Omega (\alpha U)^2 e_U (\delta_\epsilon \circ \hat{\psi} \cdot \hat{\psi}_t) d\mathbf{x} \tag{51}$$

$$\hat{\mathcal{D}}' = \int_\Omega \hat{\xi}\dot{\hat{\xi}} d\mathbf{x} = \int_\Omega \hat{\xi}(\delta_\epsilon \circ \phi \cdot \phi_t - \delta_\epsilon \circ \hat{\psi} \cdot \hat{\psi}_t) d\mathbf{x} \tag{52}$$

Substituting for ϕ_t and $\hat{\psi}_t$,

$$\mathcal{F}' = -\int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \ (\alpha U)^2 \left[e_U^2 \alpha^2 U^2 - e_U \hat{\xi} \right] d\mathbf{x} \tag{53}$$

$$\hat{\mathcal{D}}' = \int_\Omega \delta_\epsilon^2 \circ \phi \ \left[\hat{\xi} G(\phi, I) - \hat{\xi}^2 U^2 + \lambda \hat{\xi} \kappa(\delta_\epsilon^2 \circ \phi \cdot \hat{\xi}) \right] d\mathbf{x} - \int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \left[\hat{\xi}^2 - e_U \hat{\xi} (\alpha U)^2 \right] d\mathbf{x} \quad . \tag{54}$$

After adding Eq. 53 to Eq. 54 and combining the $\delta_\epsilon^2 \circ \hat{\psi}$ terms, the portion containing error e_U can be conveniently factored:

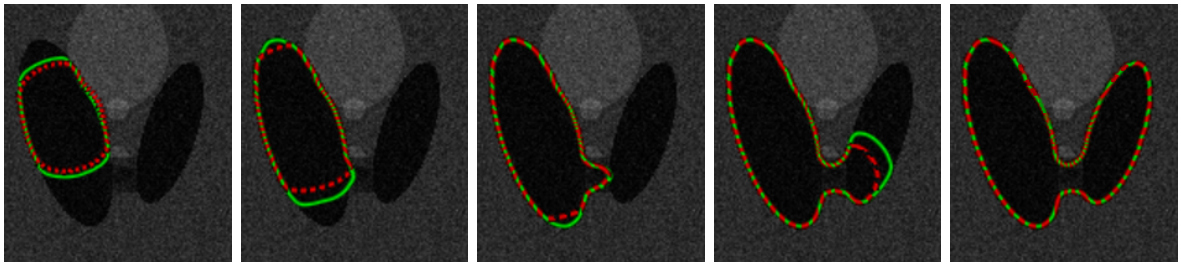
$$\begin{aligned} \mathcal{F}' + \hat{\mathcal{D}}' = &\int_\Omega \delta_\epsilon^2 \circ \phi \left[\hat{\xi} G(\phi, I) - \hat{\xi}^2 U^2 + \lambda \hat{\xi} \kappa(\delta_\epsilon^2 \circ \phi \cdot \hat{\xi}) \right] d\mathbf{x} \\ &- \int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \left[e_U^2 \alpha^4 U^4 - 2(\alpha U)^2 e_U \hat{\xi} + \hat{\xi}^2 \right] d\mathbf{x} \ , \end{aligned} \tag{55}$$

$$= \int_\Omega \delta_\epsilon^2 \circ \phi \left[\hat{\xi} G(\phi, I) - \hat{\xi}^2 U^2 + \lambda \hat{\xi} \kappa(\delta_\epsilon^2 \circ \phi \cdot \hat{\xi}) \right] d\mathbf{x} - \int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \left[e_U (\alpha U)^2 - \hat{\xi} \right]^2 d\mathbf{x} \ . \tag{56}$$

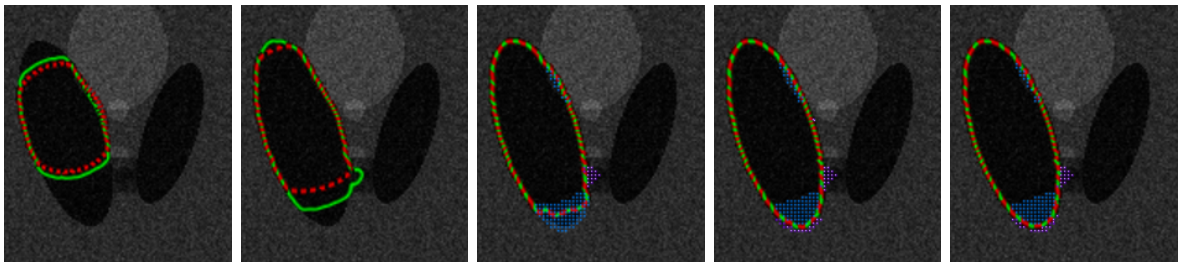
When U and λ satisfy Theorem 3.3.1, it follows that

$$V'(t) = \mathcal{F}' + \hat{\mathcal{D}}' \leq -\int_\Omega \delta_\epsilon^2 \circ \phi \cdot \hat{\xi}^2 d\mathbf{x} - \int_\Omega \delta_\epsilon^2 \circ \hat{\psi} \left[(\alpha U)^2 e_U - \hat{\xi} \right]^2 d\mathbf{x} \leq 0 \ . \tag{57}$$

Thus, $V'(t)$ is negative semidefinite and $V(t)$ bounded. ∎



(a) No user input: $\hat{\psi}$ follows ϕ.

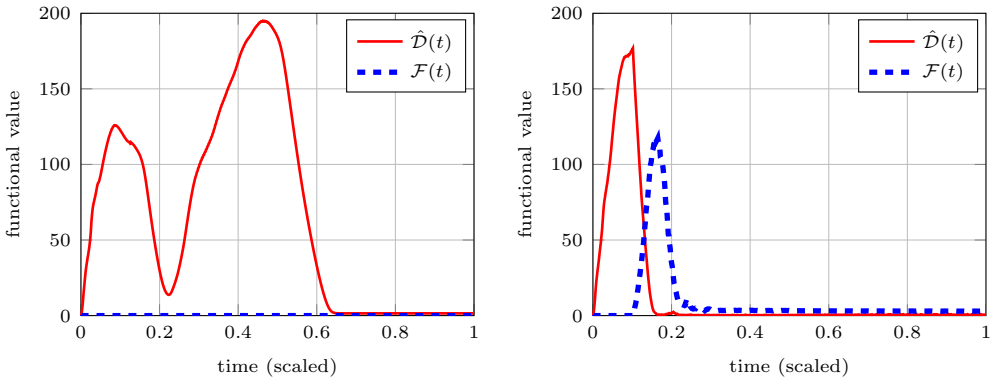


(b) With user input: sufficient U drives ϕ towards $\hat{\psi}$.

Figure 13: A synthetic image example. The human user seeks to segment only the left ellipsoid region; the open-loop system tends to creep through and segment the union of the two ellipsoids. $|\phi| \leq \epsilon$ and $|\hat{\psi}| \leq \epsilon$ are denoted by solid-green and dashed-red contours, respectively. Dotted regions in (b) indicate $|U| > 0$.

3.4.4 Synthetic Image Example

To demonstrate the coupled dynamics, this section considers a simple segmentation scenario. Figure 13 illustrates closed-loop system behavior on the synthetic image used previously in § 3.2.2. In the absence of user input, $\phi(\mathbf{x}, t)$ behaves like the open-loop system; Figure 13a shows the $\hat{\psi}$ estimate following ϕ until they both reach steady state. With user input, the estimated ideal contour mediates between user input and the open-loop segmentation dynamics. In Figure 13b, the user starts to apply input upon noticing the $\phi = 0$ boundary creeping through the bridge between the two ellipsoidal regions. Input stops and the system reaches steady-state after the user is satisfied with the displayed segmentation. Comparing Figure 13a and Figure 13b, we see that regardless of user input, the closed-loop sytem aligns the zero level-sets of ϕ and $\hat{\psi}$ at steady-state; the presence of user input in Figure 13b shifts the steady-state of ϕ and $\hat{\psi}$. In both cases the α for Eq. 49 is set to $1/U_{\mathrm{M}}$.



(a) No user input.

(b) With user input.

Figure 14: Shown above are the values of $\hat{\mathcal{D}}$ and $\mathcal{F}$ corresponding to the synthetic image example of Figure 13. *Left*: ϕ is driven only by the image-dependent term in the absence of user input. *Right*: $\mathcal{F}(t)$ rapidly grows when the user applies input. $\hat{\mathcal{D}}(t)$ rapidly shrinks as ϕ is drawn towards $\hat{\psi}$. At steady-state, $\hat{\mathcal{D}}(t) < 0.5$ and $\mathcal{F}(t) < 3$.

3.5 Application to MRI and CT Images

In this section, the feedback-augmented level-set methods are applied to two specific problems involving interactive medical image segmentation of X-ray Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) volumes. In § 3.5.1, a fractured piece of the femur is segmented in a CT volume. Next, the technique is applied to extract a patellar tendon in an MRI volume in § 3.5.2. For both applications, we first review the clinical problem. Next, an open-loop system appropriate for the image type is chosen. Finally, the closed-loop ϕ system is summarized, followed by a discussion of the segmentation results.

3.5.1 CT Segmentation with Mean-Alignment

The realignment of bone fragments after a fracture, also referred to as *fracture reduction*, is a crucial task during the operative treatment of complex bone fractures. Anatomically incorrect fracture reduction can result in severe post-traumatic complications. In order to avoid such problems and obtain an optimal fit between all relevant fracture fragments, the surgeon traditionally exposes the fractured bone by cutting the soft tissue envelope to access the fragments directly. Subsequent realignment of the recovered fracture fragments requires a trial and error approach, which prolongs surgery and increases the risk of complications for

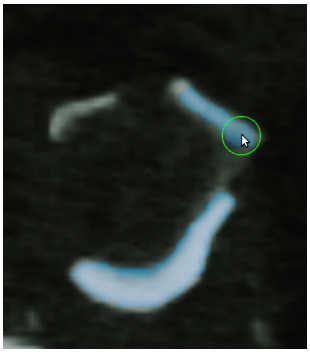
the patient. Therefore, there is a clear need for the development of less invasive techniques to reconstruct complex fractures. Segmentation of the image data to localize the fragments (as in Figure 15) is a key step first step toward computing and planning the optimal way of realigning the fractured bone.

Bone tissue generally appears very bright in CT imagery; therefore, the segmentation of bone in CT is modeled with the mean-alignment system Eq. 19. Using the control from Eq. 23 leads to the closed-loop system

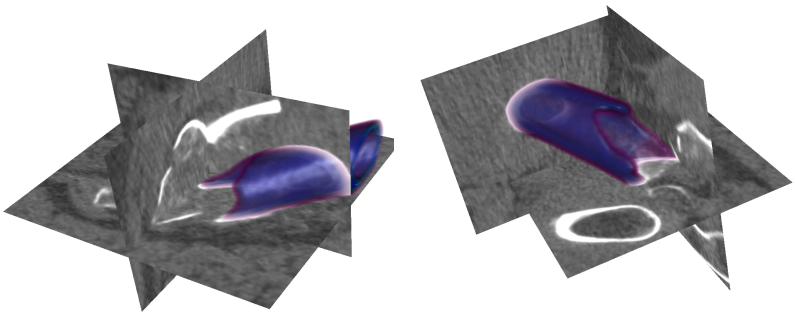
$$\phi_t = \delta_\epsilon \circ \phi \left[-(I - \mu_{\text{in}})^2 + (I - \mu_{\text{out}})^2 - \hat{\xi} U^2 + \lambda(t)\ \kappa(\delta_\epsilon^2 \circ \phi \cdot \hat{\xi}) \right] . \tag{58}$$

Note that a healthy bone can often be segmented in its entirety using the open-loop system alone, since the zero level-set of ϕ is naturally drawn to boundaries of bright objects. Interactive control becomes vital, however, when segmenting bone subject to disease or injury; accurate segmentation is not possible without feedback.

Figure 16 illustrates several aspects of the interactive system applied to the segmentation of a large bone fragment. Local maxima of $U(\mathbf{x})$ are shown as markers along with the intermediate $\phi = 0$ boundary in Figure 16(a)-(d); the green semi-opaque surface represents the user's reference $\psi = 0$ level-set. Regions of bright CT image values are quickly segmented by the open-loop system; the user generates some input along what appears to be an edge of the fractured bone where $I(\mathbf{x})$ is darker (Figure 16a). After the system has segmented this first edge and is nearly at steady-state, the user finds another edge along which to apply input (Figure 16b). A further refinement is made in Figure 16c that leads to the final steady-state segmentation of Figure 16d. The number of voxels actuated by the user's mouse strokes are plotted versus (scaled) time in Figure 16e. In a fully manual segmentation, each of the 16404 boundary voxels in Figure 16d would need to be marked by the user; the second y-axis on the right indicates the actuated voxels as a percentage of the fully manual effort. It is difficult for a person to accurately decide whether or not a fracture edge in a distant part of the volume is part of the same fragment. Figure 16e indicates that a substantial portion of time is spent with ϕ near a steady-state while the user scrolls through slices to decide where fracture edges are located, and whether these edges are part of the same



(a) User input with mouse.

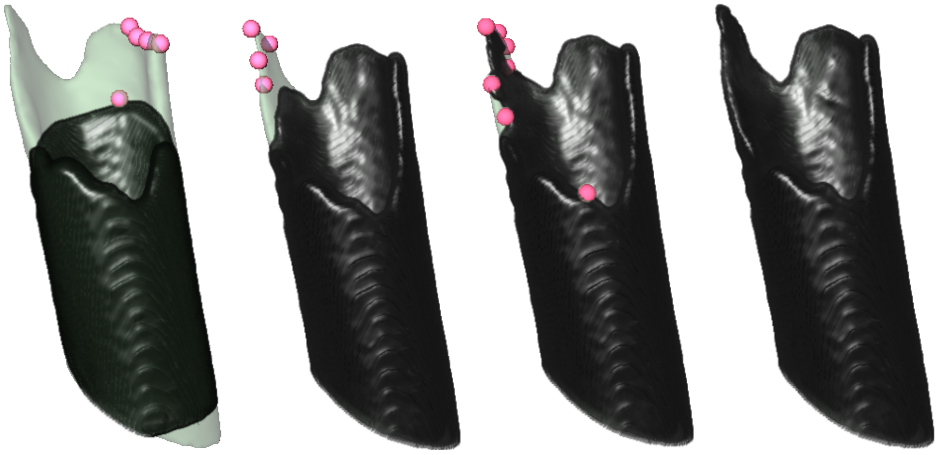


(b) Two views of final segmentation.

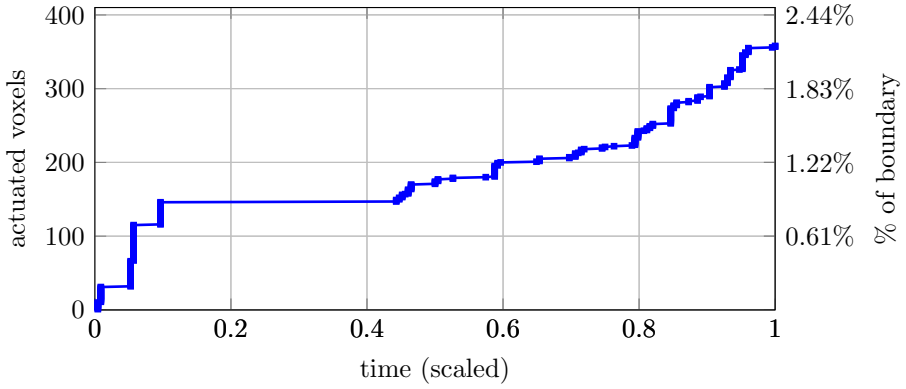
Figure 15: Segmentation of shattered hip bone fragments in a CT scan. The image volume is $156 \times 162 \times 229$ voxels with a 0.7mm grid spacing.

fragment or another one in close proximity. In Figure 15a, two light regions are determined to be part of the fragment being segmented, while a third (in the upper left) is a separate bone fragment.

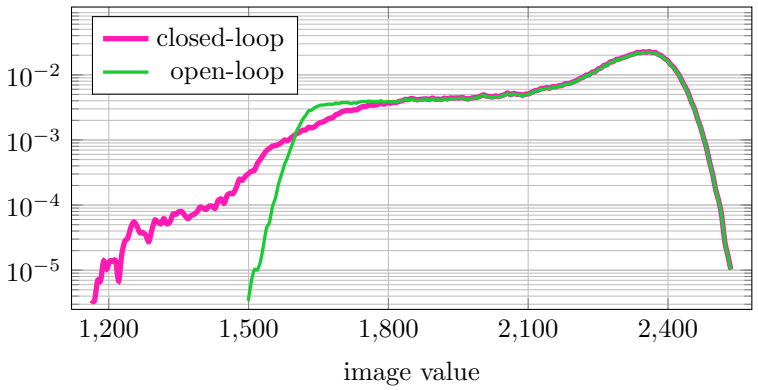
Normalized histograms of the image intensity distribution inside of the segmentation boundary at steady-state are shown in Figure 16f. Without feedback, the segmentation encloses a region with a highly peaked $I(\mathbf{x})$ histogram. In contrast, the closed-loop system reaches steady-state with a heavier-tailed intensity histogram. The distribution shift is due to the user applying input to correctly label bone near and along the jagged fragment edges. These regions are precisely where we care most about accurate segmentation, since the fragment's edges are to be matched with those of other fragments during the fracture reduction task.



(a) $t = 0.1$ (b) $t = 0.4$ (c) $t = 0.8$ (d) $t = 1.0$



(e) User-actuated voxels over time.



(f) Normalized histogram of $I(\mathbf{x})$ within segmented region.

Figure 16: In (a)-(d), regions of user input are shown as markers on the progressing segmentation (dark) overlayed on user's reference boundary (light). The segmentation in Figure 16a is the steady state of the open-loop system. In Figure 16d, the segmentation agrees with the desired reference boundary due to the closed-loop system's incorporation of user input.

3.5.2 MRI Segmentation with Localized Statistics

Surgical repair of a torn anterior cruciate ligament (ACL) requires choosing a location from which to harvest a graft of sufficient length and thickness. The most common choice today is the patellar tendon (PaT). While the width and thickness of a PaT are quite predictable based on patient height and weight, the tendon length varies widely. This variability in shape continues to complicate surgery due to mismatch between the graft and drilled tunnel, especially in "anatomic reconstruction" [50, 68, 118] where the replacement ACL is to be oriented exactly as before the injury[2]. Quantifying the variability of PaT shape and comparing it to other graft choices (namely the hamstring and quadriceps tendons) requires accurate segmentation in MRI volumes.

Soft tissue including tendons and ligaments is readily visible in MRI, unlike CT where only mineral-dense bone gives a strong response. However, images obtained by MRI have a complicated mapping between tissue type and observed intensity; segmenting soft tissue in MRI is generally more difficult than bone in CT. The distribution of intensity values in MRI arising from a particular anatomic structure will vary significantly between slices (Figure 17b), and will also overlap the distributions of other structures (Figure 17a). An effective approach for MRI segmentation is to separate regions based on spatially-varying statistics of $I(\mathbf{x})$. To do so, open-loop dynamics are chosen to use the localizing active contours of [134] that define intensity means $\mu_{in}(\mathbf{x})$ and $\mu_{out}(\mathbf{x})$ locally as integrals over a Euclidean ball of radius r. With feedback the system becomes

$$G(\phi, I) = \int_{\mathcal{B}_r(\mathbf{x})} \delta_\epsilon \circ \phi(\mathbf{y}) \left[(I(\mathbf{y}) - \mu_{\text{in}}(\mathbf{x}))^2 - (I(\mathbf{y}) - \mu_{\text{out}}(\mathbf{x}))^2 \right] d\mathbf{y} \ , \tag{59}$$

$$\phi_t = \delta_\epsilon \circ \phi \left[G(\phi, I) - \hat{\xi} U^2 + \lambda(t) \ \kappa(\delta_\epsilon^2 \circ \phi \cdot \hat{\xi}) \right] . \tag{60}$$

Despite the advantages of the underlying open-loop system, segmenting a PaT remains challenging for two reasons. First, the tendon is very thin relative to its height and width, making a satisfactory choice of r in Eq. 59 difficult. Second, $I(\mathbf{x})$ at the insertion points

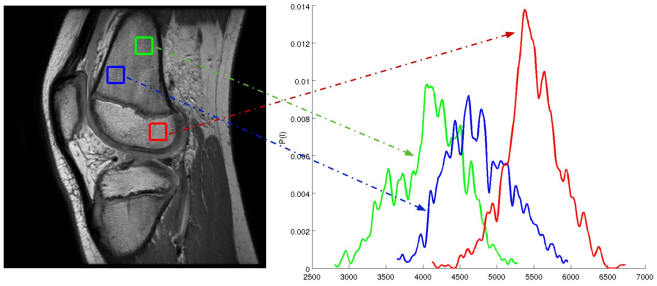
[2] as opposed to the more common approach of attaching a shorter graft to an arbitrary reachable point on the bone surface.

of the tendon has the same distribution locally as adjacent connective tissue. The human user, however, employs their anatomic knowledge to enable successful segmentation via the closed-loop system. Figure 18 shows the final result; the tendon has been segmented between its attachment on the inferior pole of the patella to the end of its insertion on the tibial tubercle. For context, the patella bone is also segmented and displayed.

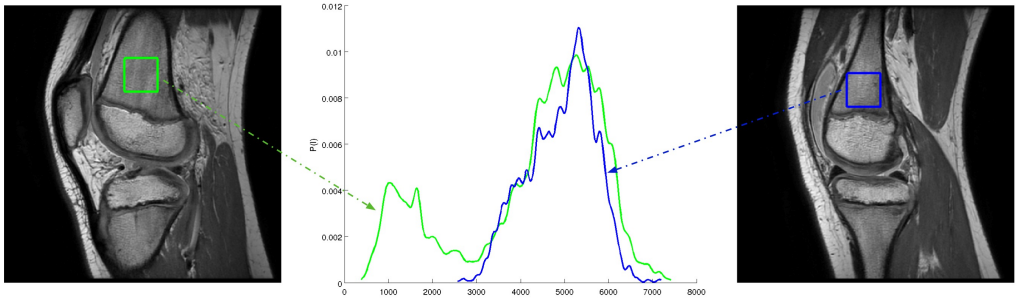
Incremental progress during interactive segmentation of the tendon is shown in Figure 19 (a)-(d). Red and blue markers denote positive and negative extrema of U, respectively, while the green semi-opaque surface represents a reference segmentation known to the human expert. As the $\phi = 0$ boundary evolves after initialization, a small amount of input yields the segmentation in Figure 19b. With the bulk of the tendon outlined, the user applies input to fill gaps in the vertical edges and remove the over-segmented regions around the insertion points at the patella and tibia bones (Figure 19c). Unlike the fracture scenario in Figure 16, the open-loop system applied to segment this tendon leads to massive "bleed-through" of the segmentation because the image distribution around the tendon insertion points is identical to that of the tendon itself. Figure 19d shows the steady-state reached by the closed-loop system; user input stabilizes the segmentation at the desired reference boundary and prevents bleed-through past the insertion points on the patella and tibia. In Figure 19e, the number of voxels actuated by the user's mouse strokes is 4.7% of what would be needed to trace all of the tendon's boundary voxels manually. Comparing Figure 16e and Figure 19e, the latter has more piecewise constant regions because the human user spends substantial time looking for anatomic markers and adjusting displayed image contrast to decide where the tendon begins and ends.

