DYNAMIC CURVE ESTIMATION FOR VISUAL TRACKING

A Thesis
Presented to
The Academic Faculty

by

Ibrahima Jacques Ndiour

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

Georgia Institute of Technology
August 2010

DYNAMIC CURVE ESTIMATION FOR VISUAL TRACKING

Approved by:

Dr. Patricio Antonio Vela, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Anthony Joseph Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Erik Verriest
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Marilyn Claire Wolf
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Jochen Teizer
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: July 12, 2010

DEDICATION

To my loving father,

Bamalick Soufi Ndiour (1947-2004).

His memory is as dear today

as in the hour he passed away.

ACKNOWLEDGEMENTS

My sincere gratitude to my advisor Dr. Patricio A. Vela to whom I remain profoundly indebted. This thesis would not have been possible without his help, support and guidance throughout the years.

I am thankful to my thesis committee: Dr. Anthony J. Yezzi for his helpful suggestions; Dr. Verriest for his detailed attention to my research; Dr. Teizer for displaying curiosity and also for providing imagery; and Dr. Wolf for remarks that helped improve the thesis.

My thanks go out to my friends and colleagues Omar Arif, Jun Yang, Gbolabo Ogunmakin, Miguel Serrano, Smriti Chopra, Ingrid Rodriguez, Juan Carlos Garcia, and Andres Roja. They brought not only brightness and intelligence to our lab, but also a lot of fun and joy.

Many thanks also go to my close friends, roommates, fellow Ph.D and so much more: Arnaud Amadjikpe, Seydou Ba, Nadia Bokassa, Mamadou Diao, Illenin Kondo, and Rodrigue Ngueyep.

My heartfelt gratitude and love to my mother. I thank her for all the love, guidance and prayers. I feel blessed to have such a loving person in my life. My love and thanks also go to my aunt Ndela, and to my brothers and sisters for their continuous support.

My deepest love and affection to my wife for all the love and care. She made this journey more agreeable than I could hope for. Her support has been invaluable. My unbounded love to my little daughter Assa who was born a week before this thesis was completed.

Last but not least, I wish to thank my departed father to whom this thesis is dedicated and to whom I owe so much. In so many ways, this successful enterprise is the reflection of his culture of excellence, which he was determined to imprint on his children. His memory remains dear to all of us who loved him.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES ix

LIST OF FIGURES x

SUMMARY xii

I INTRODUCTION 1

1.1 Prior Related Work 2

1.2 Contributions of this Thesis 3

1.3 Organization of this Thesis 4

II PRELIMINARIES 7

2.1 Shape/Curve Representations 7

2.2 Image Segmentation 10

2.2.1 Active Contour Segmentation 11

2.2.2 Graph Cut Segmentation 11

2.2.3 Bayesian Segmentation 12

2.3 Online Visual Tracking: A Recursive Bayesian Filtering Problem 13

III A PROBABILISTIC CONTOUR OBSERVER FOR ONLINE TRACKING . . 15

3.1 Introduction 15

3.1.1 Prior Related Work 15

3.1.2 Contributions 16

3.1.3 Organization 17

3.2 Observer Setup for Visual Tracking 17

3.3 Observer Components 19

3.3.1 Prediction Model 20

3.3.2 Measurement 21

3.3.3 Correction 23

3.4 Experiments and Results 26

3.4.1 Setup 26

3.4.2 Validation . . . 27
3.4.3 Comments . . . 28
3.5 Conclusion . . . 36
IV NOISE ESTIMATION AND ADAPTIVE FILTERING . . . 37
4.1 Introduction . . . 37
4.2 Quantification of Segmentation Error Through a Contrast Parameter . . 38
4.3 An Adaptive Probabilistic Filter . . . 41
4.4 Experiments and Results. . . . 42
4.5 Conclusion . . . 45
V AN OPTIMAL ESTIMATOR FOR VISUAL CONTOUR TRACKING . . . 46
5.1 Introduction . . . 46
5.1.1 Contributions . . . 47
5.1.2 Organization . . . 47
5.2 Optimal Estimator Design . . . 48
5.2.1 The Recursive Filter Structure . . . 48
5.2.2 Noise Statistics Estimation . . . 49
5.2.3 Gain Computation . . . 50
5.2.4 Filtering Equations . . . 51
5.3 Extension to Vector-Valued Images . . . 52
5.4 Algorithm and Implementation . . . 57
5.5 Complex Appearance Models and Multiple-Object Tracking . . . 58
5.6 Experiments and Results . . . 59
5.6.1 Modeling the Target and Background . . . 59
5.6.2 Optimality . . . 59
5.6.3 Comparative Performance . . . 63
5.6.4 Analysis . . . 65
5.7 Conclusion . . . 71
VI A DISTRIBUTED FILTERING TECHNIQUE . . . 72
6.1 Introduction . . . 72
6.2 Distributed Filtering for Spatial Consistency . . . 72