3.5.3 Comparison to Related Methods

In many implementations of level-set segmentation (e.g. [1, 110, 148]), the smoothing factor λ is a parameter that is set by a user. However, understanding such a parameter requires users to have more mathematics background than is typical for the medical community. Here, we set λ automatically to achieve desired behavior. The PDE control formulation here has a constant computational cost with respect to amount of user input and no abrupt

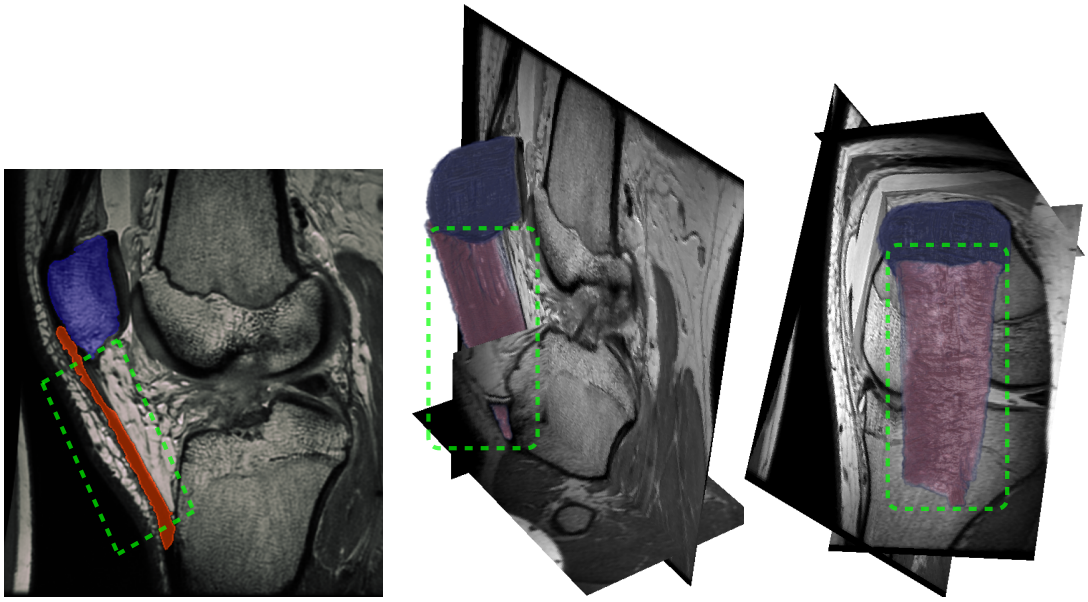


(a) One slice, overlapping $I(\mathbf{x})$ histogram



(b) One region, varying $I(\mathbf{x})$ histogram

Figure 17: Tissues within one MRI slice have overlapping intensity histograms while a single tissue across slices has a varying histogram. Separation of regions must consider the spatially-varying image statistics.

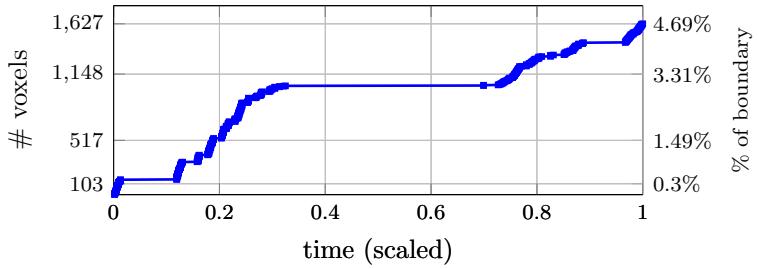


(a) Single-slice view.

(b) Two views of final segmentation.

Figure 18: A segmentation of the patella and patellar tendon in MRI, part of a study on graft selection for anterior cruciate ligament (ACL) repair. The image volume is $512 \times 512 \times 224$ voxels with a 0.4mm grid spacing.

(a) $t = 0.1$ (b) $t = 0.2$ (c) $t = 0.7$ (d) $t = 1.0$



(e) User-actuated voxels over time.

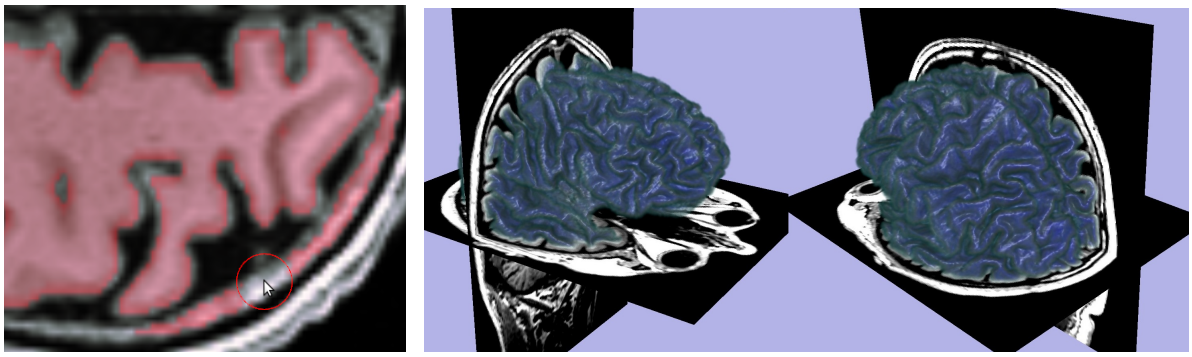
Figure 19: Regions of significant user input shown together with the changing segmentation. Locations where $U > 0$ and $U < 0$ correspond to red and blue markers, respectively. The open-loop system's tendency towards "bleed-through" near the insertion points is handled in the closed-loop system by incorporating user input with negative U.

changes to the segmentation, unlike in [52,100,149]. Under the proposed controller, sufficient input $U(\mathbf{x})$ from the user guarantees agreement with the reference state; relaxed constraints in [149] dictate that it may be impossible for the segmentation to respect the user's inputs. Rushed use of the mouse by a human is not possible in [52, 100] because constraints are exactly enforced. In contrast, the input processing used in the current work provides leeway for the user: a small $|U|$ will not dominate the open-loop dynamics. If needed, a large accumulation of $|U|$ is achieved by "scribbling" repeatedly in a region.

It is emphasized that the closed-loop control formulation in this paper does not seek

to replace existing curve evolution algorithms but rather to augment them. The control-theoretic approach enables a user to reach the desired segmentation at steady-state; running the level set evolution for a longer time will not cause the boundary to "bleed-through" or contract.

Figure 20a illustrates the same behavior of "creeping" into adjacent structures as in thesynthetic example of Figure 10. The closed-loop system Eq. 58 enables a user to accurately segment the brain surface (Figure 20b) with little effort and no understanding of algorithm parameters.



(a) User input with mouse.

(b) Two views of final segmentation.

Figure 20: Segmentation of the brain's white-matter is achieved in most of Ω with the open-loop dynamics, except for places where the bright parts of the skull are erroneously labeled (Figure 20a).

3.6 Comparison to Related Interactive Segmentation Methods

Desired anatomic boundaries will, in general, not be found by an automatic segmentation algorithm whose assumptions are violated. Several algorithms related to this paper allow the user to apply input that influences the segmentation process:

- In [100], the space of admissible Ω-partitions is restricted to the span of a finite set of radial basis functions generated by an image-dependent metric $g(I)$ and initial user-input labels at $\mathbf{x}_i \in \Omega$. Implicit basis functions $\phi(\mathbf{x}, t)$ are restricted to the form

$$\phi\left(\mathbf{x}|\left\{\lambda_i\right\}_{i=0}^{N}\right) = \lambda_0 + \sum_{i=1}^{N} \lambda_i \varphi\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|_{g(I)}}{\sigma_i}\right). \tag{61}$$

  Segmentation is posed as a minimization of a nonlinear function of the $N+1$ unknowns λ_i subject to N linear inequality constraints:

$$\begin{aligned} \min_{\lambda_i} \quad & \int_{\Omega} \mathcal{H}_\epsilon \circ \phi(\mathbf{x})|\left\{\lambda_i\right\}_{i=0}^{N} r(\mathbf{x}) d\mathbf{x} \\ \text{subject to:} \quad & \gamma_k \phi(\mathbf{x}) \geq 0, \ k = 1, \ldots, N \ . \end{aligned} \tag{62}$$

The constraints arise from locations $\mathbf{x}_i$ where the user has specified the required sign of $\phi(\mathbf{x}_i)$. While the finite-dimensional formulation is attractive when compared to an infinite dimensional PDE formulation, there are significant drawbacks to Eq. 62. Smooth basis functions will not model sharp edges or a noisy structure well; a large number of constraints will be needed when the user tries to segment in such a situation. Two problems then arise: first, new constraints can shift the feasibility cone in a way that forces the λ_i values far from the previous feasible solution. Second, the computational complexity increases over time as more user input arrives, since each input click adds more variables and constraints to the optimization program.

- Another representation of the segmentation problem is to consider pixels as vertices connected to neighboring pixels with undirected edges. The resulting graph with vertices v_i and edges e_{ij} is then subjected to a discrete optimization over vertex and edge weights. Weights penalize the membership of v_i and edges e_{ij} in the interior and on the boundary of a segmented region, respectively. In [149], input from the user is combined with an image-based cost to generate edge and vertex weights. A segmentation is then obtained by solving for a "probability" p_i of vertex v_i being in the interior of the desired region:

$$\begin{aligned} \min_{w_{ij},\, p_i} \quad & \sum_{e_{ij}} w_{ij}(p_i - p_j)^2 + \lambda \sum_{v_i} (p_i - 1/2)^2 \\ & p_i = 1 \quad \text{if user labels } v_i \text{ as interior,} \\ \text{subject to:} \quad & p_i = 0 \quad \text{if user labels } v_i \text{ as exterior,} \\ & p_i = \tfrac{1}{2} \quad \text{if user labels } v_i \text{ as boundary.} \end{aligned} \tag{63}$$

In this formulation, some heuristic is needed for setting edge weights w_{ij} and user input weight λ. However, it is not guaranteed that the user-specified points actually end up in the final segmentation as stated by the user unless $\lambda \to \infty$. But if λ is large, a degenerate $p_i = 1/2$ solution can arise for the pixels that were not part of the user input. Additionally, when the user provides more inputs over time, the segmentation boundary can change dramatically upon solving an updated version of Eq. 63. Several helpful heuristics for choosing w_{ij} are noted in [149] to make dramatic changes in the

boundary less likely, but the underlying issue remains. In particular, the heuristics for limiting changes in a region boundary assume that the region's interior has a unique color distribution that does not occur elsewhere in the image. This assumption rarely holds for medical image segmentation since many anatomic structures have similar brightness in CT and MRI scans.

- Energy-minimizing implicit functions for interactive segmentation are considered in [53]; a global surface-fitting algorithm interpolates between user-drawn curves in 3-D. The formulation involves only the geometry of the segmentation boundary and user-drawn regions; it does not depend on the image data. There is an implied feedback via the human user, since the initialization of contours can be modified after each new visualized result.

3.7 Conclusion

This paper has presented a modeling approach that enables control-theoretic analysis and design for interactive medical image segmentation. Results shown for a synthetic image (§ 3.4.4) and real medical volumes (Section 3.5) agree with theoretical expectations of system performance. Section 3.5 illustrated two qualitatively different situations: (1) gradual expansion of the boundary to bound the entire femur fragment and (2) prevention of "bleed-through" or over-segmentation with the patellar tendon. In both situations, the user is able to drive the segmentation to a desired steady state and to do so with much less effort in terms of actuated voxels than manual segmentation. In summary, the PDE control formulation enables us to guarantee a user's ability to reach a reference segmentation state while also absolving them of the need to understand mathematical details or use precise mouse movement.

Successful use of the closed-loop algorithm by medical students motivates several future extensions. If a single image contains several objects of interest, they would need to be extracted sequentially in the current framework. Such a sequential de-coupled approach does not address natural constraints of the geometry and involves re-editing common boundaries.

A coupled formulation using an open-loop system of PDEs such as in [148] together with a vector of control inputs would prevent region overlap and reduce the user's effort when segmenting multiple regions. Informative visualization is vital for efficient interaction, since performance of any interactive segmentation method is limited by how quickly and accurately a user can infer the segmentation state [46, 59, 115]. An interesting extension to the theory would consider the feedback between visualization and the creation of user input; for example, it may be desirable to confine movement of the boundary to regions that are observable from the user's viewpoint in 3D.

CHAPTER IV

POSE INVARIANT CLOSED-LOOP TRACKING

A visual servoing algorithm using region-based measurements of a planar target is derived to achieve fixed performance characteristics in the presence of depth and target pose uncertainty. Lyapunov-based control design directly incorporates data from image-based pose estimation to make disturbances due to unknown target velocity have bounds independent of target-camera relative pose and distance. Several image processing schemes are presented for computing the required region-based geometric quantities. Simulation with a 3D synthetic environment incorporates the effect of pixel coordinate quantization and difficulty of consistent feature-matching between frames. Implementing the control algorithm with template-based iterative pose estimation confirms the expected invariant tracking performance across a variety of environments, including targets having repetitive texture and background clutter.

4.1 Introduction

This paper examines the geometry associated with tracking a planar, or nearly planar, object in order to derive a visual servoing controller whose performance is invariant with respect to target pose. The specific servoing task specifies that a target with unknown motion be kept centered in the image as viewed by a moving monocular camera equipped with actuators. *Pose-invariance* means that a target whose unknown velocity has a fixed bound is tracked with image-coordinate error that does not scale with range or in-place rotation of the target.

Tracking is an estimation procedure that generates a spatial description of the target within the image. This description is typically either a point or a closed region in the image. Visual tracking is inherently nonlinear, and thus necessitates design procedures from nonlinear control when closing the loop. Nonlinear strategies employed in the present

work are addressed in [69,73]. Recently these nonlinear design techniques have been applied to adaptive camera calibration [63] and motion trajectory tracking [145].

Typical systems in hardware consider the vision estimation algorithms as black-box measurement sources. Image formation, rigid body motion, and projective geometry are now well-understood [51,85,102], as are the significant difficulties in computing solutions to the related *inverse* problems. A number of papers in the control field work towards the goal of linking control-theoretic design to an understanding of geometric image formation, with a particular focus on homography decomposition and depth estimation from correspondence of projected points [25,37,38,57,58,77,87,99]. Feedback terms used in these works require as input a stream of solutions to inverse problems in computer vision, such as reliable feature-correspondence and homography-decomposition for rigid pose estimation. Linear solutions to these problems come from computer vision literature, where they are derived under assumptions of zero or highly structured measurement error. Accurate, numerically-stable geometric estimation requires nonlinear optimization and outlier-rejection schemes [51,85,138].

Recently developed point-feature algorithms such as LDAHash [137] and BRIEF [20] seek to speed up the evaluation and comparison times of individual features by using effective data structures. The performance on large-scale image-retrieval and 3D reconstruction is not directly indicative of the control outcome of using feature-matching algorithms. The need to find groups of features that encode a consistent pose necessitates a costly subset-selection step, such as RANSAC. A source of reliable area and segmentation measurements is an integral part of the proposed invariant servoing; both feature-based and template-image-based pose estimation schemes can be used, though the latter are significantly more reliable in practice.

Contribution. In [65], a depth-invariant visual servoing strategy was derived by assuming the existence of a variable focal length monocular camera. Augmenting the system dynamics to include the rate of change of the focal length for area stabilization neutralized the nonlinear effects of the camera projection equations. The present work expands on [66]

where visual servoing control is constructed so as to achieve invariance with respect to **(1)** absolute range to target, and **(2)** in-place rotation by the target. Furthermore, when upper bounds on the target velocity are known, the Lyapunov analysis is exploited to determine the minimal gain choice that will guarantee the desired performance (in terms of servoing error).

It is assumed that a template image $\mathbb{I}_0$ of the target is available and that it corresponds to a planar region viewed head-on (*i.e.*, with the target plane's normal parallel to the optical axis). A control policy is sought such that future images, $\mathbb{I}(t)$, capture the target as close to the image center as possible. Implementation has a prerequisite of segmentation and pose estimation algorithms.

Organization. The remainder of this chapter is organized as follows. System dynamics and geometric quantities of interest for visual tracking are introduced in Section 4.2. A Lyapunov-based controller is synthesized in Section 4.3, with the sources of variability due to range and pose illustrated. Section 4.4 addresses the compensation of region-based measurements, thus enabling pose-invariant servoing. Section 4.5 presents several pose estimation algorithms that generate the required measurements. Section 4.6 verifies the expected tracking performance of the closed-loop system in a virtual 3D environment. Finally, Section 4.7 summarizes results, applicability, and relationship to the existing literature.

4.2 Formulation: Perspective Vision

Constructing a visual-servoing controller invariant to range and target pose requires an analysis of three related concepts which describe the system's dynamics and image-based measurements:

- Independent Euclidean motion of the camera and target. Interaction between these two motions describes the dynamics of 3D points in the camera frame.
- Pinhole camera projective image-formation, which projects 3D point coordinates to 2D image points. Disturbances due to unknown target motion become dependent on depth.

- Region-based segmentation, which provides target area and centroid measurements. Projective image-formation means that these quantities are biased by target pose, particularly in-place rotation.

4.2.1 Rigid Camera Motion

A rigid body transformation can be represented by a rotation $R \in SO(3)$ and a Euclidean translation $T \in \mathbb{R}^3$. The set of all such configurations, described in homogeneous matrix form, is

$$SE(3) \doteq \left\{ \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} | R \in SO(3), T \in \mathbb{R}^3 \right\} , \tag{64}$$

and is called the special Euclidean group in three dimensions. It is a subgroup of the four dimensional general linear group $GL(4)$. Let us denote a particular element in $SE(3)$ by g_P^C. Elements of $SE(3)$ also describe transformations of reference frames for points; g_P^C maps a homogeneous coordinate point q^P in the world frame to q^C in the camera frame via

$$q^C(t) = g_P^C(t) q^P(t) . \tag{65}$$

Since control is to be employed via camera motion, the dynamical system lends itself to analysis most naturally with a camera-centric (or *body*) formulation of the dynamics. Differentiating Eq. 65 yields

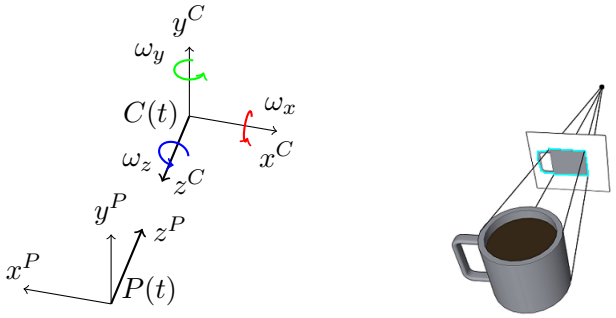
$$\dot{q}^C(t) = \dot{g}_P^C q^P(t) + g_P^C(t) \dot{q}^P(t) , \tag{66}$$

where the time derivative of the $SE(3)$ map g_P^C satisfies

$$\dot{g}_P^C \left(g_P^C\right)^{-1} = \begin{pmatrix} \left(R_P^C\right)^T \dot{R}_P^C & -\left(R_P^C\right)^T \dot{T}_P^C \\ 0 & 0 \end{pmatrix} = \zeta_C^C(t) . \tag{67}$$

$\zeta_C^C(t)$ is the body velocity as it describes motion of the camera with respect to its own moving coordinate frame via a skew-symmetric block matrix $\hat{\omega}(t)$ and a translational vector $\nu(t)$:

$$\hat{\omega} \doteq \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} , \ \zeta_C^C(t) \doteq \begin{pmatrix} \hat{\omega}(t) & \nu(t) \\ 0 & 0 \end{pmatrix} . \tag{68}$$



(a) System

(b) Measurement

Figure 21: The system consists of a moving pan-tilt camera observing a moving target via a monocular, projective camera. Camera and target reference frames denoted by $C(t)$ and $P(t)$, respectively. Rotation control acts on the camera via $\hat{\omega} = (\omega_x, \omega_y, \omega_z)^T$.

Due to $\frac{d}{dt}((R_C^P)^T R_C^P) = 0$, $\hat{\omega}$ satisfies $\dot{R}_C^P = R_C^P \hat{\omega}$. Figure 21a illustrates the camera-frame angular motion encoded in $\hat{\omega}$. Using ζ_C^C, the dynamics of observed 3D point q^C are given by

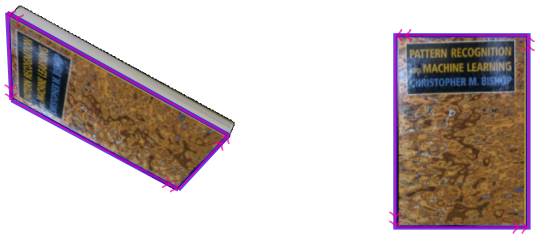
$$\dot{q}^C = \zeta_C^C \left(g_P^C q^P\right) + g_P^C \dot{q}^P = \zeta_C^C q^C + g_P^C \dot{q}^P \ . \tag{69}$$

The velocity $\dot{q}^P$ represents unknown target motion in the world-frame, leading to camera-frame input disturbance $(v_x^C, v_y^C, v_z^C, 0)^T = g_P^C \dot{q}^P$.

4.2.2 Observed Image Dynamics

Coordinates $q^C(t)$ on the target's surface in 3D are not directly observable for a monocular camera. Measurements are generated by a projective transformation [85], whereby depth information is lost. Setting $q^C = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^T$ and using the model of a pinhole camera with fixed image size and isotropic scaling, the resulting projected coordinates are given by

$$x_c = f\frac{\mathrm{X}}{\mathrm{Z}}, \quad y_c = f\frac{\mathrm{Y}}{\mathrm{Z}}, \quad \Pi(q^C) \doteq \begin{pmatrix} x_c \\ y_c \end{pmatrix}. \tag{70}$$



(a) Observed. (b) Rectified.

Figure 22: Segmentation of the target is defined via a parametrized closed curve. Above are the direct observation with perspective warp (a) and rectified version (b).

With a fixed focal-scale[1] f, point-dynamics in the image coordinate system are given by

$$\frac{d}{dt}\Pi(q^C) = \mathbf{D}\Pi(q^C)\left(\dot{q}^C\right) \tag{71}$$

$$= \begin{pmatrix} \frac{f}{\mathrm{Z}} & 0 & -\frac{f\mathrm{X}}{\mathrm{Z}^2} & 0 \\ 0 & \frac{f}{\mathrm{Z}} & -\frac{f\mathrm{Y}}{\mathrm{Z}^2} & 0 \end{pmatrix} \left(\zeta_C^C q^C + g_P^C \dot{q}^P\right) . \tag{72}$$

This is a product of the measurement's *interaction matrix* [24] with a twist-induced motion. Assuming rotation-only camera motion, the resulting image-coordinate dynamics are

$$\begin{pmatrix} \dot{x_c} \\ \dot{y_c} \end{pmatrix} = \begin{pmatrix} y_c\omega_z - \omega_y f + \frac{\omega_x x_c y_c - \omega_y x_c^2}{f} + \frac{f v_x^C - x_c v_z^C}{\mathrm{Z}} \\ -x_c\omega_z + \omega_x f - \frac{\omega_y x_c y_c - \omega_x y_c^2}{f} + \frac{f v_y^C - y_c v_z^C}{\mathrm{Z}} \end{pmatrix}. \tag{73}$$

4.2.3 Region-Based Formulation

Obtaining measurements in image-coordinates is necessary for control. It is desirable to not rely simply on *feature-point-matching* but rather use the entirety of an available template view of the target. Region-based segmentation, illustrated in Figure 22, provides such measurements. In particular, closed-loop tracking requires the computation of target centroid and area in the image plane.

Define the image domain to be Ω, with a region $\mathcal{S}(t) \subset \Omega$ occupied by the target. Closed-loop tracking seeks to move the camera so as to keep the target object's centroid in

[1] a function of *focal length*, angle of view, and pixel scaling

the image center. A direct way to measure the image-coordinate centroid of the target is

$$\bar{x}_c(t) = \frac{\int_{\mathcal{S}} x_c dx_c dy_c}{\int_{\mathcal{S}} dx_c dy_c} \ , \quad \bar{y}_c(t) = \frac{\int_{\mathcal{S}} y_c dx_c dy_c}{\int_{\mathcal{S}} dx_c dy_c} \ . \tag{74}$$

Algorithms are often formulated to estimate the location and motion of $\mathcal{S}(t)$'s boundary. Such a representation implicitly defines $\mathcal{S}(t)$ and is denoted here as

$$\Gamma(t,s) = \begin{pmatrix} x_c(t,s) \\ y_c(t,s) \end{pmatrix} \ , \tag{75}$$

with time-derivative $\Gamma_t(t,s)$. Integration in Γ and $\mathcal{S}$ are related via the vector calculus identities. Thus, area occupied by the target is equivalently defined by

$$A(t) = \int_{\mathcal{S}} 1 \ dx_c dy_c = \frac{1}{2} \oint_{\Gamma} \left\langle \begin{pmatrix} x_c(t,s) \\ y_c(t,s) \end{pmatrix}, \mathbf{N} \right\rangle ds \ . \tag{76}$$

Consider a target of finite extent that is planar, and whose normal to the plane aligns with the camera's optical axis z^C. The target's area in acquired images will vary with depth. For a template view of the target from a known distance z_0,

$$A_0 = \int_{\mathcal{S}_0} dx_c dy_c = \int_{\mathcal{A}} \frac{f^2}{\mathrm{z}_0^2} d\mathrm{X} d\mathrm{Y} = \frac{f^2}{\mathrm{z}_0^2} \int_{\mathcal{A}} d\mathrm{X} d\mathrm{Y}, \tag{77}$$

where $\mathcal{S}_0$ is the segmented region of the target, and $\mathcal{A}$ gives the collection of coplanar coordinates associated to the target in 3D. A later view of the same target at the distance $\mathrm{z}(t)$ has the area

$$A(t) = \int_{\mathcal{S}(t)} dx_c dy_c = \int_{\mathcal{A}(t)} \frac{f^2}{\mathrm{z}^2} d\mathrm{X} d\mathrm{Y} = \frac{\mathrm{z}_0^2}{\mathrm{z}^2(t)} A_0. \tag{78}$$

Thus, the ratio of the two areas is equivalent to the ratio of the two depths squared,

$$\alpha(t) \doteq A(t)/A_0 = \mathrm{z}_0^2/\mathrm{z}^2(t). \tag{79}$$

Knowledge of the area ratio provides an inversely proportional estimate of the target depth. This information can be strategically used for visual servoing instead of an observer-based depth estimation scheme.

This section has presented the necessary details of rigid camera motion, projective geometry, and region integrals in the image plane. With this analysis in hand, the next sections construct measurement algorithms and a control scheme to overcome the pose-dependent disturbances and integral terms.

4.3 Closed-Loop Tracking

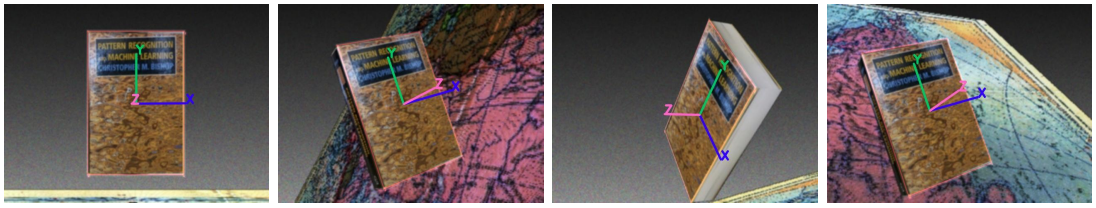


Figure 23: Tracking necessitates segmenting the target and assigning an estimated pose. Shown above are several example frames during tracking; pose is estimated by a nonlinear optimization program.

This section derives a Lyapunov-based controller for visual servoing. Dynamics of the target's image-plane centroid are made input-to-state stable, keeping it centered in the field of view as shown in Figure 23. The role of target *pose* enters via the disturbances and is addressed in § 4.4.4. First, the general visual-tracking control law is derived. Inputs are ω_x and ω_y, with roll-rate $\omega_z = 0$:

$$\begin{aligned} \dot{\bar{x}}_c &= -\omega_y f + \frac{\omega_x \bar{x}_c \bar{y}_c - \omega_y \bar{x}_c^2}{f} + \frac{f v_x^C - \bar{x}_c v_z^C}{\mathrm{z}} \\ \dot{\bar{y}}_c &= \omega_x f - \frac{\omega_y \bar{x}_c \bar{y}_c - \omega_x \bar{y}_c^2}{f} + \frac{f v_y^C - \bar{y}_c v_z^C}{\mathrm{z}} \ . \end{aligned} \tag{80}$$

Proposition 4.3.1 *The control strategy*

$$\omega_x(t) = -\frac{K \bar{y}_c}{f} \ , \quad \omega_y(t) = \frac{K \bar{x}_c}{f} \tag{81}$$

renders the system Eq. 80 GES under zero target velocity. As a consequence Eq. 80 is input-to-state stable when there is bounded velocity target motion.

Proof Consider the candidate Lyapunov function

$$V(x) = \frac{1}{2}(\bar{x}_c^2 + \bar{y}_c^2) \tag{82}$$

Using the previously computed $\dot{\bar{x}}_c, \dot{\bar{y}}_c$, the associated time-derivative for V is

$$\begin{aligned}\dot{V} &= \bar{x}_c\dot{\bar{x}}_c + \bar{y}_c\dot{\bar{y}}_c \\ &= \bar{x}_c\bar{y}_c\omega_z - \omega_y f\bar{x}_c - \bar{x}_c\bar{y}_c\omega_z + \omega_x\bar{y}_c f \\ &\quad + \frac{\omega_x\bar{x}_c^2\bar{y}_c - \omega_y\bar{x}_c^3}{f} + \frac{f\bar{x}_c v_x^C - \bar{x}_c^2 v_z^C}{\mathrm{z}} \\ &\quad - \frac{\omega_y\bar{x}_c\bar{y}_c^2 - \omega_x\bar{y}_c^3}{f} + \frac{f\bar{y}_c v_y^C - \bar{y}_c^2 v_z^C}{\mathrm{z}}\end{aligned} \tag{83}$$

The control input terms ω_x and ω_y are the tilt and pan angular rates. The relative target motion in the camera frame v^C is unknown. Let $\omega_x = -K\bar{y}_c/f$ and $\omega_y = K\bar{x}_c/f$, where K is to be refined later. The time derivative becomes,

$$\begin{aligned}\dot{V} &= -K\left(\bar{x}_c^2 + \bar{y}_c^2\right) - Kf^{-2}\left(\bar{x}_c^2 + \bar{y}_c^2\right)^2 \\ &\quad + \frac{f}{\mathrm{z}}\left(\bar{x}_c v_x^C + \bar{y}_c v_y^C\right) - \frac{v_z^C}{\mathrm{z}}\left(\bar{x}_c^2 + \bar{y}_c^2\right)\end{aligned} \tag{84}$$

with a negative-definite part (the first two terms) and a disturbance part (the last two terms). When $v^C(t) = 0$, then

$$\dot{V} = -K\left(\bar{x}_c^2 + \bar{y}_c^2\right) - Kf^{-2}\left(\bar{x}_c^2 + \bar{y}_c^2\right)^2, \tag{85}$$

and the system Eq. 80 is clearly GES and Lipschitz. When $v^C(t) \neq 0$, then the system Eq. 80 is bounded and Lipschitz when z is bounded from below[2], f is bounded from above, and $v^C(t)$ is bounded and Lipschitz. Since these conditions are met and the image domain is finite, the closed-loop system is Lipschitz and GES when unforced by the disturbance. Under disturbances, it is ISS [136].

Note that in Eq. 84, the disturbance scales according to the distance of the target along the optical axis. This simple visual servoing law will have variable performance depending on the target depth. The attracting invariant set radius will vary with z. It is clear that the invariant set radius can be made independent of z by scaling the gain by a factor of $\mathrm{z}^{-1}(t)$, but the needed range z is not observable with a single projective camera.

[2] $z(t) > \underline{\mathrm{z}} > 0$ holds when the target is in the camera's field of view.

One approach is to use a recursive estimation scheme that provides a depth-estimate [4, 84, 103]. A drawback to this class of solutions is the possibility of introducing non-convergence or instability of the estimator. It is assumed in deriving theoretical stability and convergence criteria for range-estimation schemes that we can correctly find feature-point matches between adjacent frames. This is rarely possible due to minor occlusions, motion-blurring, lighting variation, *etc.* As soon as a feature-matching error occurs, the estimator is apt to behave outside the expected specifications.

Instead of using the target's *range*, it possible to use its *area* as a proxy. Area is directly measured by segmentation of incoming frames. Thus, using it as a modifier on control signals does *not* entail the added complexity and potential convergence problems of an observer being used for estimates of $\mathrm{z}(t)$. However, measurements of area vary with the perspective image warping; in-place rotation of the target will change the observed area. In the next section, the geometry of projective images is explored in order to properly compensate for changes in target range and orientation.

4.4 Area Rectification from Estimated Pose

An image of a planar object will, in general, not have the object plane normal to the optical axis. The area and centroid of an observed target will be warped by a perspective transformation. In-place rotation of the target shrinks the apparent image area as compared to the area in a head-on view. Invariance of the proposed visual servoing controller with respect to depth relies on area ratio measurements as a proxy for relative depth. Thus, it is necessary to rectify observations of the target region so that area scales properly with range alone and not due to the rotation-induced warping.

This section describes how to rectify the imaged object by undoing the apparent effect of estimated rotation (or by applying the estimated translation). As part of the process one must estimate the transformation associated to a planar object given a known image or segmentation template. Compensation requires a computation of the determinant of the transformation's Jacobian.

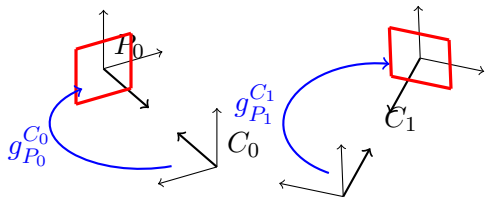


Figure 24: Depiction of frames for template view and new observed view.

4.4.1 Effect of Target Rigid Body Displacement

From an initial camera view of the object to a new camera view of the object, four reference frames exist as per Figure 24. The first pair (C_0, P_0) belong to the camera and the object associated to the initial view, while the second pair (C_1, P_1) describes the new frames of the camera and object configurations.

Assume the initial camera view is such that $g_{P_0}^{C_0}$ consists of no rotation, $R_0 = \mathbb{1}$ and translation along the optical axis only, $d_0 = (0, 0, \mathrm{Z}_0)^T$, with the plane normal parallel to the optical axis. This view provides a head-on view of the planar target. All points on the planar target are described by the coordinates $q^{P_0} = (\mathrm{X}, \mathrm{Y}, 0, 1)^T$ and map to the camera-frame coordinates $q^{C_0} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}_0, 1)^T$.

Projections of points on the planar target to image coordinates for both views are

$$r^0 = \Pi\left(g_{P_0}^{C_0} q^{P_0}\right) = \frac{f}{\mathrm{Z}_0}\begin{pmatrix}\mathrm{X}_0\\ \mathrm{Y}_0\end{pmatrix}, \quad \text{and} \tag{86}$$

$$r^1 = \Pi\left(g_{P_1}^{C_1} q^{P_1}\right) = \Pi\left(g_{P_1}^{C_1} g_{P_0}^{P_1} q^{P_0}\right) = \Pi\left(g_{P_0}^{C_1} q^{P_0}\right). \tag{87}$$

The mapping taking the coordinates r^0 to r^1 has a nice structure, whose determinant provides clues regarding the true area. The following proposition is well known:

Proposition 4.4.1 *Let $g_{P_0}^{C_1} \in SE(3)$ be composed of translation $T = (t_x, t_y, t_z)^T$ and rotation R. Points on the planar object have the form $q^{P_0} = (\mathrm{X}, \mathrm{Y}, 0, 1)$. These 3D co-planar points map to the observed image coordinate r^1 via*

$$P(\mathrm{X}, \mathrm{Y}) = \left(f\frac{R_{11}\mathrm{X}+R_{12}\mathrm{Y}+t_x}{R_{31}\mathrm{X}+R_{32}\mathrm{Y}+t_z}, f\frac{R_{21}\mathrm{X}+R_{22}\mathrm{Y}+t_y}{R_{31}\mathrm{X}+R_{32}\mathrm{Y}+t_z}\right) \tag{88}$$

with associated Jacobian determinant

$$\det(DP) = f^2(R_{31}\mathrm{X} + R_{32}\mathrm{Y} + t_z)^{-3} \cdot \Big((R_{11}R_{22} - R_{12}R_{21})t_z + (R_{12}R_{31} - R_{11}R_{32})t_y + (R_{21}R_{32} - R_{22}R_{31})t_x\Big). \quad (89)$$

Derivation The template is a planar patch in 3D; all of the z-coordinates on the patch in the camera frame are at Z_0. It is translated by $-\mathrm{Z}_0$ to align the two frames, then rotated about its centroid. As the z-coordinates are all zero, this nullifies the third column of R. The patch is then translated by (t_x, t_y, t_z) in the camera frame to the location from which it is imaged. Expanding the transformation of X, Y, we see that $P(\cdot)$ is a specialized case of $\Pi(\cdot)$ where the domain is a coplanar, camera-axis-aligned set of points:

$$r^1 = \Pi\left(g_{P_0}^{C_1} q^{P_0}\right) = \Pi\left(\begin{pmatrix} R_{11}\mathrm{X} + R_{12}\mathrm{Y} + t_x \\ R_{21}\mathrm{X} + R_{22}\mathrm{Y} + t_y \\ R_{31}\mathrm{X} + R_{32}\mathrm{Y} + t_z \\ 1 \end{pmatrix}\right) = P(\mathrm{X}, \mathrm{Y}) \quad (90)$$

Algebraically computing the differential of P with respect to X and Y gives the determinant. As one might expect, Eq. 89 yields a zero determinant for rotations of $\pm\frac{\pi}{2}$. This corresponds to a degenerate mapping of the planar patch into a line; it is assumed that target motion is sufficiently slow such that successful closed-loop tracking prevents this degeneracy.

4.4.2 Area Integral Compensation

Generation of the area ratio will require one of two potential image pairings, both of which are depicted in Figure 25. There exists a transformation g whose perspective transform P maps from the template view (top left) to the current view (bottom right). One may consider the area ratio between the template and a translated version, or between the current segmentation and a rotated version of the template (area ratios occur between a lower element and the corresponding upper element in the diagram).

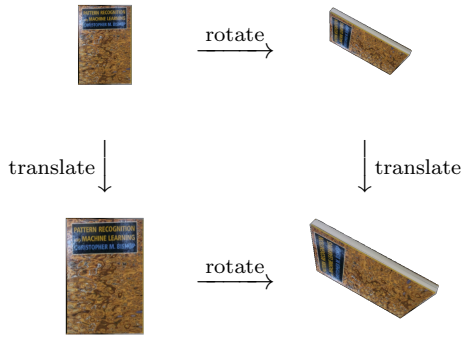


Figure 25: Commutative Diagram: Translation and Rotation
Commutative diagram: template view is translated and rotated and projected to the 2D image plane.

As in § 4.2.3, the image domain is Ω with the target region given by $\mathcal{S} \subset \Omega$. Let the head-on image of the target lead to Ω_0 and $\mathcal{S}_0$. Through a direct application of the change-of-variable theorem and the Jacobian Eq. 89, the area of the target in domain Ω_0 is

$$A(t) = \int_{\mathcal{S}} dx_c dy_c \tag{91}$$

$$= \int_{\mathcal{S}_0} \frac{1}{f^2} det(P\,(x_c/f, y_c/f)) dx_c dy_c \ . \tag{92}$$

To obtain the area ratios, define P_T/P_R to be the perspective transformation arising from only applying the translation/rotation part of g. The measurement α results from either one of the following computations

$$\alpha(t) = \frac{f^2 \int_{\mathcal{S}} dx_c dy_c}{\int_{\mathcal{S}_0} det(P_R(x_c/f, y_c/f) dx_c dy_c} \tag{93}$$

or

$$\alpha(t) = \frac{\int_{\mathcal{S}_0} det(P_T(x_c/f, y_c/f)) dx_c dy_c}{f^2 \int_{\mathcal{S}_0} dx_c dy_c}. \tag{94}$$

It is important to note that the camera focal length and the target distance in the template view need not be known precisely. The area ratio cancels out any scale ambiguity introduced by errors in both values. The controller derived in Section 4.3 need only use one of the ratios to achieve invariance with respect to target rotation. Which to use will depend on the estimation algorithm utilized, as each algorithm has differing sensitivities in R and T.

4.4.3 Centroid Integral Compensation

In a similar manner, the target centroid values can be compensated to be unaffected by in-place rotation and perspective skew,

$$\bar{x}_c(t) = \frac{\int_{\mathcal{S}_0} \left\langle (1,0)^T, P_T(x_c/f, y_c/f) \right\rangle dx_c dy_c}{\int_{\mathcal{S}_0} dx_c dy_c}, \tag{95}$$

$$\bar{y}_c(t) = \frac{\int_{\mathcal{S}_0} \left\langle (0,1)^T, P_T(x_c/f, y_c/f) \right\rangle dx_c dy_c}{\int_{\mathcal{S}_0} dx_c dy_c}\ . \tag{96}$$

4.4.4 Modified Gain due to Pose

Define the time-varying gain $K(t) \doteq K_0 \alpha^{\frac{1}{2}}(t)$. The variable gain has a beneficial effect on the time derivative of V.

Proposition 4.4.2 *Given that the area ratio $\alpha(t)$ between the current imaged target and a template version viewed at a fixed, constant distance is available, then the control strategy Eq. 97-Eq. 99:*

$$\omega_x(t) = -\frac{K(t)\bar{y}_c}{f} \tag{97}$$

$$\omega_y(t) = \frac{K(t)\bar{x}_c}{f} \tag{98}$$

$$K(t) = K_0 \alpha^{\frac{1}{2}}(t) \tag{99}$$

modifies Theorem 4.4.2 such that the region of stability is invariant to depth for known bounds on target velocity.

Proof Since from Eq. 79 area ratio α is related to depth by $\alpha^{\frac{1}{2}} = \mathrm{z}_0/\mathrm{z}(t)$, Equation Eq. 85 becomes:

$$\dot{V} = \frac{1}{\mathrm{z}}\Big[-K_0 \mathrm{z}_0 \left(\bar{x}_c^2 + \bar{y}_c^2\right) - K_0 \mathrm{z}_0 f^{-2} \left(\bar{x}_c^2 + \bar{y}_c^2\right)^2 + f\left(\bar{x}_c v_x^C + \bar{y}_c v_y^C\right) - v_z^C \left(\bar{x}_c^2 + \bar{y}_c^2\right)\Big], \tag{100}$$

meaning that the depth is not a consideration when seeking to dominate the disturbance for estimating the invariant set radius.

Since the $\alpha(t)$ term is a function of the directly measured area, using it as a modifier on control gains does *not* entail the added complexity and potential convergence problems of an observer estimating $\mathrm{z}(t)$.

4.4.5 Region of Convergence Size via K_0

Define the pixel-centroid and disturbance velocity as $\mathbf{x} \doteq (\bar{x}_c, \bar{y}_c)^T$ and $\mathbf{v} \doteq (v_x^C, v_y^C)^T$, respectively. Suppose that the depth-change disturbance has bound $|v_z^C| \leq v_L$ while the $\mathbf{v}$ disturbances have the bound $\|\mathbf{v}\|_2 \leq v_M$. Then, Equation Eq. 85 rewritten below,

$$\dot{V} = -K\|\mathbf{x}\|_2^2 \left(1 + \|\frac{\mathbf{x}}{f}\|_2^2\right) + \frac{f}{\mathrm{z}} \langle \mathbf{x}, \mathbf{v} \rangle - \frac{v_z^C}{\mathrm{z}} \|\mathbf{x}\|_2^2, \tag{101}$$

simplifies further, using the velocity bounds, to

$$\dot{V} \leq \frac{1}{\mathrm{z}} \Bigg[- K_0 \mathrm{z}_0 \|\mathbf{x}\|_2^2 \left(1 + \|\frac{\mathbf{x}}{f}\|_2^2\right) + f v_M \|\mathbf{x}\|_2 + v_L \|\mathbf{x}\|_2^2 \Bigg] . \tag{102}$$

which follows from Cauchy-Schwarz and the definition of K in Eq. 97. Using this gain enables the set $\{\mathbf{x} \mid \dot{V}(\mathbf{x}) \leq 0\}$ to depend only on K_0 and $\mathbf{x}$ itself; the z scales the rate but cannot change the sign of $\dot{V}$.

Let $\rho = \|\mathbf{x}\|_2$ denote a critical pixel coordinate radius outside of which $\dot{V}$ should be guaranteed negative. Choosing a $K_0(\rho)$ to fall within this range is straight-forward via

$$K_0 = \frac{f v_M + \rho v_L}{\mathrm{z}_0 \left(\rho + \rho^3 / f^2\right)} . \tag{103}$$

Thus we have a design procedure for selecting the nominal gain K_0 so as to achieve a particular pixel-coordinate error norm. Stability exists for ρ within the image bounds while picking ρ near 0 leads to a high gain that is not desirable due to high-frequency unmodelled dynamics[3].

An example plot of the gain versus the desired error performance radius is shown in Figure 26; application-specific bounds on the target velocities and the required centroid

[3]Such as pixel coordinate quantization and camera platform jitter.

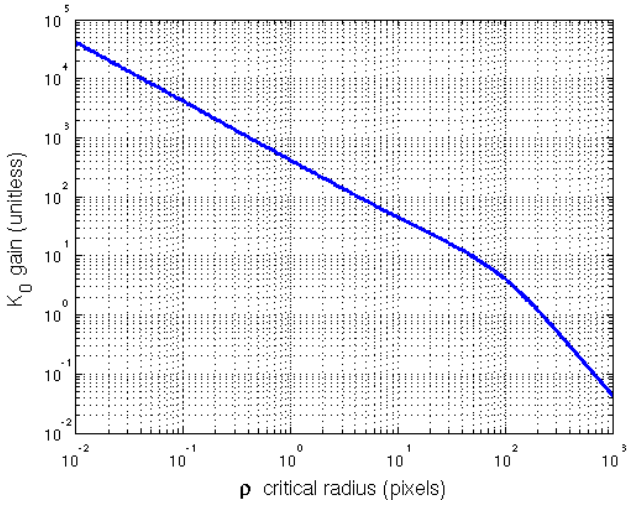
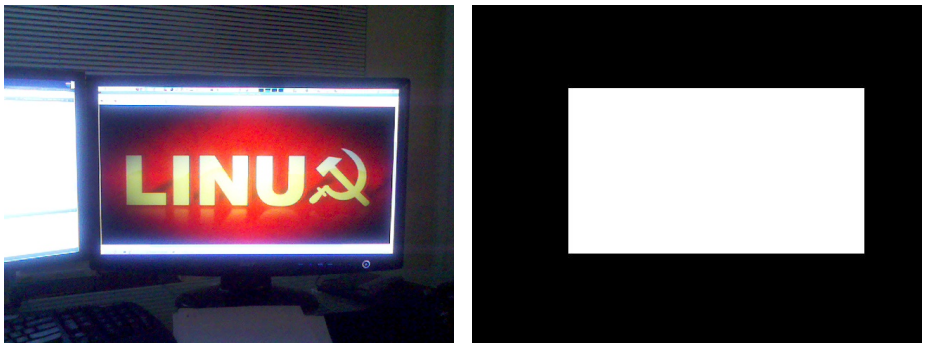


Figure 26: Gain K_0 can be chosen so as to guarantee a particular radius ρ of the set where $\dot{V} \leq 0$. Here, $v_M = 20.0$, $v_L = 20.0$, $z_0 = 5.0$, and $f = 100$.

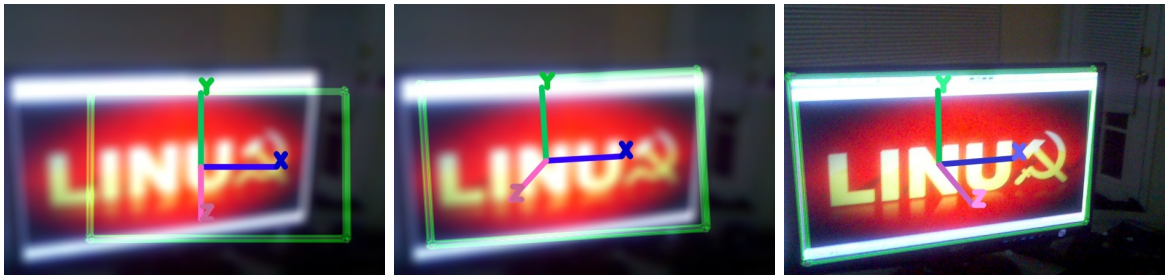
bound, ρ, drive the placement of K_0 along the curve. Of course, maintaining this error bound under range and rotation changes over time requires pose-rectified area measurements. Rectification of the observed region is achieved using modified surface-integrals that take into account $P(x, y)$, the projective mapping of planar 3D coordinates into the 2D image. Given template region and warped regions, there are several ways in which the perspective effect can be estimated; these are considered next in Section 4.5.

4.5 Pose Estimation Algorithms

Implementing pose-invariant visual servoing as presented in the preceding sections requires measurements of the relative pose between camera and target. The relative pose, g_O^C, is decomposed into $R = \exp(\hat{\omega})$ and T. Visual servoing strategies to-date have focused on *feature correspondence* and *homography decomposition* methods for obtaining such data. Alternatively, one can use *region-based* methods that iteratively minimize a parametrized function. In practice this yields a more robust and accurate estimate of pose, but with more computational overhead. This section introduces and compares the potential pose-estimation algorithms in the context of closed-loop tracking.

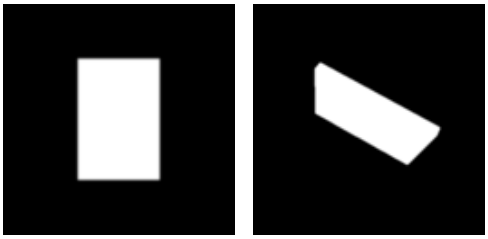


(a) Template image. (b) Region-of-interest mask.



(c) Several frames during minimization, in scale-space [27, 32]

Figure 27: An illustration of image-template-based pose estimation. The initial pose estimate is placed far from the correct solution for visual clarity. Several intermediate frames show the template region-of-interest aligning with the new observed $\mathbb{I}(t)$.



(a) ϕ_0 (b) ϕ_1

Figure 28: Segmentation-based pose estimation assumes that the target region in $\mathbb{I}(t)$ has been denoted by indicator function ϕ_1. It then seeks to parametrically map the template's ϕ_0 to ϕ_1.

4.5.1 Template-Based Estimation

Template-based tracking utilizes the template image to generate a reference image patch defined over a (strict) sub-domain of the image domain, $\mathcal{S}_0 \subset \Omega_0$. The patch $\mathbb{T} : \mathcal{S}_0 \to \mathbb{R}$ must be warped under P to align with a patch within the new image $\mathbb{I}(t)$. The target in the current image is located by minimizing the scalar objective function

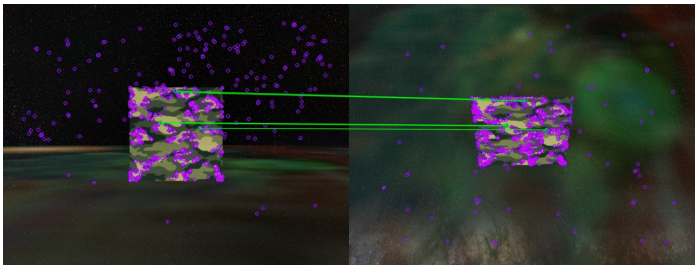
$$E(\omega, T) = \mathbf{d}(\mathbb{T} \circ P^{-1}, \mathbb{I}(t)) \ . \tag{104}$$

Optimizing over ω and T matches the pixel intensities of the template to the pixel intensities of the image. Template pixel intensities at the coordinates $(x_c, y_c)^T \subset \Omega_{\mathbb{T}}$ are compared to the current image pixel intensities at the perspective transformed coordinates $(x'_c, y'_c)^T = P(x_c, y_c)$. A simple choice of $\mathbf{d}$ is given by taking the sum-of-squared-differences of image values. However, this cost function is sensitive to noise and, in particular, illumination changes. Incorporating a region-based cost when defining Eq. 104, such as the difference between image means inside and outside of $\mathcal{S}(t)$ [23], typically improves robustness and convergence from poor initialization. An example of template-based pose estimation is shown in Figure 27. $\mathbb{T}$ is defined as as the product of the full template image frame and a region-of-interest mask. Several iterations of minimizing $E(\omega, T)$ are shown in Figure 27c. A distinguishing characteristic of the template-image-based approach is that one solves for pose parameters and target region simultaneously; the region $\mathcal{S}(t)$ in $\mathbb{I}(t)$ is implicitly defined by the range to which P maps $\mathcal{S}_0$.

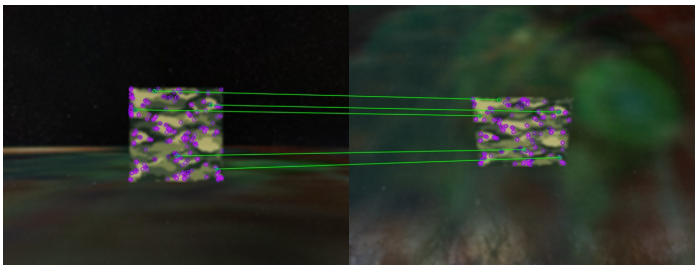
It is possible that photometric data in the template is not trusted; for example, when strong changes in illumination are expected. Then, the special case of *segmentation-based tracking* can be employed. Segmentation-based tracking decouples boundary-contour extraction from pose-estimation entirely. First, the boundary of the target over time is followed by a black-box function that provides segmentations at each frame over time. These contain an *indicator function* $\phi : \Omega \to \mathbb{R}$ evaluating to unity for target points and zero for background points. Segmentation is achieved by minimizing a function encoding for the error in the current segmentation and for violation of any prior information. After the segmentation of $\mathbb{I}(t)$ is computed, the following optimization solves for the parametric warp $P : \Omega_0 \subset \mathbb{R}^2 \to \Omega \subset \mathbb{R}^2$ that aligns a template indicator function $\phi_0 : \Omega_0 \to \mathbb{R}$ (obtained from segmenting the template image) to the one currently observed $\phi_1 : \Omega \to \mathbb{R}$ (obtained from segmenting the current image). Define the following cost function:

$$E(\omega, T) = \frac{1}{2} \int_{\Omega_1} \left(\phi_1 - \phi_0 \circ P^{-1}\right)^2 \, dudv \;, \tag{105}$$

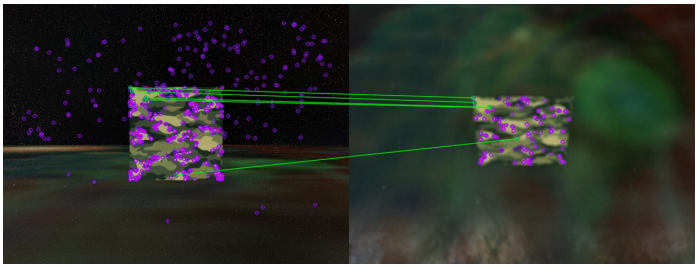
which uses the inverse map since the forward map sends Ω_0 to Ω_1. Two example indicator functions are shown in Figure 28 corresponding to the template ϕ_0 and warped observation



(a) Sharp Template and Observation



(b) Blurred Template and Observation



(c) Sharp Template, Blurred Observation

Figure 29: Feature-based pose estimation is quite sensitive to imaging conditions and repetitive textures. Notice that slight changes in sharpness of the image can radically alter the number of features found by SURF [10]. In the above example images, RANSAC-matching [26,124] a template to an observation of varying motion-blur can result in large pose-estimate errors due to wrong matches.

ϕ_1.

4.5.2 Homography-Based Methods

A well-known [51, 85] expression gives the composition of a planar homography matrix H from the rigid transformation of a planar point-set:

$$H = \lambda \left(R + \frac{1}{\mathrm{z}_0} T N^T \right) \in \mathbb{R}^{3x3} , \tag{106}$$

where $N = (0, 0, 1)^T$ is the normal vector of the template planar patch and λ is a free parameter. Since $\mathrm{z} = 0$ in the object frame, the homography matrix is related to the

previously defined projective map $P(\cdot)$ by

$$\lambda \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} = \underbrace{K\left[\mathbf{r}_1, \mathbf{r}_2, T\right]}_{H} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix} , \tag{107}$$

with *intrinsic-parameter* or *camera-calibration* matrix K and rotation matrix columns $\{\mathbf{r}_1, \mathbf{r_2}\}$. Note that we assume the image origin to to be centered and pixels to have $1:1$ aspect ratio, hence K is diagonal with entries $(f, f, 1)$.

Estimates of H, denoted by $\hat{H}$, can be generated from feature-point correspondence. However, estimates of the constituent R and T from *de*-composition of $\hat{H}$ are often numerically unstable in practice. Algebraic analysis suggests several possible solutions arising from the singular-value decomposition of H, of which only two have valid (positive) depth [51,85]. However, pixel coordinate quantization and point-to-point correspondence errors mean that the available measurement $\hat{H}$ will in fact *not* have the structure of Eq. 106. Instead, $\hat{H}$ will deviate from H to give $\hat{H} = H + e_H$, where the error term e_H causes a violation of algebraic assumptions in recovering pose from homography; see for example [5, 45] for mathematical analysis of the null-space perturbation problem.

Linear estimation methods from point-correspondence are at least an order of magnitude faster than the nonlinear region-based optimization approaches noted in § 4.5.1 but are numerically unreliable. Even when K is precisely known, the presence of e_H means that the first two columns of $K^{-1}\hat{H}$ are not orthonormal; the result does not contain valid entries of a rotation matrix. The error will scale dramatically depending on lighting conditions, camera focus, and erroneous point-matches; an example using a camouflage-painted target is shown in Figure 29.

The consensus in existing literature on homography-decomposition suggests first normalizing $\hat{H}$ due to the unreliable $\hat{KR}$ sub-matrix. However, for the proposed control system, it is preferred to *not* normalize $\hat{H}$ and thus preserve the measured $\hat{KT}$ sub-matrix. The determinant of Eq. 89 can be used in the form $det(P_T(\mathrm{X}, \mathrm{Y}))$ wherein the R is set to the

identity matrix. When speed is of the utmost concern, this approach for computing area-ratio $\alpha(t)$ can be employed in lieu of more accurate region-based methods. The estimated translation for generating area ratio $\alpha(t)$ and centroids $\{\bar{x}_c(t), \bar{y}_c(t)\}$ is given by

$$\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \doteq \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \hat{H} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} . \tag{108}$$

4.5.3 A Note on the Methods

The field of visual tracking in computer vision has a history of studying both template-based and segmentation-based tracking [150], and more generally alignment by parameterized warps [7]. There are a large variety of tracking algorithms optimized for these techniques [32,40,90]. By casting the visual servoing framework to rely on template-alignment, we can leverage the extensive literature associated to robust, high performance tracking algorithms for its computational implementation.

Consistent feature-point detection and matching is difficult to attain, unless the environment is tightly controlled. To use feature-based tracking, it is typically necessary to first detect salient features, e.g. using the SURF descriptor [10]. A subset of similar features in both template and new observed image can be found via RANSAC [26,124]; a subset of detected points must map via a single transformation.

4.6 Simulation Results

Several test scenarios demonstrate the combined tracking and pose-estimation method. Pose-invariant visual servoing is implemented in a closed-loop 3D simulator environment. Measurements are extracted from rendered images and used to update camera pose. Analytically predicted invariance properties of tracking error are verified. That is, the proposed time-varying gain using a rectified area measurement keeps image-plane tracking error within a fixed bound that depends only on target velocity magnitude. In comparison,

a gain that is constant or uses a warped area value causes strong fluctuations in image-plane tracking error. Lastly, we illustrate several tradeoffs in selecting a feature-based or template-region-based pose estimation algorithm.

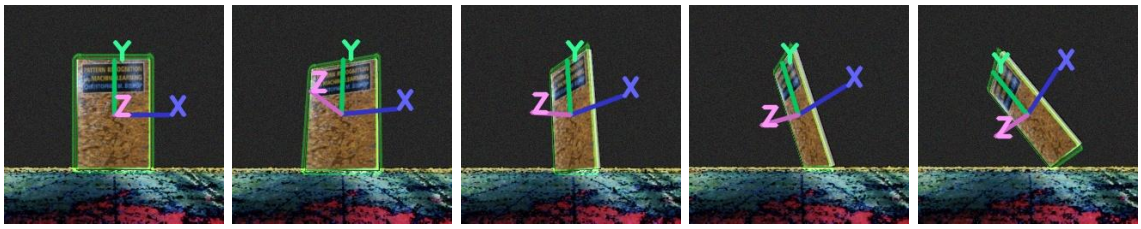
4.6.1 Open-Loop Pose Estimation

To demonstrate some of the computational steps and the improvements possible for moving towards alternative pose estimation methods, a experiment was performed utilizing a planar region undergoing pure rotations only (constant translational displacement). Area estimation, pose estimation, and computational aspects were investigated.

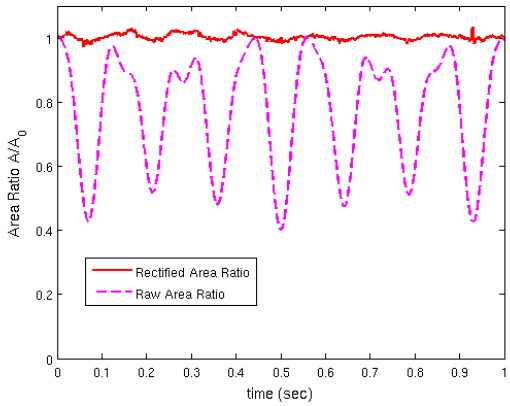
Area Estimation: The area of the target in the image plane was computed and compared to the area of the target as computed via Eq. 94. The compensated area remains fairly constant with some fluctuation due to estimation errors. The estimation errors arise from the pixel quantization effects associated to images, continuously changing measurements are not possible. The simulation provides confidence that the image-based estimates are sufficiently accurate for computation of the area integral.

Pose Estimation: Errors in ω rotation estimates for this sequence are plotted in Figure 31. Feature-based pose estimation gives a much larger error than the region-based algorithm due to the non-distinct texture of the target. RANSAC matching of the detected features will often find some set of points and transformation that does not respect the region-boundary of the target in $\mathbb{I}$; for example, this occurs in the two frames shown in Figure 31c. When the pose is close to the identity, or if lighting and pixel quantization aligns to the target in a fortunate way, the RANSAC set gives an accurate pose estimate, and in the matches for the two frames found in Figure 31b.

Computation: It is natural to ask whether the formulation of region-based pose estimation is necessarily too slow due to computational burden, despite the benefits in accuracy. Average execution time for representative image sizes is shown in Figure 32. Iterative minimization to implicitly yield a pose estimate incorporates the BOBYQA routine [28, 123]



(a) Pure rotation: images



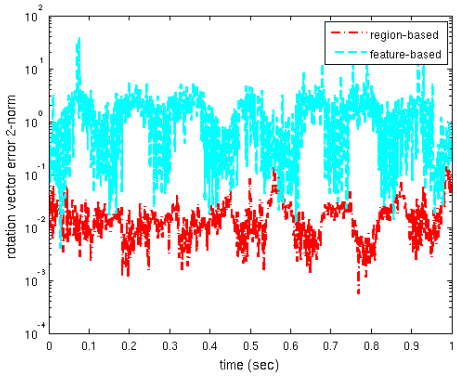
(b) Pure rotation: area ratio

Figure 30: Under pure rotation of the target, direct computations of area fluctuate significantly. The compensated area integral keeps the ratio A/A_0 close to unity.

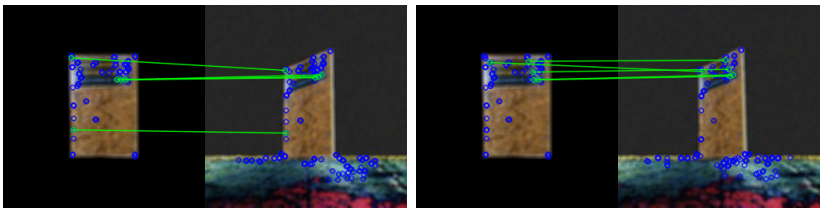
to replace error-prone and compute-intensive analytic gradients. During the iterative optimization process, quadratic approximations $\hat{E}$ to the objective function $E(\omega, T)$ are periodically recomputed; performing the necessary evaluations of $E(\cdot)$ to create $\hat{E}$ in parallel dramatically speeds up the solver. Because our application of interest is *tracking*, a close initialization is typically available; computation time is significantly reduced. Note that a single-frame delay is not the same as the system throughput; an application can still achieve high frame-rate by pipelining multiple computation threads [96].

4.6.2 Depth-Invariant Servoing

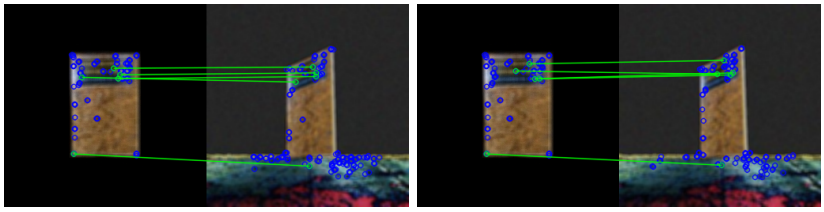
To verify depth-invariance in closed-loop operation, the camera is placed at a fixed location in space while the target moves along a sinusoidal trajectory that approaches then recedes from the camera as depicted in Figure 33b. The velocity limits here correspond to the $K_0(\rho)$ plot Figure 33a, where $v_M = 60.0$, $v_L = 2.0$, $z_0 = 5.0$, and $f = 225$. A gain $K_0 = 300.0$ is chosen such that critical radius of approximately 10.0 pixels is expected.



(a) Pose Estimation: $\|\hat{\omega} - \omega\|_2$ error



(b) Feature-Based Pose Estimates: Correct Matches



(c) Feature-Based Pose Estimates: Wrong Matches

Figure 31: Template-region-based pose estimation gives a much tighter error bound due to frequent incorrect correspondences in feature-based pose estimation.

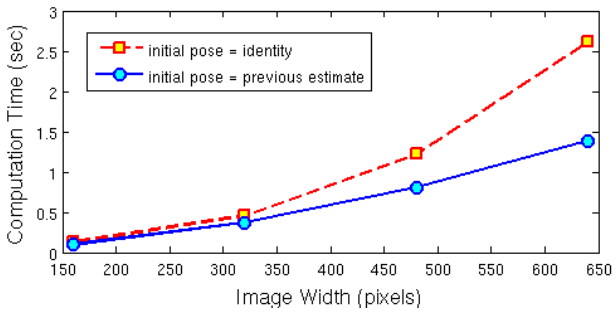
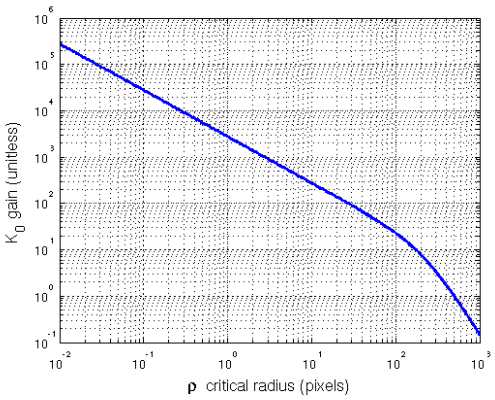
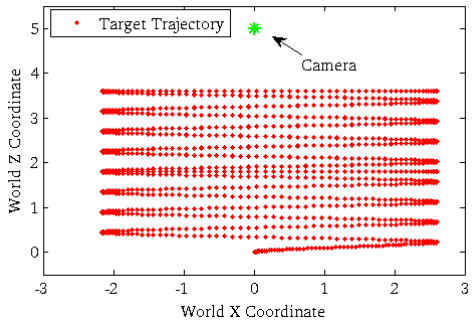


Figure 32: Average execution time for region-based posed estimation, with identity and previous frame's estimate as initialization. Images used are all of $4:3$ aspect ratio, with widths $640, 480, 320$, and 160.

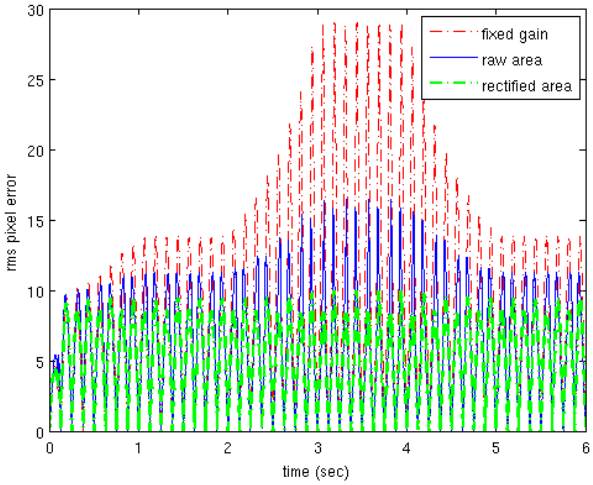
Centroid error norm over time is plotted in Figure 33c. Note that the commonly used



(a) $K_0(\rho)$ curve



(b) Target approaches then recedes from camera.



(c) Centroid error 2-norm during tracking.

Figure 33: Performance comparison: using the area-rectified $K = \alpha^{1/2}K_0$ law (green) gives a bound on the image centroid that is independent of range and target pose, for given velocity bounds. The tighter bound is particularly noticeable when the target approaches the camera. Images are 128×128 pixels.

fixed-gain visual servoing strategy has depth-varying performance while the proposed control algorithm has a consistent error bound. Variances of the centroid norm are 40.6, 15.0, and 7.9 when using fixed gain, raw observed area, and rectified area, respectively. Snapshots of the simulated visual servoing task can be found in Figure 34. The changing depth and orientation can be clearly seen. A planar ground surface is also depicted in the background to help visualize the camera pan and tilt movements during tracking.

As expected analytically, servoing with a fixed pan/tilt gain leads to error that varies dramatically with range. A time-varying gain using rectified area makes the positive-invariant

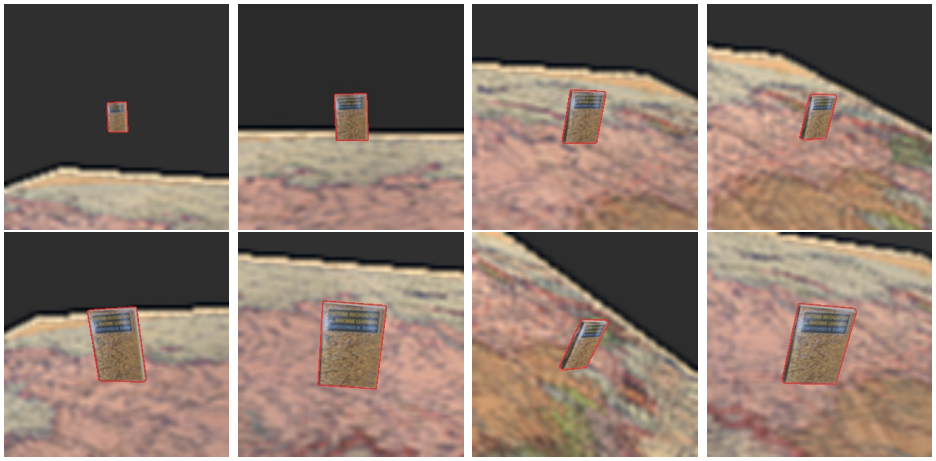


Figure 34: Snapshots of fixed-zoom range-invariant target tracking. Available at http://www.youtube.com/watch?v=ssoixPyYaPQ.

set radius independent of range and in-place target rotation. Implementation of the invariant control scheme can employ either feature-based, segmentation-based, or template-based pose algorithms of Section 4.5. Under favorable imaging conditions, both will enable pose-invariant closed-loop tracking. Feature-based estimation is generally faster computationally but suffers from lack of robustness due to lighting variation, template modeling error, and partial occlusion.

4.7 Conclusion

The proposed closed-loop tracking algorithm uses a time-varying pan/tilt gain to enable performance independent of range and object pose. Several methods for obtaining the necessary pose and rectified area measurements have been introduced. Experimental verification confirms that the controller exhibits thee desired invariance properties. Most accurate segmentation and pose estimation performance is attained with the template-region-based class of algorithms.

Invariant servoing enables a systematic approach to parameter selection for closed-loop tracking. Since the image-plane error depends only on target velocity, it is straightforward to determine whether a camera system with particular focal length f and maximum target velocity $\mathbf{v}$ can keep the target in the field of view. Otherwise, the system would use excess control effort for a distant target, and possibly lose track if it came closer. Selecting a

nominal gain to achieve a desired pixel-error bound is readily done with a $K_0(\rho)$ curve, such as Figure 26

Given the current state of computer speed and parallel-processing ability, it is suggested that other visual-servoing applications may find it no more computationally costly to use image-based parameter optimization instead of the feature-matching paradigm. In the proposed pose-invariant servoing running on a desktop machine, the rotation-estimate error norm less than 0.05 radians is consistently obtained for tracking in 320×240 images in an average of 0.37 seconds on a desktop machine. Phrasing the pose and area measurement as a concise optimization problem also leads to a better a-priori understanding of which imaging conditions could increase output error and removes the ad-hoc algorithmic steps of feature-based routines.

CHAPTER V

AUGMENTED CURVE EVOLUTION IN VISION-BASED CONTROL

5.1 Introduction and Problem Scope

A method of improving performance in closed-loop target tracking using level set segmentation is developed. Unmanned aerial vehicle or next-generation missile guidance are the primary applications of interest. Previous work has taken a point of view wherein algorithms extract measurements from imagery, while control is the use of these measurements to generate signals driving physical actuators. This approach decouples image processing from control entirely. However, a side effect of a decoupled approach is that computer vision algorithms do not have knowledge of the physical system state or control signals being applied during image capture. The contribution of this work is illustrated by the feedback diagram Figure 35.

Failing to take state and control signals into consideration negatively affects vision algorithms by reducing robustness and increasing delay due to computational complexity; Figure 36 and Figure 37 illustrate two common failure modes in a simulated tactical video. Avoiding these failure modes is nontrivial for the reasons summarized in Table 1. Unlike videos taken in an office setting, outdoor tactical imagery is not very amenable to algorithms based on matching feature points.

In the present work, a new model of the vision-based closed-loop tracking problem is

Table 1: VBC issues that make tracking difficult.

1. Disturbances influence not only the camera position and orientation but the image, e.g. due to motion blur.
2. Increasing the time during which the segmentation evolves will lead to more frame drops and even larger apparent frame to frame displacements.

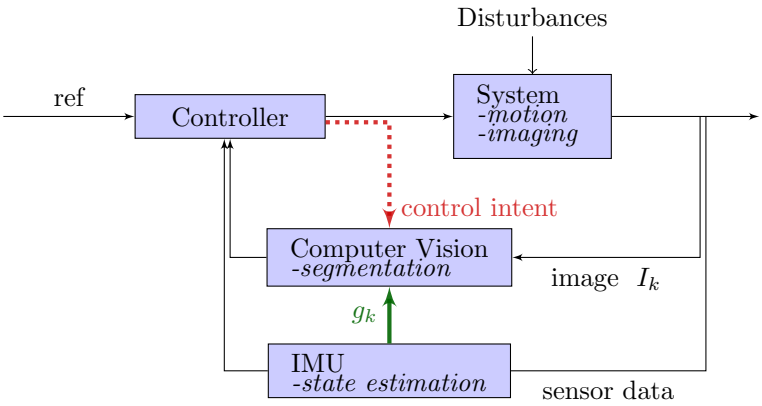
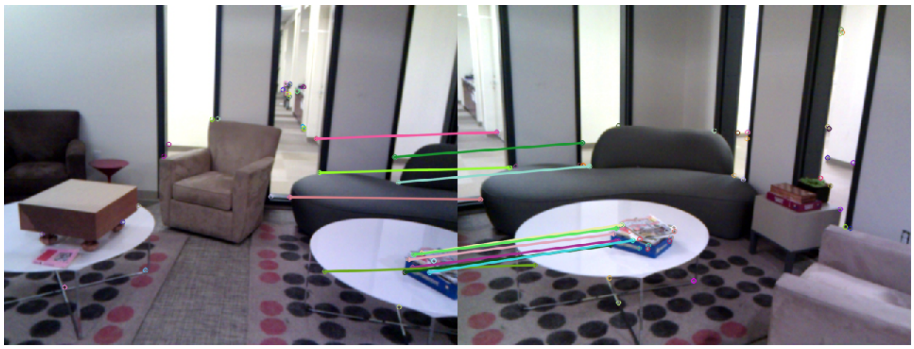


Figure 35: Block diagram illustrating the scope of this paper. The solid green and dotted red signals are considered in § 5.2.2 and Section 5.3, respectively. Making the vision subsystem aware of control intent prevents induced track loss when a fast-moving target is tracked in the presence of ego-motion disturbances.

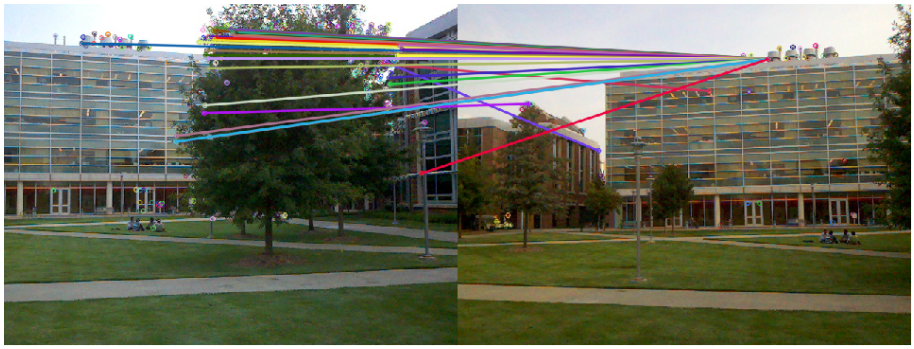
proposed: the algorithm responsible for segmentation of the target is viewed as a separate controllable subsystem rather than merely a measurement source. This proposed technique is quite different from existing "sensor fusion" techniques. Traditionally, data obtained from physical sensors (IMU) and computer vision are treated as two independent measurement sources.

A number of papers have considered ways to fuse measurement sources, for example by multi-rate extended Kalman filters; however, the underlying behavior of the image algorithms remains unchanged. [72,83,89,107]. Many techniques are published for improving the robustness and reliability of measurements extracted from image processing, for example a coarse-to-fine strategy for feature registration [14] and "distribution fields" for synthesizing an objective function with a wide basin of attraction [132]. A related problem of maintaining visibility has been studied; [33] stabilize pose in $SE(3)$ in a manner that prevents loss of feature points due to limited view angle and self occlusion. Similarly, leader-follower formations in obstacle filled environments is studied in [111].

A more closely related work is [60], which formulates an a KLT tracker initialized using measurements from a 3-axis gyro. Integrating the gyroscope measurements between image frames provides an approximate frame-to-frame rotation R_t^{t+1}. As expected, initially transforming the warp parameters p_0 by R_t^{t+1} enables convergence of the "inverse compositional



(a) indoor scene, controlled lighting (Willowgarage Office)



(b) outdoor scene, texture and lighting variability (Biotech Quad)

Figure 36: Feature matching is difficult for many scenarios of practical interest due to motion blur and lack of unique robust features in the scene.

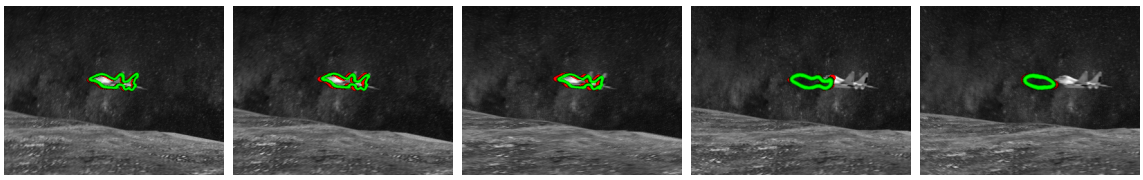


Figure 37: Loss of track due to jitter. Running an algorithm for more iterations is not satisfactory because the computational delay and hence displacement between frames are increased.

image alignment" to a correct correspondence between image features in frames I_t and I_{t+1}, even when the feature points move many ($n > 20$) pixels due to camera rotation. Unlike [60], the present work uses image segmentation algorithms that can successfully track in environments with few or no discernible "features". Additionally, the ramifications of IMU-augmented image processing on closed-loop tracking are considered here both in the theory and experiments. Experiments in [60] for open-loop tracking evaluate the loss of track at sufficient angular rates and residual pixel distances after IMU compensation. The present work considers not only physical angular rate but also the apparent frame to frame angular

Table 2: VBC problem assumptions.

1. Exact geometry of the target is unknown. Distance from the camera is between $z_{\min}$ and $z_{\max}$.
2. Measurements of frame to frame Euclidean transform g_k and intrinsic parameter matrix K of the camera are available,
3. A sensible discriminative model exists in the form of a level set PDE that will correctly segment the target given stationary camera and target motion slow relative to frame rate.

displacement, which is affected by computation time in the segmentation subsystem.

Exogenous Data for PDE Feedback: In the proposed model, image segmentation algorithms are not viewed as black box measurement sources; rather, they are reformulated under the assumption of access to physical sensor data in addition to the video stream. The scope is restricted to algorithms that iteratively minimize a functional to localize the target in each image frame. We view the nominal iterative algorithm as an open-loop system; IMU data is then used to generate a control to modify the algorithm's behavior and thus create a closed-loop system. Figure 38 illustrates the process. The assumptions of the present work are summarized in Table 2.

It is vital to understand the key difference between estimation/fusion algorithms based on filtering and the proposed technique. In filtering, a predicted state and a measurement are simultaneously acquired and merged into a final state estimate; in contrast, the present work (as well as [60]) considers the use of a prediction step to extract a more reliable state measurement from image data.

5.2 IMU-Compensated Initialization (Type I)

Known motion controls acting on the camera platform are exploited here to directly affect contour motion via an affine transformation. Doing so enables active contour performance for tracking to be robust in the presence of physical control signals and disturbances. As illustrated in Figure 39, tracking failure due to fast camera motion can avoided by an

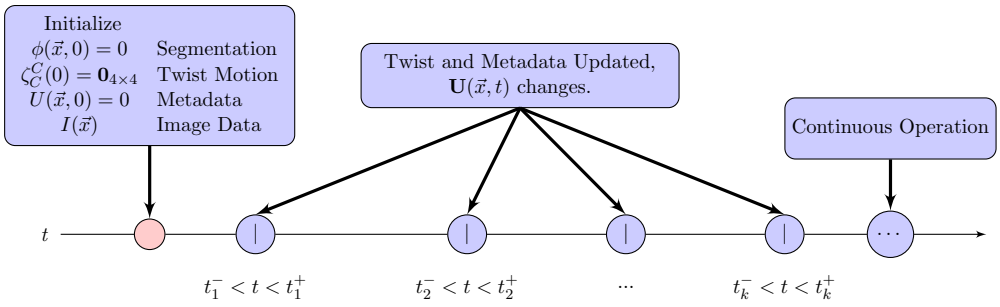


Figure 38: VBC Algorithm Timeline.

appropriate compensation scheme. A less obvious advantage of affine compensation is that closed-loop performance improves due to reduced computational delay.

$\phi(\mathbf{x},t)$ is the segmentation state. New images and meta-data available at times t_k. $D(\mathbf{x})$ is a compensated initialization due to data from physical sensors; in principle D could also come from image registration at the cost of additional delay and source of potentially large error. The segmentation system is

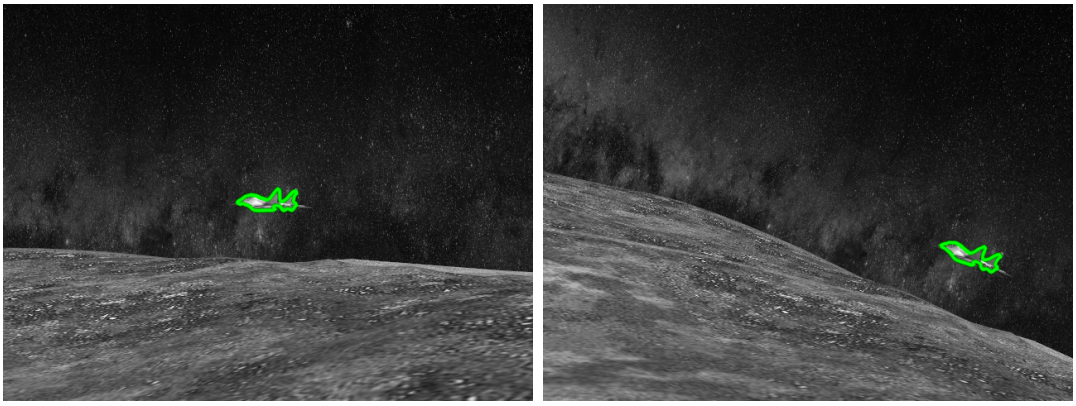
$$\phi_t = \begin{cases} \delta_\epsilon \circ \phi\left[G(\phi, I_k) + \lambda\kappa(\phi)\right] & \text{if} \quad t_k \le t \le (t_k + \mathrm{T}) < t_{k+1} \\ 0 & \text{if} \quad (t_k + \mathrm{T}) < t < t_{k+1} \end{cases} \tag{109}$$

$$\phi(D(\mathbf{x}), t_k) = \phi(\mathbf{x}, t_{k-1}) \ . \tag{110}$$

During the time interval between t_k and $t_k+\mathrm{T}$, the segmentation PDE is active. A temporal gap up to t_{k+1} explicitly models the overhead of data transfer between the algorithm and image stream.

Let g_k be the measured Euclidean transformation between camera poses during the reception of images I_{k-1} and I_k. An affine warp to obtain an initialization $\phi(\mathbf{x}, t_k)$ is then generated as a change of coordinates from $D(\mathbf{x}) \doteq \mathbf{y}$ to $\mathbf{x}$:

$$D(\mathbf{x}) = \Pi_{f,z_0} \circ \left[g_k^{-1} (\Pi_{f,z_0}^{-1} \circ \mathbf{x}) \right] \ . \tag{111}$$



(a) I_{k-1} (b) I_k

Figure 39: Yaw & roll, before and after. Starting from the initial pose (*left*), yaw and roll transform the camera pose. Warping the contour in response to IMU measurement g_k enables a satisfactory initialization in the current image I_k. When $\hat{\nu}_k$ is identically zero, uncertainty in depth does not affect the result.

g_k is decomposed further via $\hat{\omega}_k$ and $\hat{\nu}_k$:

$$g_k = \exp \begin{pmatrix} \hat{\omega}_k & \hat{\nu}_k \\ 0 & 0 \end{pmatrix}, \; \hat{\omega}_k \doteq \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}, \; \hat{\nu}_k \doteq \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_z \end{pmatrix} . \tag{112}$$

5.2.1 Evaluation Criteria

Segmentation Error:

$\psi(\mathbf{x}, t)$ is the true (unknown) signed-distance function with respect to the target's silhouette within the image. As in the interactive segmentation formulation of Chapter 3, the error terms ξ and $\mathcal{D}$ are defined by

$$\xi \doteq (\mathcal{H}_\epsilon \circ \phi_k - \mathcal{H}_\epsilon \circ \psi_k) \; , \quad \mathcal{D}_k \doteq \frac{1}{2} \int_\Omega \xi^2 d\mathbf{x} \; . \tag{113}$$

Angular Displacement:

Net angular displacement Δ_k between images I_{k-1} and I_k is defined using the $\hat{\omega}_k$ component of the frame-to-frame transform (112):

$$\Delta_k \doteq \|(\omega_x, \omega_y, \omega_z)^T\|_2 \; . \tag{114}$$

Even when physical angular velocity is constant, variability in Δ_k can occur because the computation time in tracking affects observed angular displacement. This effect is captured in the simulation scheme used in the present results.

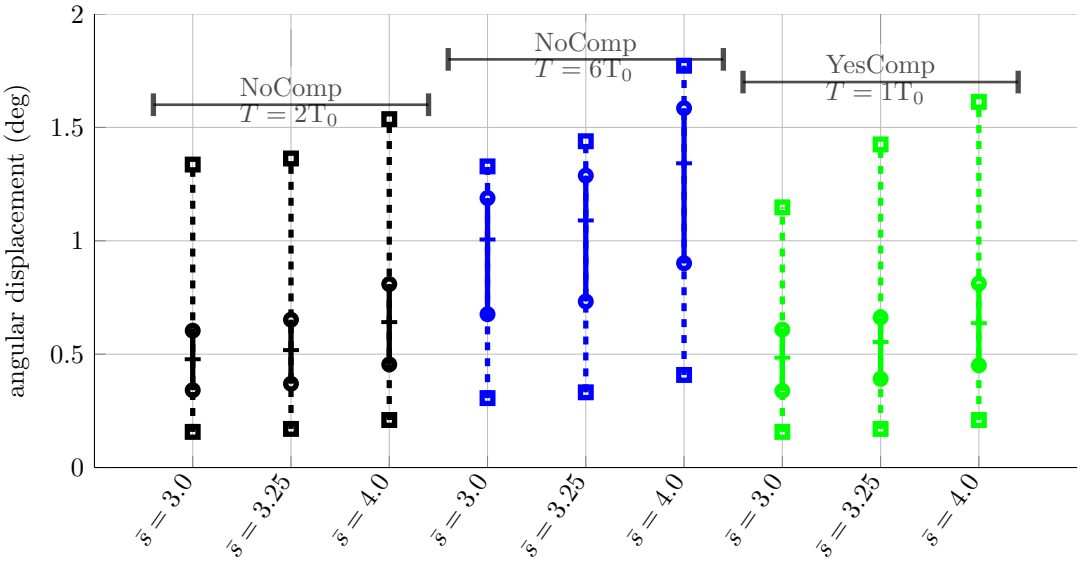
5.2.2 Open-Loop Results

Consider the following box plots Figure 40b and Figure 40a, which are generated in an open-loop tracking scenario with stationary target and camera affected only by external disturbances. The scale of disturbance-induced motion is denoted by $\bar{s}$; test sequences use values 3.0, 3.25, and 4.0 for $\bar{s}$. Three modes of tracking are used; corresponding to the columns of Table 3 and the colored groups of box plots shown in Figure 40.

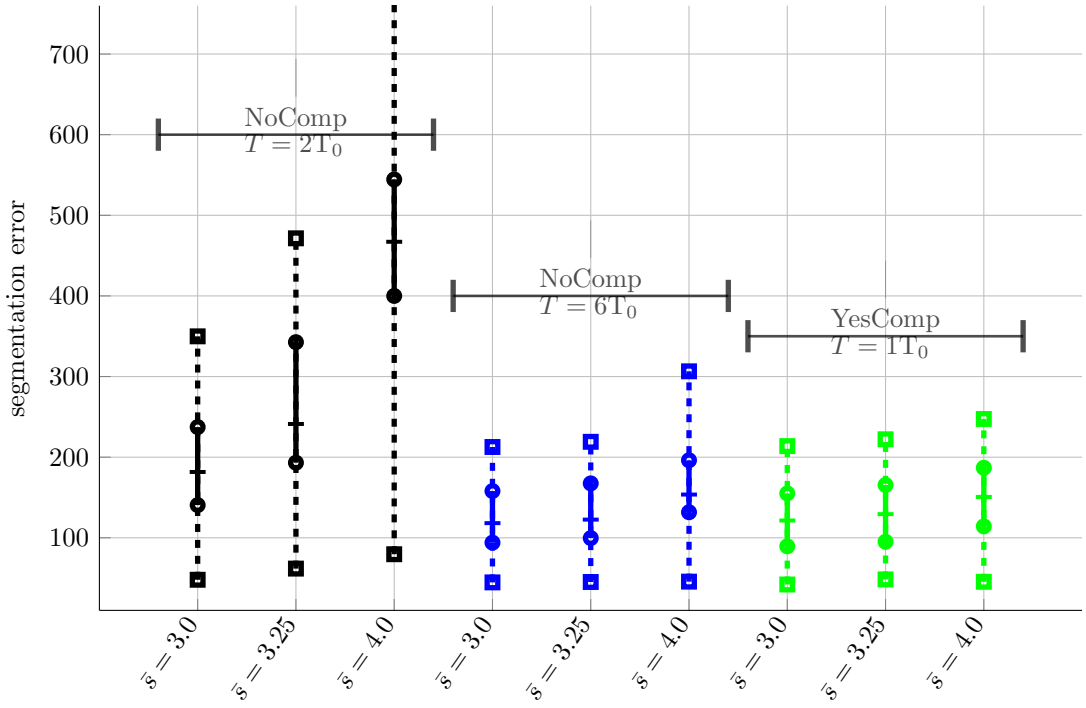
Without IMU compensation, two undesirable aspects are apparent in the tracker performance. Making the level set run faster to reduce dropped frames and inter-frame angular displacement causes low tracking accuracy. Conversely, running the level set for a longer physical time improves segmentation accuracy but raises the angular displacement due to dropped frames. The compensated initialization scheme achieves both low angular displacement and accurate segmentation. [60] considered levels of angular displacement at which a tracking algorithm starts to fail; however, from the closed-loop simulation discussed above, we see that the story is more complicated since the processing speed affects the apparent angular displacement. Even though an algorithm might fail at a certain level of angular displacement, a sufficiently fast execution speed can keep the Δ_k low and tracking will succeed.

Table 3: Three open-loop tracking modes are evaluated at three levels of motion disturbance in terms of angular displacement and segmentation error.

	scale of disturbance	$T = 2T_0$ NoComp	$T = 6T_0$ NoComp	$T = 1T_0$ YesComp
median Δ_k (angular displacement)				
	$\bar{s} = 3.00$	0.48	1.06	0.48
	$\bar{s} = 3.25$	0.51	1.09	0.55
	$\bar{s} = 4.00$	0.64	1.34	0.63
median $\mathcal{D}_k$ (segmentation error)				
	$\bar{s} = 3.00$	182	118	121
	$\bar{s} = 3.25$	241	123	129
	$\bar{s} = 4.00$	467	154	151



(a) Angular displacement Δ



(b) Segmentation error $\mathcal{D}$

Figure 40: Nine boxplots of angular displacement and segmentation error corresponding to three tracking modes run at three levels of ego-motion disturbance.

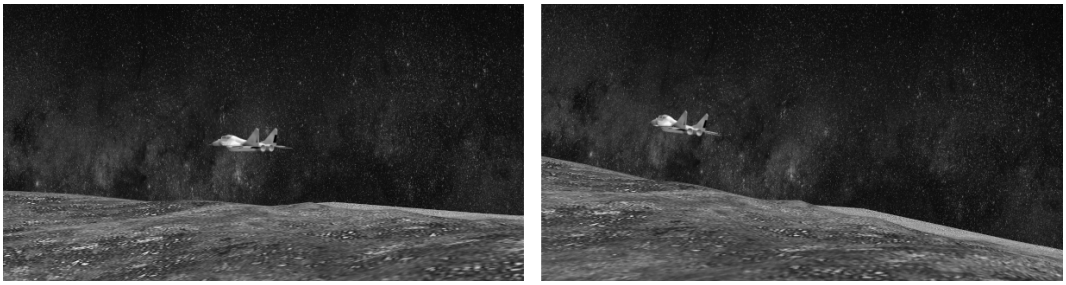
5.2.3 Limitations

Despite the benefits of using an IMU-compensated initialization ϕ_0, the technique has several limitations due to uncertainty in the measurements and environment. A compensated initialization will enable the tracking contour to stay within the region of attraction around the target in I_k even for very large displacements when the following ideal conditions are satisfied:

1. known target distance z_0, target size much smaller than z_0.
2. exact correspondence between frame-to-frame transform g_k and images $\{I_{k-1}, I_k\}$.

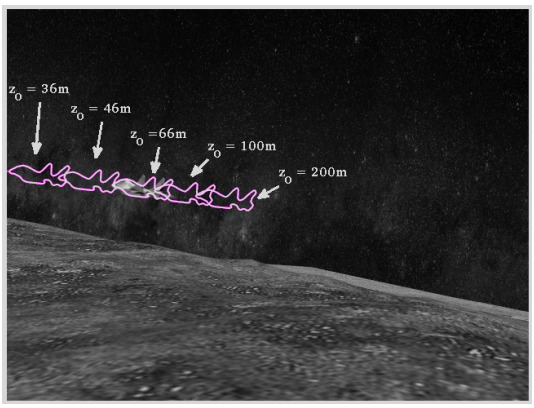
Depth Uncertainty:

First, the baseline distance to the target is not precisely known; therefore, the accuracy of ϕ_0 is sensitive to translational camera motion. Illustrated in Figure 41.



(a) raw images I_k, I_{k+1}.

(b) I_k, true $z_0 = 64$m.



(c) I_{k+1}, effect of z_0 uncertainty.

Figure 41: Lateral divert & roll, before and after. Compensation must now consider the unknown depth. It is assumed that tactical imagery is not amenable to methods based on matching point features.

Data Synchronization:

IMU and image acquisition/processing run not only at different rates but with some random time-varying delay. The IMU measurements will generally have negligible delay relative to the true current pose while the image I_k received concurrently with g_k will be delayed by a random time τ_k; accuracy of ϕ_0 is sensitive to temporal alignment. Illustrated in Figure 42.

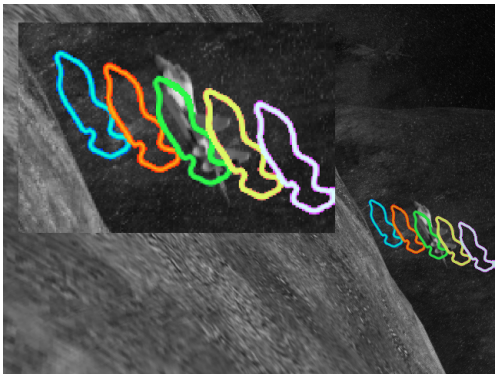


Figure 42: Compensation is sensitive to synchronization of images and IMU. Shown are several possible ϕ_0 contours for varying levels of delay, with a constant control being applied.

5.3 Closed-Loop Tracking with IMU Compensation

Applying the IMU compensation when performing closed-loop tracking leads to an undesirable loss of track for fast-moving targets (§ 5.3.1). Solving a blind-source-separation problem (§ 5.3.2) enables the closed-loop system to use a modified compensation scheme: ϕ_0 is warped only in response to disturbances, not the nominal motion control.

5.3.1 Problem: Induced Track Loss

Performing a state-dependent affine warp inside the image processing algorithm induces a feedback path that is a potential source of tracking failure. A control scheme that seeks to keep the target centered in the image uses the the current ϕ_k state to generate motion

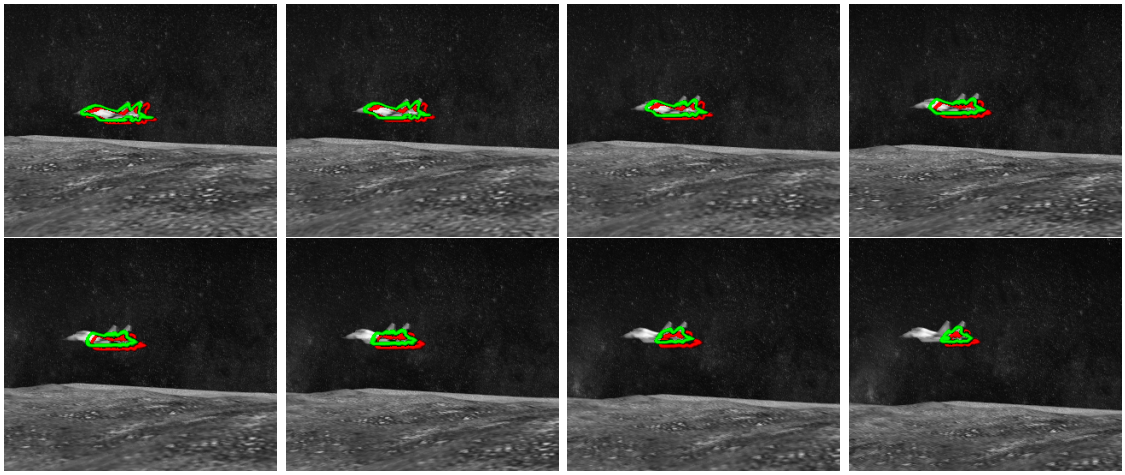


Figure 43: Image sequence illustrating induced track loss. Frame-to-frame transforms g_k are due to tracking motion of the camera panning left to follow the target. Red: warped ϕ_0 initial boundary, Green: final boundary. Each movement of the contour due to g_k moves the contour further from the target center, eventually causing loss of track. Compensation must use knowledge of the control signal to avoid the illustrated scenario.

signals; for example, by setting pitch and yaw inputs according to the target centroid:

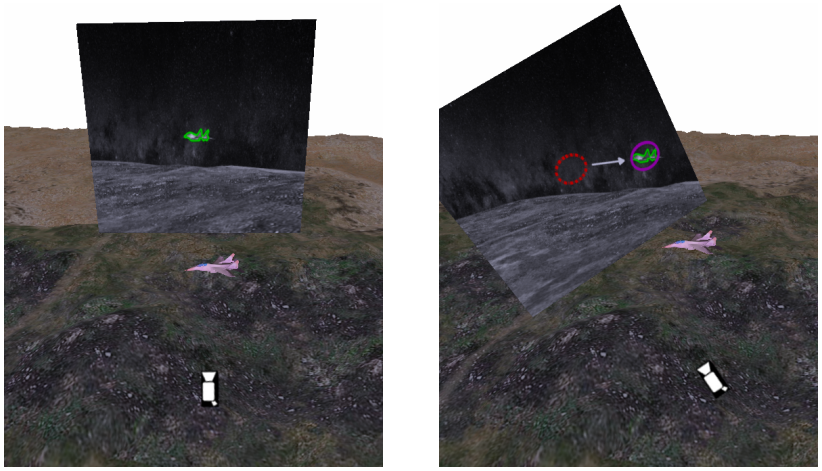
$$\bar{x}_c = \frac{\int_\Omega x_c \mathcal{H}_\epsilon \circ \phi \; dx_c dy_c}{\int_\Omega \mathcal{H}_\epsilon \circ \phi \; dx_c dy_c} \; , \bar{y}_c = \frac{\int_\Omega y_c \mathcal{H}_\epsilon \circ \phi \; dx_c dy_c}{\int_\Omega \mathcal{H}_\epsilon \circ \phi \; dx_c dy_c} \tag{115}$$

$$\omega_x(t) = -K\bar{y}_c/f \; , \quad \omega_y(t) = K\bar{x}_c/f \; . \tag{116}$$

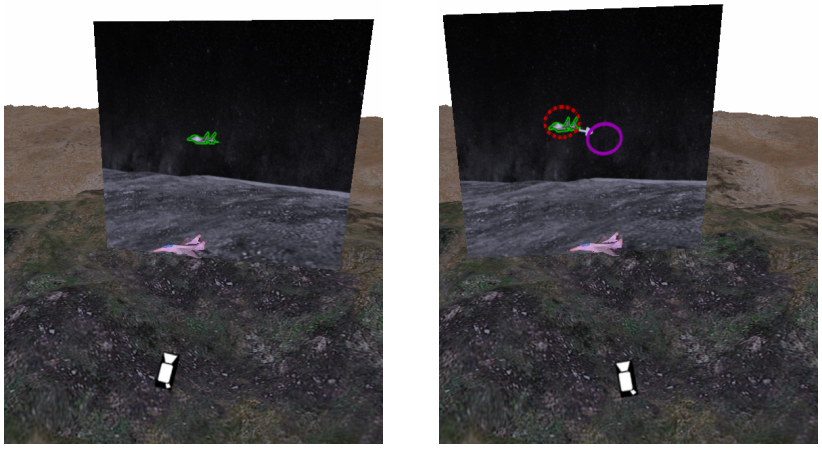
When the target is moving and the camera is driven by a control such as Eq. 115, warping ϕ_0 by the inverse frame-to-frame transform will cancel the motion control's effect and make the initialization of the image-dependent system worse. Figure 43 shows an image sequence exhibiting induced track loss. The camera pans left to follow the fast-moving target; frame-to-frame motion consists solely of the intended control. A possible solution is to increase the time of curve evolution during tracking. However, as in §5.2.2, the detrimental consequences of such an approach are increased computational delay and angular displacement between frames.

5.3.2 Separating Control from Disturbance

The transformation between frames is modeled by a composition of external disturbance g_k^d with estimated control input g_k^C. This input is taken to be piecewise constant and composed



(a) Compensating for disturbance.



(b) Compensating for intended motion.

Figure 44: World view of a an induced track loss scenario. Camera ego-motion caused by a disturbance can be compensated for rather directly (top). However, camera motion caused by tracking controls should not be compensated for; undoing the motion moves the contour away from the target and causes track loss (bottom). Note: camera images are not to scale with color top-view of environment.

of angular and linear parts scaled by τ_k, the time elapsed between frames:

$$g_k \doteq g_k^d g_k^C, \quad \tau_k \doteq t_k - t_{k-1}, \quad g_k^C = \exp\left\{\tau_k \begin{pmatrix} \omega_k^C & \nu_k^C \\ 0 & 0 \end{pmatrix}\right\} . \tag{117}$$

Compensating the tracking contour only for external disturbance g_k^d will preserve the desirable effects of motion control g_k^C. Separating these two components of g_k can be posed systematically as a constrained optimization problem. Constraints are due to the known structure of g_k^C, i.e. the known limits of motion control and their dependence on the contour state. The general formulation is as follows:

$$\text{Minimize:} \qquad P(\omega_k^d) + Q(\nu_k^d) \tag{118}$$

$$\text{Subject to:} \qquad g_k^d = g_k (g_k^C)^{-1} \tag{119}$$

$$\mathbf{W}(\phi, \omega_k^C, \tau_k) \leq 0 \tag{120}$$

$$\mathbf{V}(\phi, \nu_k^C, \tau_k) \leq 0 \ . \tag{121}$$

When control acts only on azimuth and elevation of the camera using target centroid measurements according to Eq. 115, all translation disturbances can be considered as target motion and the composition constraint Eq. 119 reduces to one involving only the difference $\omega_k - \omega_k^C$. The known goal of getting the target to appear centered in the image dictates that ω_x and ω_y components of ω_k^C should be along a direction similar to $(\bar{y}_c, \bar{x}_c)^T$. Thus, general optimization problem 118-120 is cast as the following specific form:

$$\text{Minimize:} \qquad \|\omega_k^d\|_2^2 \tag{122}$$

$$\text{Subject to:} \qquad \omega_k^d = \omega_k - \omega_k^C \tag{123}$$

$$|(1,0,0) \cdot \omega_k^C| \leq \tau_k \min\{\omega_{\mathrm{M}}, |K\bar{y}_c/f|\} \tag{124}$$

$$|(0,1,0) \cdot \omega_k^C| \leq \tau_k \min\{\omega_{\mathrm{M}}, |K\bar{x}_c/f|\} \tag{125}$$

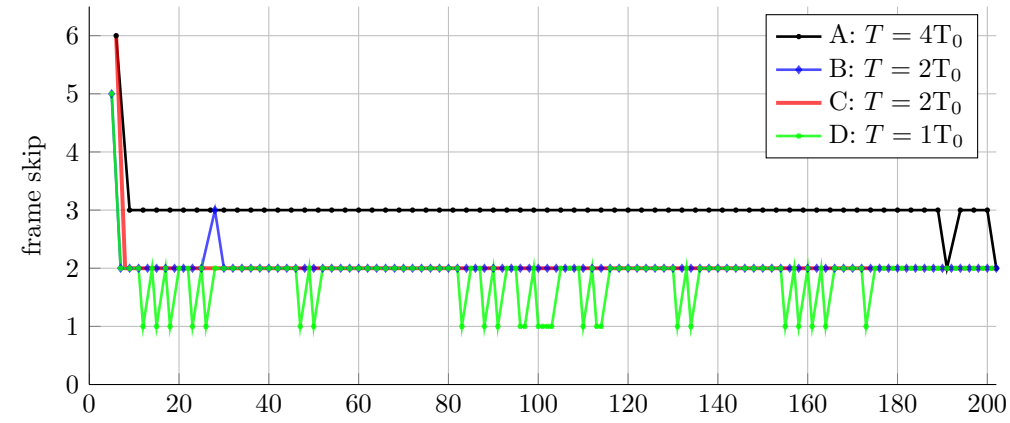
$$(0,0,1) \cdot \omega_k^C = 0 \tag{126}$$

$$\measuredangle \left\{ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \omega_k^C, \begin{pmatrix} \bar{y}_c \\ \bar{x}_c \end{pmatrix} \right\} - \theta_{\mathrm{M}} \leq 0 \ . \tag{127}$$

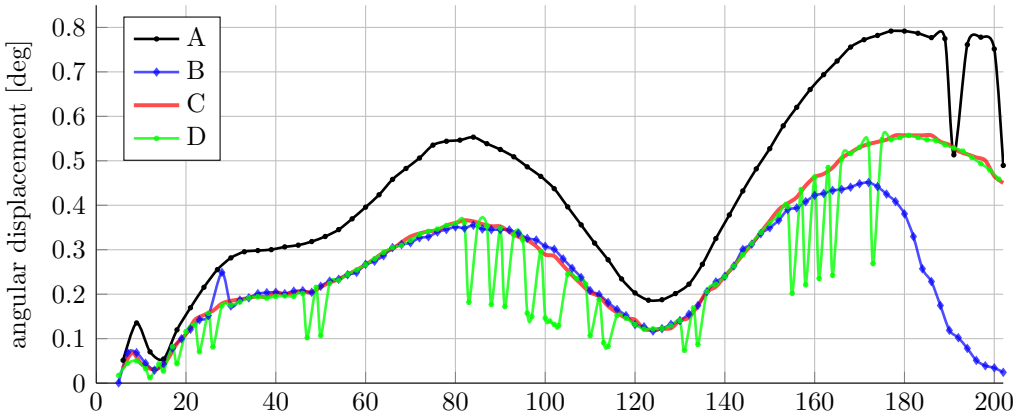
Next, the above optimization program is evaluated in § 5.3.3 with a closed-loop tracking scenario. Both ego-motion disturbances and a fast-moving target are present.

5.3.3 Simulation Results

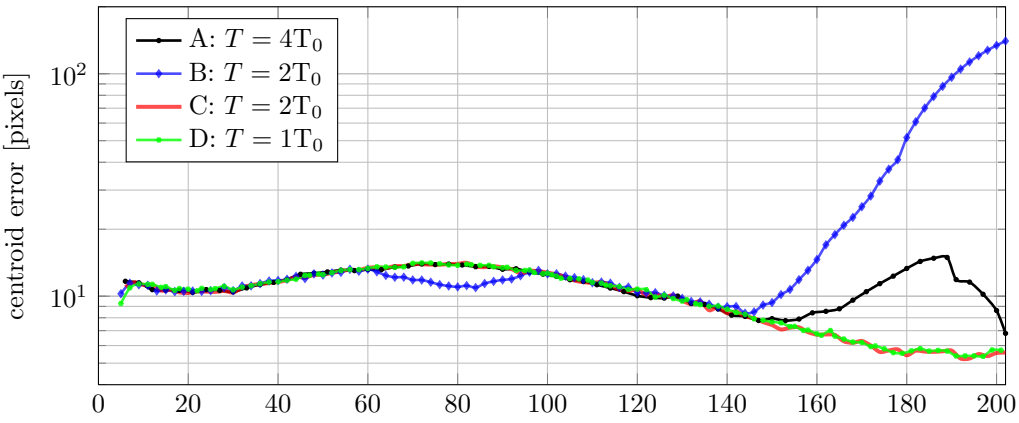
Computational delay, angular displacement, and centroid error obtained by four closed-loop configurations are shown in Figure 45; the target follows a constant trajectory in each of the four cases. In configurations A and B, compensation of the entire g_k is performed. Level set evolution in A uses a duration $T = 4\mathrm{T}_0$ and is able to not lose track, although the centroid error is quite large around frame 180 and penalties of large computational delay and angular displacement are incurred. Configuration B uses duration $T = 2\mathrm{T}_0$; fewer frames are dropped and angular displacement drops, but ultimately this tracker loses the target after frame 140. Configurations C and D compute the estimated control g_k^C and disturbance g_k^d; compensation uses only the g_k^d term. Configuration C uses the same level set duration as B and not only achieves comparable dropped frames and angular displacement but maintains track throughout the sequence. Lastly, configuration D demonstrates that, without losing track, lag and angular displacement can be reduced even further when the level set duration is set to $T = 1\mathrm{T}_0$.



(a) frames skipped due to computational delay



(b) angular displacement between processed frames



(c) centroid error, tracked vs. truth

Figure 45: Closed-loop figures of merit. Performance improves when using the proposed IMU-compensated strategy. Both ego-motion disturbances and feedback motion control are present.

5.4 Regulation of Area and Shape: Metaimage U (Type II)

Tracking with contours as opposed to fixed-shape templates more accurately delineates the pixels constituting the target and background [105, 150]. Contour-based tracking generates an accurate localization of the boundary, thus enabling further processing such as pose estimation [18, 36, 128] and target identification [54, 78].

The deformable nature of contours can lead to erroneous segmentation of the target. As in the ISIS problem, segmentation errors occur when the underlying algorithm assumptions do not hold. Unlike ISIS, however, there is no "user" to make corrections in a tracker. A feedback control affecting ϕ_t will come instead from prior knowledge of what visible area and shape the target will generate in a video stream. A number of papers have proposed augmented functionals that consider shape priors in addition to photometric properties [31, 41, 121]. The novel contribution of the present work is to cast the use of prior shape knowledge as a control synthesis problem within the same framework as interactive static image segmentation (ISIS).

Prior knowledge or measurements of the environment in conjunction with the image formation model are used to generate an exogenous meta-image $\mathbf{U}(\mathbf{x}, t) : \Omega \to \mathbb{R}^m$. Using $\mathbf{U}$ as part of the control affecting ϕ_t enables the tracking algorithm to use not only raw image data but also additional information about the target and environment. The closed-loop system with control $f(U)$ becomes

$$\phi_t = \delta_\epsilon \circ \phi \left[G(\phi, I) + \lambda\kappa(\phi) + f(U) \right] \ . \tag{128}$$

From a design perspective, this formulation is convenient because it separates the encoding of U from the synthesis of control $f(\cdot)$. Eq. 128 is a very general case; the rest of this section considers a specific composition of U that regulates the estimated target boundary and consequently improves tracking through clutter.

To regulate the observed area and shape of the target, let U be the tuple

$$U(\mathbf{x}) \doteq \begin{pmatrix} A(\phi) \\ \| \mathbf{x} - [\bar{x}_c, \bar{y}_c]^T \|_p \end{pmatrix}, \tag{129}$$

and define the functionals

$$A(t) = \int_\Omega \mathcal{H}_\epsilon \circ \phi \; d\mathbf{x} \; , \tag{130}$$

$$S(t) = \int_\Omega \mathcal{H}_\epsilon \circ \phi \bullet \left[\| \mathbf{x} - [\bar{x}_c, \bar{y}_c]^T \|_p > r_{\max} \right] d\mathbf{x} \; , \tag{131}$$

$$\tag{132}$$

where the term in brackets takes values 0 or 1. To regulate A and S, the following functions of U are used:

$$f_1(U) = \begin{cases} +G_{\text{M}} - \lambda\kappa & \text{if} \quad A(t) < \quad A_{\min} \\ -G_{\text{M}} - \lambda\kappa & \text{if} \quad A(t) > \quad A_{\max} \\ 0 & \qquad \text{otherwise,} \end{cases} \tag{133}$$

$$f_2(U) = \begin{cases} -G_{\text{M}} - \lambda\kappa & \text{if} \quad \| \mathbf{x} - [\bar{x}_c, \bar{y}_c]^T \|_p > \quad r_{\max} \\ 0 & \qquad \text{otherwise.} \end{cases} \tag{134}$$

Proposition 5.4.1 *Applying only $f_1(U)$ leads to $A' \leq 0$ when $A \geq A_{max}$ and $A' \geq 0$ for $A \leq A_{min}$. Similarly, $f_2(U)$ acting alone on ϕ_t causes $S' \leq 0$. Derivation:*

$$A' = \int_\Omega (\delta_\epsilon \circ \phi) \bullet \phi_t \; d\mathbf{x} \tag{135}$$

$$= \int_\Omega (\delta_\epsilon^2 \circ \phi) \bullet \left[G + \lambda\kappa(\phi) + f_1 \right] \; d\mathbf{x} \; . \tag{136}$$

From the definition (133), $A > A_{max}$ implies $f_1 = -\lambda\kappa - G_{\text{M}}$ and thus $A' \leq 0$. Similarly, $A < A_{min}$ implies $f_1 = -\lambda\kappa + G_{\text{M}}$ and $A' \geq 0$. $S' \leq 0$ by a similar argument using f_2 from (134):

$$S' = \int_\Omega (\delta_\epsilon \circ \phi) \bullet \phi_t \bullet \left(\| \mathbf{x} - [\bar{x}_c, \bar{y}_c]^T \|_p > r_{max} \right) \tag{137}$$

$$= \int_\Omega (\delta_\epsilon^2 \circ \phi) \bullet \left(\| \mathbf{x} - [\bar{x}_c, \bar{y}_c]^T \|_p > r_{max} \right) \bullet \left[G + \lambda\kappa(\phi) + f_2 \right] \; d\mathbf{x} \; , \tag{138}$$

with the integrand being zero or negative when the first bracketed term is 0 or 1, respectively. Thus, $S' \leq 0$. ∎

In each of the three inequalities in the proposition, equality occurs if and only if ϕ is initialized in a degenerate manner such that $\int_\Omega \delta_\epsilon^2 \circ \phi \; d\mathbf{x} = 0$. Combining f_1 and f_2 in the

closed-loop segmentation system gives

$$\phi_t = \delta_\epsilon \circ \phi \left[G(\phi, I) + \lambda\kappa(\phi) + \underbrace{f_1(U)}_{\text{area regulation}} + \underbrace{f_2(U)}_{\text{shape regulation}} \right] . \tag{139}$$

The net effect of the feedback control above is to (a) keep target area within an expected range and (b) quash any portions of the contour that are far from the target centroid. The $-\lambda\kappa$ portion of f_1 and f_2 is necessary to produce the desired behavior when G_{M} is nearly zero due to photometric properties of the current image.

5.4.1 Relationship with the Exisiting Literature

Many powerful algorithms have been proposed that use a combination of photometric and shape-based energies [31, 41, 113, 121, 126]. The ideas therein are compatible with the feedback control perspective used here. For example, [126] defines "confidence field" σ_m that is minimizes an auxiliary functional $E(\sigma_m|\phi_m)$ where ϕ_m is a superposition of training shapes that best represent the current segmentation ϕ. Gradient descent for ϕ acts on an augmented functional that contains both a shape-driven term due to σ_m and an image-dependent term. Thus, the σ_m can alternatively be viewed as an instance of meta-image U in the present formulation.

There are several advantages of the feedback control formulation. First, it is generally sufficient to stabilize functionals of interest; one can simply use the functional values themselves in feedback and reduce the need to set weighting parameters in a sensitive static optimization problem. Second, additional terms can be applied in $f(U)$ without needing the details of how those terms are computed. Control design tools such as observers and backstepping can subsequently be used to generate the required input.

The ease of use provided by the proposed approach is clearly seen in the simplicity of area regulation with Eq. 133. To define a gradient flow for ϕ using an augmented functional that when minimized has the same switching property for constraining area, we would need a term of the form

$$E_{\text{area}}(\phi) = \alpha \left(\int_\Omega \mathcal{H}_\epsilon \circ \phi \; d\mathbf{x} \; , \; E_{\text{image}}(\phi) \right) \cdot \int_\Omega \mathcal{H}_\epsilon \circ \phi \; d\mathbf{x} \; . \tag{140}$$

The weight α depends on both area and the range of values taken by the image-dependent $E_{\text{image}}(\phi)$. Further complicating matters, α would need to be differentiable with respect to ϕ for gradient descent to make sense. The proposed control-oriented method regulates individual Lyapunov functionals, avoiding the need to carefully define terms like Eq. 140 and minimize a monolithic weighted energy. Each Lyapunov functional is more manageable analytically and takes on values that are easy to interpret.

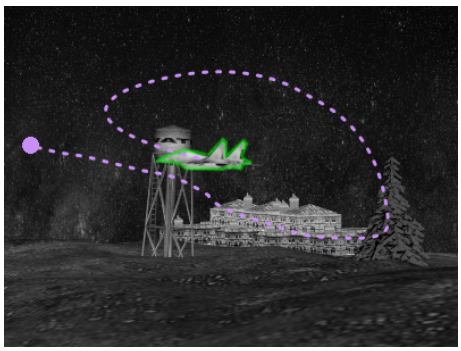
5.4.2 Simulation Results: Tracking Through Clutter

A closed-loop tracking scenario is generated with the target path going through strong ground clutter as shown in Figure 46a. The open-loop system uses a localized variant of the mean separation cost. While this nominal system performs well for the light target on fairly darker backgrounds, it tends to err in the presence of bright clutter such as the water tower and building in Figure 46b. In contrast, Figure 46c illustrates successful tracking of the target when the target boundary is regulated by $f(U)$ as described in the previous section.

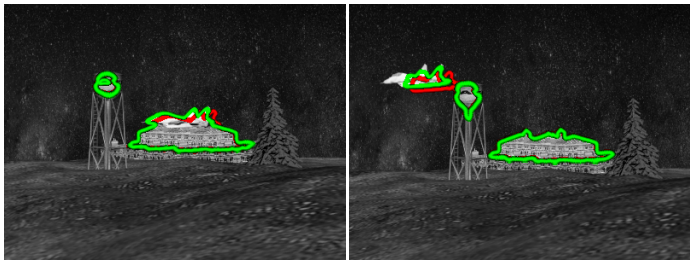
Selecting the parameters needed to generate constituent f_1 and f_2 is straightforward when reasonable estimates of target's range and shape are available. The bounding box for the MiG-35 target is shown in Figure 47; the visible cross-sectional area from a side view is $A_0 = 6.0\text{m}^2$. The intrinsic parameter matrix K is known and the target range satisfies $z_0 \in [70, 90]\text{m}$. Values for $A_{\min}$ and $A_{\max}$ are generated by

$$A_{\min} = A_0 \times (f/z_{\max})^2, \quad A_{\max} = A_0 \times (f/z_{\min})^2.$$

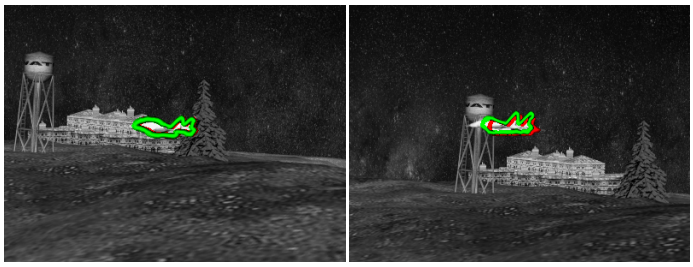
Several quantities of interest are plotted in Figure 48. First, the loss of track by the nominal system is apparent from the centroid error; it spikes over 100 pixels when the contour is stuck in the clutter. The regulated version keeps the centroid error below 10 pixels throughout the scenario. Second, the regulated tracker maintains target area within the desired bounds. Lastly, the delays in this system are seen to be state-dependent; larger (unregulated) contour area increases the computational burden of level set evolution.



(a) target path through clutter.

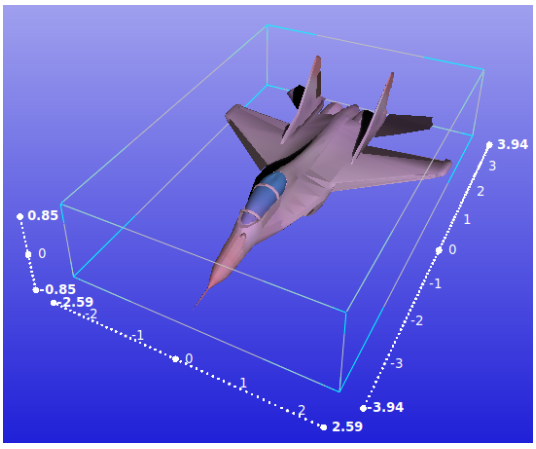


(b) failure with open-loop system.



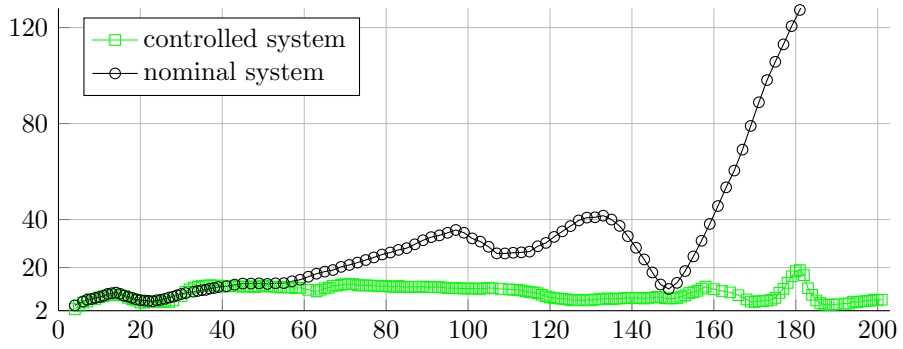
(c) success with closed-loop system.

Figure 46: Sample results, tracking target through clutter.

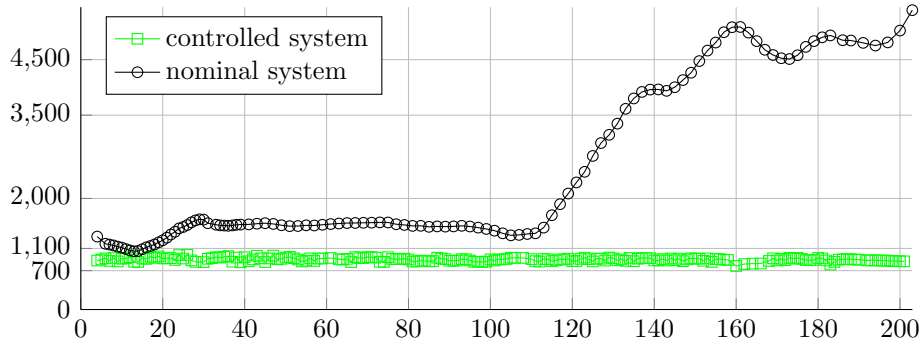



```
K=[1013.32, 0, 319.5;
      0, 1013.32, 239.5;
      0,  0,  1]
Amax = 6.0 * (f/70)^2 = 1259.0
Amin = 6.0 * (f/90)^2 =  761.0
```

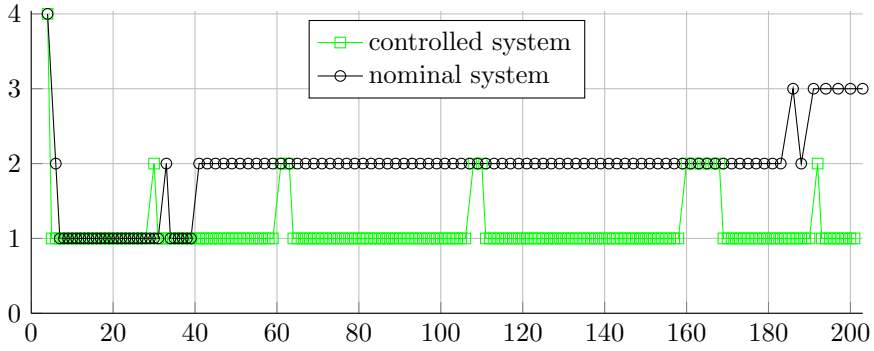
Figure 47: Left: bounding box for MiG-35 model. Right: intrinsic parameter matrix and computation of contour area limits.



(a) centroid error, tracked vs truth.



(b) area of tracking contour.



(c) frames skipped due to computational delay.

Figure 48: Regulated contour area and shape: simulation results of closed-loop tracking through clutter.

5.5 Discussion

Traditionally, data obtained from physical sensors (IMU) and computer vision are treated as two independent measurement sources. A new model of the vision-based closed-loop tracking problem has been formulated: the algorithm responsible for segmentation of the target is viewed as a separate controllable sub-system rather than merely a measurement source.

The proposed formulation enables improved tracking performance in several ways. First, IMU-compensated initialization of the segmentation algorithm can successfully track a target under strong camera jitter that would otherwise cause tracking failure. Computational delays in image processing are reduced due to the compensated initialization, which improves net system performance. Second, combining IMU readings and prior target knowledge enables the creation of a meta-image to regulate the area and shape of the contour. Tracking a target through both ground clutter and above the horizon can then be handled by augmenting the iterative segmentation algorithm; a control input to the image algorithm subsystem encodes the effect of the prior shape knowledge on target appearance relative to its background. The augmented algorithm enables tracking that is more reliable than would be possible without considering the IMU data; the target is not easily lost when it transitions between ground clutter and sky backgrounds. Applications such as boost-phase intercept in ballistic missile defense [21] and UAV formation flight [88] can benefit substantially from developing guidance control jointly with image processing algorithms as in this work.

REFERENCES

[1] Seg3D: Volumetric Image Segmentation and Visualization. Scientific Computing and Imaging Institute (SCI).

[2] A. Top, G. H. and Abugharbieh, R., "Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3d image segmentation," in *MICCAI*, 2010.

[3] Acosta, G. and Durán, R., "An optimal poincare inequality in l^ 1 for convex domains," *PROCEEDINGS-AMERICAN MATHEMATICAL SOCIETY*, vol. 132, no. 1, pp. 195–202, 2004.

[4] Astolfi, A., Hsu, L., Netto, M., and Ortega, R., "Two solutions to the adaptive visual servoing problem," *IEEE transactions on robotics and automation*, vol. 18, no. 3, pp. 387–392, 2002.

[5] Avrachenkov, K. and Haviv, M., "Perturbation of null spaces with application to the eigenvalue problem and generalized inverses," *Linear Algebra and its Applications*, vol. 369, pp. 1–25, 2003.

[6] Aw, S., Halmagyi, G., Haslwanter, T., Curthoys, I., Yavor, R., and Todd, M., "Three-dimensional vector analysis of the human vestibuloocular reflex in response to high-acceleration head rotations. ii. responses in subjects with unilateral vestibular loss and selective semicircular canal occlusion," *Journal of neurophysiology*, vol. 76, no. 6, p. 4021, 1996.

[7] Baker, S. and Matthews, I., "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[8] Balafar, M., Ramli, A., Saripan, M., and Mashohor, S., "Review of brain MRI image segmentation methods," *Artificial Intelligence Review*, vol. 33, no. 3, pp. 261–274, 2010.

[9] Barles, G., Soner, H., and Souganidis, P., "Front propagation and phase field theory," *SIAM Journal on Control and Optimization*, vol. 31, p. 439, 1993.

[10] Bay, H., Tuytelaars, T., and Van Gool, L., "SURF: Speeded up robust features," *Proceedings of European Conference Computer Vision*, pp. 404–417, 2006.

[11] Ben Ayed, I. and Mitiche, A., "A region merging prior for variational level set image segmentation," *Image Processing, IEEE Transactions on*, vol. 17, no. 12, pp. 2301–2311, 2008.

[12] Ben-Zadok, N., Riklin-Raviv, T., and Kiryati, N., "Interactive level set segmentation for image-guided therapy," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pp. 1079–1082, IEEE, 2009.

[13] Blandin, S., Litrico, X., and Bayen, A., "Boundary stabilization of the inviscid burgers equation using a lyapunov method," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1705–1712, IEEE, 2010.

[14] Bleser, G. and Stricker, D., "Advanced tracking through efficient image processing and visual-inertial sensor fusion," *Computers & Graphics*, vol. 33, no. 1, pp. 59–72, 2009.

[15] Boothe, R., *Perception of the visual environment*. Springer Verlag, 2002.

[16] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1124–1137, 2004.

[17] Boykov, Y., Veksler, O., and Zabih, R., "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[18] Brox, T., Rosenhahn, B., and Weickert, J., "Three-dimensional shape knowledge for joint image segmentation and pose estimation," *Pattern Recognition*, pp. 109–116, 2005.

[19] Brox, T. and Weickert, J., "Level set segmentation with multiple regions," *Image Processing, IEEE Transactions on*, vol. 15, no. 10, pp. 3213–3218, 2006.

[20] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., "BRIEF: Binary robust independent elementary features," *Proceedings of European Conference on Computer Vision*, pp. 778–792, 2010.

[21] Canavan, G., *Missile Defense for the 21st Century*. Heritage Foundation, 2003.

[22] Casas, J. and Simpson, S., *Insect Mechanics and Control*. Academic Press, 2007.

[23] Chan, T. and Vese, L., "Active contours without edges," *IEEE Transactions on image processing*, vol. 10, no. 2, pp. 266–277, 2001.

[24] Chaumette, F. and Hutchinson, S., "Visual servo control II: Advanced approaches," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[25] Chen, J., Dixon, W., Dawson, M., and McIntyre, M., "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 406–415, 2006.

[26] Choi, S., Kim, T., and Yu, W., "Performance evaluation of RANSAC family," in *Proceedings of British Machine Vision Conference*, pp. 1–12, 2009.

[27] Collins, R., "Mean-shift blob tracking through scale space," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 234–240, 2003.

[28] Conn, A., Scheinberg, K., and Vicente, L., *Introduction to derivative-free optimization*. Society for Industrial Mathematics, 2009.

[29] Corke, P., *High-performance visual closed-loop robot control*. PhD thesis, University of Melbourne, 1994.

[30] Corke, P. and Good, M., "Dynamic effects in visual closed-loop systems," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 671–683, 1996.

[31] Cremers, D., "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1262–1273, 2006.

[32] Cremers, D., Rousson, D., and Deriche, R., "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *International Jounral of Computer Vision*, vol. 72, no. 2, pp. 195–215, 2007.

[33] Cunha, R., Silvestre, C., Hespanha, J., and Aguiar, A. P., "Vision-based control for rigid body stabilization," *Automatica*, vol. 47, no. 5, pp. 1020 – 1027, 2011.

[34] C.Westbrook, ed., *MRI at a Glance*. Malden, MA: Wiley Blackwell, 2010.

[35] Dambreville, S., Rathi, Y., and Tannenbaum, A., "A framework for image segmentation using shape models and kernel space shape priors," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1385–1399, 2007.

[36] Dambreville, S., Sandhu, R., Yezzi, A., and Tannenbaum, A., "Robust 3D Pose Estimation and Efficient 2D Region-Based Segmentation from a 3D Shape Prior," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, pp. 169–182, Springer-Verlag Berlin, Heidelberg, 2008.

[37] Dani, A. and Dixon, W., "Single Camera Structure and Motion Estimation," *Visual Servoing via Advanced Numerical Methods*, pp. 209–229, 2010.

[38] Dani, A., Gans, N., and Dixon, W., "Position-based visual servo control of leader-follower formation using image-based relative pose and relative velocity estimation," in *American Control Conference*, pp. 5271–5276, 2009.

[39] De Boer, R., Vrooman, H., Ikram, M., Vernooij, M., Breteler, M., Van der Lugt, A., and Niessen, W., "Accuracy and reproducibility study of automatic MRI brain tissue segmentation methods," *NeuroImage*, vol. 51, no. 3, pp. 1047–1056, 2010.

[40] Deutsch, B., Graessl, C., F., B., and Denzler, J., *A Comparative Evaluation of Template and Histogram-Based 2D Tracking Algorithms*, pp. 269–276. Springer, 2005.

[41] El-Baz, A. and Gimelfarb, G., "Robust Medical Images Segmentation Using Learned Shape and Appearance Models," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pp. 281–288, 2009.

[42] El-Farra, N., Armaou, A., and Christofides, P., "Analysis and control of parabolic pde systems with input constraints," *Automatica*, vol. 39, no. 4, pp. 715–725, 2003.

[43] E.Nickoloff and P.Alderson, "Radiation exposure to patients from ct," *American Journal of Roentgenology*, vol. 177, no. 2, pp. 285–287, 2001.

[44] Ettinger, S., Nechyba, M., Ifju, P., and Waszak, M., "Vision-guided flight stability and control for micro air vehicles," vol. 3, pp. 2134–2140, 2002.

[45] Filar, J., Hudson, I., Mathew, T., and Sinha, B., "Analytic perturbations and systematic bias in statistical modeling and inference," *Arxiv preprint arXiv:0805.2228*, 2008.

[46] Fogal, T. and Krüger, J., "Tuvok, an architecture for large scale volume rendering," in *Proceedings of the 15th International Workshop on Vision, Modeling, and Visualization*, pp. 57–66, 2010.

[47] Gao, Y., Tannenbaum, A., and Kikinis, R., "Simultaneous multi-object segmentation using local robust statistics and contour interaction," *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pp. 195–203, 2011.

[48] Garcia-Pardo, P., Sukhatme, G., and Montgomery, J., "Towards vision-based safe landing for an autonomous helicopter," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 19–29, 2002.

[49] Gomes, J. and Faugeras, O., "Reconciling distance functions and level sets," in *Biomedical Imaging, 2002. 5th IEEE EMBS International Summer School on*, pp. 15–pp, IEEE, 2002.

[50] Hammond, K., Xerogeanes, J., and Todd, D., "Anatomic landmarks utilized for physeal-sparing, anatomic anterior cruciate ligament reconstruction: An mri-based study (ss-68)," *Arthroscopy: The Journal of Arthroscopic & Related Surgery*, vol. 27, no. 5, pp. e66–e67, 2011.

[51] Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[52] Heckel, F., Konrad, O., Karl Hahn, H., and Peitgen, H., "Interactive 3d medical image segmentation with energy-minimizing implicit functions," *Computers & Graphics*, 2011.

[53] Hecker, F., K., O., Horst, K., and Peitgen, H., "Interactive 3d medical image segmentation with energy-minimizing implicit functions," *Journal of Computers and Graphics*, vol. 35, no. 2, pp. 275–287, 2011.

[54] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., and Lepetit, V., "Gradient response maps for real-time detection of textureless objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 876–888, 2012.

[55] Hol, J., Schön, T., and Gustafsson, F., "Modeling and calibration of inertial and vision sensors," *The International Journal of Robotics Research*, vol. 29, no. 2-3, p. 231, 2010.

[56] Hsieh, J., *Computed tomography: principles, design, artifacts, and recent advances*, vol. 114. Society of Photo Optical, 2003.

[57] Hu, G., Gans, N., and Dixon, W., "Quaternion-based visual servo control in the presence of camera calibration error," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 5, pp. 489–503, 2010.

[58] Hu, G., Gans, N., Fitz-Coy, N., and Dixon, W., "Adaptive homography-based visual servo tracking control via a quaternion formulation," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 1, pp. 128–135, 2009.

[59] Hulusić, V., Debattista, K., Aggarwal, V., and Chalmers, A., "Maintaining frame rate perception in interactive environments by exploiting audio-visual cross-modal interaction," *The Visual Computer*, pp. 1–10, 2011.

[60] Hwangbo, M., Kim, J.-S., and Kanade, T., "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and gpu implementation," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1755–1774, 2011.

[61] Jeong, W., Beyer, J., Hadwiger, M., Vazquez, A., Pfister, H., and Whitaker, R., "Scalable and interactive segmentation and visualization of neural processes in EM datasets," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1505–1514, 2009.

[62] Joshi, N., Kang, S., Zitnick, C., and Szeliski, R., "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 30, 2010.

[63] Kapadia, A., Braganza, D., Dawson, D., and Mclntyre, M., "Adaptive camera calibration with measurable position of fixed features," in *American Control Conference, 2008*, pp. 3869–3874, 2008.

[64] Karasev, P., Kolesov, I., Chudy, K., Tannenbaum, A., Muller, G., and Xerogeanes, J., "Interactive MRI Segmentation with Controlled Active Vision," in *IEEE Conference on Decision and Control*, 2011.

[65] Karasev, P., Serrano, M., Vela, P., and Tannenbaum, A., "Visual closed-loop tracking with area stabilization," in *American Control Conference (ACC), 2010*, pp. 6955–6961, IEEE, 2010.

[66] Karasev, P., Serrano, M., Vela, P., and Tannenbaum, A., "Depth Invariant Visual Servoing," in *to appear IEEE Conference on Decision and Control*, 2011.

[67] Kelly, J., Saripalli, S., and Sukhatme, G., "Combined visual and inertial navigation for an unmanned aerial vehicle," in *Field and Service Robotics*, pp. 255–264, Springer, 2008.

[68] Kercher, J., Xerogeanes, J., Tannenbaum, A., Al-Hakim, R., Black, J., and Zhao, J., "Anterior cruciate ligament reconstruction in the skeletally immature: an anatomical study utilizing 3-dimensional magnetic resonance imaging reconstructions," *Journal of Pediatric Orthopaedics*, vol. 29, no. 2, p. 124, 2009.

[69] Khalil, H., *Nonlinear systems, Third Edition*. Prentice Hall, 2002.

[70] Kim, J., Fisher III, J., Yezzi, A., Cetin, M., and Willsky, A., "A nonparametric statistical method for image segmentation using information theory and curve evolution," *Image Processing, IEEE Transactions on*, vol. 14, no. 10, pp. 1486–1502, 2005.

[71] Kim, S., Bazin, J., Lee, H., Choi, K., and Park, S., "Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data," *Radar, Sonar & Navigation, IET*, vol. 5, no. 8, pp. 814–823, 2011.

[72] Kneip, L., Weiss, S., and Siegwart, R., "Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 2235–2241, IEEE, 2011.

[73] Kokotović, P. and Arcak, M., "Constructive nonlinear control: a historical perspective," *Automatica*, vol. 37, no. 5, pp. 637–662, 2001.

[74] Kouachi, S. and Al-Eid, E., "Explicit invariant regions and global existence of solutions for reaction diffusion systems with a full matrix of diffusion coefficients and nonhomogeneous boundary conditions.," *Int. J. Open Problems Comput. Math*, vol. 5, no. 1, 2012.

[75] Krstic, M. and Smyshlyaev, A., "Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays," *Systems & Control Letters*, vol. 57, no. 9, pp. 750–758, 2008.

[76] Krstic, M. and Smyshlyaev, A., "adaptive control of pdes"," *Annual Reviews in Control*, vol. 32, pp. 149–160, 2008.

[77] Kumar, D. and Jawahar, C., "Robust homography-based control for camera positioning in piecewise planar environments," *Computer Vision, Graphics and Image Processing*, pp. 906–918, 2006.

[78] Lai, K., Bo, L., Ren, X., and Fox, D., "Detection-based object labeling in 3d scenes," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1330–1337, IEEE, 2012.

[79] Lankton, S. and Tannenbaum, A., "Localizing region-based active contours," *Image Processing, IEEE Transactions on*, vol. 17, no. 11, pp. 2029–2039, 2008.

[80] LeVeque, R. J., *Numerical methods for conservation laws*. Birkhäuser, 1992.

[81] Li, C., Xu, C., Gui, C., and Fox, M., "Level set evolution without re-initialization: A new variational formulation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 430–436, Ieee, 2005.

[82] Li, C., Xu, C., Gui, C., and Fox, M., "Distance regularized level set evolution and its application to image segmentation," *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3243–3254, 2010.

[83] Li, M. and Mourikis, A., "Improving the accuracy of ekf-based visual-inertial odometry," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 828–835, IEEE, 2012.

[84] Ma, L., Cao, C., Hovakimyan, N., Woolsey, C., and Dixon, W., "Fast estimation for range identification in the presence of unknown motion parameters," *IMA Journal of Applied Mathematics*.

[85] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S., "An Invitation to 3D Vision: From Images to Models Springer Verlag," 2003.

[86] Magnis, L., Vazquez, R., and others, "Nonlinear control of the viscous burgers equation: Trajectory generation, tracking, and observer design," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, p. 021012, 2009.

[87] Malis, E., Hamel, T., Mahony, R., and Morin, P., "Estimation of Homography Dynamics on the Special Linear Group," *Visual Servoing via Advanced Numerical Methods*, pp. 133–150, 2010.

[88] Mariottini, G., Morbidi, F., Prattichizzo, D., Vander Valk, N., Michael, N., Pappas, G., and Daniilidis, K., "Vision-Based Localization for Leader–Follower Formation Control," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1431–1438, 2009.

[89] Martinelli, A., "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *Robotics, IEEE Transactions on*, vol. 28, pp. 44 –60, feb. 2012.

[90] Matthews, L., Ishikawa, T., and Baker, S., "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[91] McGuinness, K. and O'Connor, N., "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[92] McGuinness, K. and O'Connor, N., "Toward automated evaluation of interactive segmentation," *Computer Vision and Image Understanding*, 2011.

[93] McInerney, T. and Terzopoulos, D., "Deformable models in medical image analysis: a survey," *Medical image analysis*, vol. 1, no. 2, pp. 91–108, 1996.

[94] Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M., "Pixhawk: A system for autonomous flight using onboard computer vision," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 2992–2997, IEEE, 2011.

[95] Merino, L., Wiklund, J., Caballero, F., Moe, A., De Dios, J., Forssen, P., Nordberg, K., and Ollero, A., "Vision-based multi-uav position estimation," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 53–62, 2006.

[96] Meyer-Baese, U., *Digital signal processing with field programmable gate arrays*. Springer Verlag, 2007.

[97] Michailovich, O., Rathi, Y., and Tannenbaum, A., "Image segmentation using active contours driven by the bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2787–2801, 2007.

[98] Mitiche, A. and Ayed, I., "Multiregion segmentation," *Variational and Level Set Methods in Image Segmentation*, pp. 59–81, 2011.

[99] Montijano, E. and Sagues, C., "Fast pose estimation for visual navigation using homographies," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2704–2709, IEEE, 2009.

[100] Mory, B., Ardon, R., Yezzi, A., and Thiran, J., "Non-euclidean image-adaptive radial basis functions for 3d interactive segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 787–794, IEEE, 2009.

[101] Moschidis, E. and Graham, J., "Simulation of User Interaction for Performance Evaluation of Interactive Image Segmentation Methods," *Proceedings of Medical Image Analysis and Understanding 2009 (MIUA 2009)*, pp. 209–213, 2009.

[102] Murray, R., Li, Z., and Sastry, S., *A mathematical introduction to robotic manipulation*. CRC, 1994.

[103] Nath, N., Tatlicioglu, E., and Dawson, D., "Range identification for nonlinear parameterizable paracatadioptric systems," *Automatica*, 2010.

[104] Neumann, T., "Modeling insect compound eyes: Space-variant spherical vision," in *Biologically Motivated Computer Vision*, pp. 360–367, Springer, 2010.

[105] Niethammer, M., Tannenbaum, A., and Angenent, S., "Dynamic active contours for visual tracking," *Automatic Control, IEEE Transactions on*, vol. 51, no. 4, pp. 562–579, 2006.

[106] Niethammer, M., Vela, P., and Tannenbaum, A., "Geometric observers for dynamically evolving curves," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1093–1108, 2008.

[107] Nützi, G., Weiss, S., Scaramuzza, D., and Siegwart, R., "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 287–299, 2011.

[108] Olabarriaga, S. and Smeulders, A., "Interaction in the segmentation of medical images: A survey," *Medical image analysis*, vol. 5, no. 2, pp. 127–142, 2001.

[109] Osher, S., "Level set methods," *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pp. 3–20, 2003.

[110] P. Yushkevich, J. P., Hazlett, H., Smith, R., Ho, S., Gee, J., and Gerig, G., "User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, pp. 1116–1128, 2006.

[111] Panagou, D. and Kumar, V., "Maintaining visibility for leader-follower formations in obstacle environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1811–1816, IEEE, 2012.

[112] Paragios, N. and Deriche, R., "Coupled geodesic active regions for image segmentation: A level set approach," *Computer VisionECCV 2000*, pp. 224–240, 2000.

[113] Paragios, N., "A level set approach for shape-driven segmentation and tracking of the left ventricle," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 6, pp. 773–776, 2003.

[114] Paulus, M., Gleason, S., Easterly, M., and Foltz, C., "A review of high-resolution x-ray computed tomography and other imaging modalities for small animal research.," *Lab animal*, vol. 30, no. 3, p. 36, 2001.

[115] Peng, H., Ruan, Z., Long, F., Simpson, J., and Myers, E., "V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets," *Nature biotechnology*, vol. 28, no. 4, pp. 348–353, 2010.

[116] Petitjean, C. and Dacher, J., "A review of segmentation methods in short axis cardiac MR images," *Medical Image Analysis*, p. 1, 2010.

[117] Pflüger, H. and Duch, C., "Dynamic neural control of insect muscle metabolism related to motor behavior," *Physiology*, vol. 26, no. 4, pp. 293–303, 2011.

[118] Piasecki, D., Bach, B., Orias, A., and Verma, N., "Anterior cruciate ligament reconstruction can anatomic femoral placement be achieved with a transtibial technique?," *The American journal of sports medicine*, vol. 39, no. 6, pp. 1306–1315, 2011.

[119] Pieper, S., Halle, M., and Kikinis, R., "3D Slicer," in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pp. 632–635, IEEE, 2005.

[120] Pieper, S., Lorensen, B., Schroeder, W., and Kikinis, R., "The na-mic kit: Itk, vtk, pipelines, grids and 3d slicer as an open platform for the medical image computing community," in *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pp. 698–701, IEEE, 2006.

[121] Platero, C., Tobar, M., Sanguino, J., and Poncela, J., "Appearance and shape prior alignments in Level Set Segmentation," *Pattern Recognition and Image Analysis*, pp. 282–289, 2009.

[122] Porikli, F., Tuzel, O., and Meer, P., "Covariance tracking using model update based on lie algebra," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 728–735, IEEE, 2006.

[123] Powell, M., "The BOBQYA algorithm for bound constrained optimization without derivatives," Tech. Rep. RNA2009/06, University of Cambridge, Dept. of Applied Mathematics and Theoretical Physics, 2009.

[124] Raguram, R., Frahm, J., and Pollefeys, M., "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," *Proceedings of European Conference Computer Vision*, pp. 500–513, 2008.

[125] Rother, C., Kolmogorov, V., and Blake, A., "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.

[126] Rousson, M. and Paragios, N., "Prior knowledge, level set representations & visual grouping," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 231–243, 2008.

[127] Russo, G. and Smereka, P., "A remark on computing distance functions," *Journal of Computational Physics*, vol. 163, no. 1, pp. 51–67, 2000.

[128] Sandhu, R., Dambreville, S., Yezzi, A., and Tannenbaum, A., "A nonrigid kernel-based framework for 2d-3d pose estimation and 2d image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1098–1115, 2010.

[129] Sato, H., Berry, C., Peeri, Y., Baghoomian, E., Casey, B., Lavella, G., VandenBrooks, J., Harrison, J., and Maharbiz, M., "Frontiers: Remote radio control of insect flight," *Frontiers in Integrative Neuroscience*, vol. 3, 2010.

[130] Sato, H., Berry, C., Peeri, Y., Baghoomian, E., Casey, B., Lavella, G., VandenBrooks, J., Harrison, J., and Maharbiz, M., "Remote radio control of insect flight," *Frontiers in integrative neuroscience*, vol. 3, 2009.

[131] Sethian, J., *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 2000.

[132] Sevilla-Lara, L. and Learned-Miller, E., "Distribution fields for tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1910 –1917, june 2012.

[133] Sharp, C., Shakernia, O., and Sastry, S., "A vision system for landing an unmanned aerial vehicle," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 1720–1727, IEEE, 2001.

[134] S.Lankton and A.Tannenbaum, "Localizing region-based active contours," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2029–2039, 2008.

[135] Smyshlyaev, A., Meurer, T., and Krstic, M., "Further results on stabilization of shock-like equilibria of the viscous burgers pde," *Automatic Control, IEEE Transactions on*, vol. 55, no. 8, pp. 1942–1946, 2010.

[136] Sontag, E., "Input to state stability: Basic concepts and results," *Nonlinear and Optimal Control Theory*, pp. 163–220, 2008.

[137] Strecha, C., Bronstein, A., Bronstein, M., and Fua, P., "LDAHash: Improved matching with smaller descriptors," *to appear IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[138] Szeliski, R., "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[139] Teare, S. and Restaino, S., *Introduction to image stabilization*, vol. 73. Society of Photo Optical, 2006.

[140] Terzopoulos, D. and Szeliski, R., *Tracking with Kalman snakes. Active Vision, A. Blake and A. Yuille*. MIT Press, Cambridge, 1992.

[141] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., and others, "Stanley: The robot that won the darpa grand challenge," *The 2005 DARPA Grand Challenge*, pp. 1–43, 2007.

[142] Tsang, W., Stone, A., Aldworth, Z., Otten, D., Akinwande, A., Daniel, T., Hildebrand, J., Levine, R., and Voldman, J., "Remote control of a cyborg moth using carbon nanotube-enhanced flexible neuroprosthetic probe," in *Micro Electro Mechanical Systems (MEMS), 2010 IEEE 23rd International Conference on*, pp. 39–42, IEEE, 2010.

[143] Vazquez, R., Coron, J.-M., Krstic, M., and Bastin, G., "Local exponential h2 stabilization of a 2 x 2 quasilinear hyperbolic system using backstepping," in *2011 CDC/ECC*, IEEE, 2011.

[144] Vela, P., Betser, A., Malcolm, J., and Tannenbaum, A., "Vision-Based Range Regulation of a Leader-Follower Formation," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 442–448, 2009.

[145] Wang, C., Liang, Z., Du, J., and Liang, S., "Robust Stabilization of Nonholonomic Moving Robots with Uncalibrated Visual Parameters," in *American Control Conference, 2009. ACC'09.*, pp. 1347–1352, 2009.

[146] Whitaker, R., "A level-set approach to 3d reconstruction from range data," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.

[147] Wu, S., Decker, S., Chang, P., Camus, T., and Eledath, J., "Collision sensing by stereo vision and radar sensor fusion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 4, pp. 606–614, 2009.

[148] Y. Gao, A. T. and Kikinis, R., "Simultaneous multi-object segmentation using local robust statistics and contour interaction," in *MICCAI Medical Computer Vision, Workshop*, 2010.

[149] Yang, W., Cai, J., Zheng, J., and Luo, J., "User-friendly interactive image segmentation through unified combinatorial user inputs," *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2470–2479, 2010.

[150] Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, 2006.

[151] Yushkevich, P., Piven, J., Hazlett, H., Smith, R., Ho, S., Gee, J., and Gerig, G., "User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[152] Zergeroglu, E., Dawson, D., de Queiroz, M., and Setlur, P., "Robust visual-servo control of robot manipulators in the presence of uncertainty," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 93–106, 2003.

[153] Zufferey, J., *Bio-inspired flying robots: experimental synthesis of autonomous indoor flyers*. EFPL Press, 2008.

[154] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axiomes et equations fondamentales du traitement d'images," *C. R. Acad. Sci. Paris*, 315:135–138, 1992.

[155] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.* **29**, pp. 845-866, 1992.

[156] G. Ben Arous, A. Tannenbaum, and Ofer Zeitouni, "Stochastic approximations of curve shortening flows," *Journal of Differential Equations* **195** (2003), pp. 119-142.

[157] S. Angenent, A. Tannenbaum, A. Yezzi, and O. Zeitouni"Curve shortening and interacting particle systems." in *Statistics and Analysis of Shapes* edited by Hamid Krim and A. Yezzi, Birhauser, 2006, pages 303-313.

[158] S. Angenent, "On the formation of singularities in the curve shortening flow," *J. Differential Geometry* **33**, pp. 601-633, 1991.

[159] V. I. Arnold, *Mathematical Methods in Classical Mechanics*, Springer-Verlag, New York.

[160] A. Blake and A. Yuille, *Active Vision*, MIT Press, Cambridge, Mass., 1992.

[161] A. Blake and M. Isard, *Active Contours*, Springer-Verlag, 1998.

[162] H. Buseman, *Convex surfaces*, Interscience Publ. (1958).

[163] H. Blum, "Biological shape and visual science," *J. Theor. Biology* **38**, pp. 205-287, 1973.

[164] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic snakes," *Int. J. Computer Vision* **22**, pp. 6179, 1997.

[165] M. G. Crandall, H. Ishii, and P.– L. Lions, "Users guide to viscosity solutions of second order partial differential equations," *Bulletin of Amer. Math. Soc.* **27**, pp. 1-67, 1992.

[166] M. Gage and R. S. Hamilton, "The heat equation shrinking convex plane curves," *J. Differential Geometry* **23**, pp. 69-96, 1986.

[167] M. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Differential Geometry* **26**, pp. 285-314, 1987.

[168] M. Grayson, "Shortening embedded curves," *Annals of Mathematics* **129**, pp. 71-111, 1989.

[169] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal curvature flows: from phase transitions to active vision," *Archive for Rational Mechanics and Analysis*, vol. 134, pp. 275–301, 1996.

[170] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "On the evolution of curves via a function of curvature, I: the classical case," *J. of Math. Analysis and Applications* **163** (1992), pp. 438-458.

[171] C. Kipnis and C. Landim, *Scaling limits of interacting particle systems*, Springer-Verlag, New York, 1999.

[172] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, Boston, 1992.

[173] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer Verlag, 2003.

[174] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics* **79**, pp. 12-49, 1988.

[175] B. ter Haar Romeny (editor), *Geometry Driven Diffusion in Computer Vision*, Kluwer, Holland, 1994.

[176] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis* Cambridge University Press, 2001.

[177] J. A. Sethian, "Curvature and the evolution of fronts," *Commun. Math. Phys.* **101**, pp. 487-499, 1985

[178] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 2nd edition, 1999.

[179] J. A. Sethian, *An Analysis of Flame Propagation*, Ph. D. Dissertation, University of California, 1982.

[180] J. Smoller, *Shock Waves and Reaction-diffusion Equations*, Springer-Verlag, New York, 1983.

[181] G. A. Sod, *Numerical Methods in Fluid Dynamics*, Cambridge University Press, Cambridge, 1985

[182] D. Terzopoulos and R. Szelski, "Tracking with Kalman snakes," in *Active Vision* edited by A. Blake and A. Zisserman, MIT Press, Cambridge, Mass., 1992.

[183] T. K. Ushijima and S. Yazaki, Convergence of a crystalline algorithm for the motion of a closed convex curve by a power of curvature $V = K^{\alpha}$, *SIAM. J. Numer. Anal.* **37** (2000), pp. 500-522.