6.2.1 A Standard Distributed Kalman Filtering Method 73
6.2.2 Application to Visual Tracking 73
6.2.3 Implementation 74
6.3 Experiments and Results 75
6.4 Conclusion 78
VII LOCAL OPTIMAL FILTERS FOR DYNAMIC CURVE ESTIMATION . . . 79
7.1 Introduction 79
7.1.1 Prior Related Work 80
7.1.2 Contributions 80
7.1.3 Organization 81
7.2 State-Space Representation 81
7.3 Transverse Curve Coordinates 82
7.3.1 The Characteristic Vector Field 82
7.3.2 The Family of 1D Transverse Coordinate Frames. 84
7.4 First-Order Curve Filtering for Visual Tracking 85
7.4.1 Optimal First-Order Curve Filtering 85
7.4.2 Extension to Visual Tracking 87
7.5 Second-Order Curve Filtering 89
7.5.1 Dynamical Prediction Models 90
7.5.2 Measurement Model 92
7.5.3 Update Model 92
7.6 Experiments and Results 94
7.6.1 Error Measurement 94
7.6.2 Static Filtering 95
7.6.3 Tracking Using the First-Order Filter 97
7.6.4 Tracking Using the Second-Order Filter 98
7.7 Conclusion 103
VIII GEOMETRIC AVERAGING FOR STATISTICAL METHODS 104
8.1 Introduction 104
8.1.1 Contributions 104

8.1.2 Organization 105
8.2 The Geometric Averaging Filter 105
8.2.1 The Update Model 105
8.2.2 Observations 105
8.2.3 Implementation 106
8.3 Filtering the Ensemble Tracker 106
8.3.1 Ensemble Tracking 106
8.3.2 Experimentation 107
8.4 Conclusion 109
IX CONCLUSION 110
REFERENCES 113

LIST OF TABLES

1 State Motion Models. 21

2 State Correction Model. 26

3 Comparative performance of applied algorithms. 34

4 Error metrics when using alternative measurement strategies. 36

5 Filtering equations for the visual tracking system. 52

6 Filtering equations using the information form. 75

7 Quantitative comparison metrics on sequence 1. 97

8 Quantitative comparison metrics on sequence 5. 102

LIST OF FIGURES

1 Outline of the thesis. 6
2 Shape representations. 8
3 Implicit curve representations. 10
4 Structure of an observer for a visual tracking system. 18
5 Measurements of the group and shape. 19
6 Tracking a noisy infra-red sequence. 29
7 Boxplots of the comparative statistics on the infra-red sequence. 30
8 Sample frames from construction video sequences. 31
9 Snapshots of construction sequence 1. 32
10 Snapshots of construction sequence 2. 32
11 Snapshots of the Construction Sequence 3 (strikeouts indicate loss of track). 33
12 Snapshots of the Construction Sequence 4 (strikeouts indicate loss of track). 35
13 Sample frame and snapshots of the aquarium sequence. 35
14 Image samples and corresponding target/background distributions. 39
15 Segmentation error vs Bhattacharyya coefficient. 40
16 Optimal gain curve. 42
17 Adaptive filtering results for different levels of noise curruption. 43
18 Adaptive filtering results on a real sequence. 44
19 Modeling target/background with Gaussian pdf's. 60
20 Quantitative assessment of optimality on grayscale sequences. 61
21 Quantitative assessment of optimality on color sequences. 62
22 Quantitative assessment of performance on grayscale sequences. 67
23 Quantitative assessment of performance (on a highly deformable target). . . 68
24 Quantitative assessment of performance on a color sequence (1). 69
25 Quantitative assessment of performance on a color sequence (2). 70
26 Network topology for distributed filtering (4-connectivity). 74
27 Quantitative comparison of performance for sequence 1. 76
28 Quantitative comparison of performance for sequence 3. 77
29 Topology of curve comparison and transverse curve coordinates. 84

30 Expected Laplace error versus Bhattacharya coefficient. 95
31 Static filtering experiments. 96
32 Snapshots of Sequence 1. 98
33 Quantitative comparison of performance for two sequences. 100
34 Comparison of performance for sequence 4. 101
35 Snapshots of Sequence 5. 102
36 Geometric averaging applied to the ensemble tracker: sample frames. 108
37 Geometric averaging applied to the ensemble tracker: L_2 error metric. . . . 109

SUMMARY

In computer vision, visual tracking can be simply described as the action of locating/detecting a given object in an image sequence by means of an algorithm. Tracking is often trivial for humans to perform, thanks to a highly effective pair of visual sensors, the use of other senses and superior cognitive capabilities. On the other hand, automated visual tracking is difficult for a variety of reasons; the primary of which include: significant clutter, non-trivial camera motion, occlusions, imaging noise and imperfect classification models. In this thesis, the visual tracking problem is tackled as a target contour estimation problem in the face of corrupted measurements.

The major aim of this thesis is to design robust recursive curve filters for the purpose of accurate visual contour-based tracking. The state-space representation adopted comprises of a group component and a shape component describing the rigid motion and the non-rigid shape deformation respectively; filtering strategies on each component are then decoupled. Due to the infinite dimension of the shape manifold, there is not a unique filtering update model for the shape component. Shapes being often described implicitly as the iso-contours of higher dimensional functions, the filtering strategy depends on the choice of the embedding function.

This thesis considers two implicit shape descriptors, a classification probability field and the traditional signed distance function, and aims to develop an optimal probabilistic contour observer and locally optimal curve filters. For the former, introducing a novel probabilistic shape description simplifies the filtering problem on the infinite-dimensional space of closed curves to a series of point-wise filtering tasks. The definition and justification of a novel update model suited to the shape space, the derivation of the filtering equations and the relation to Kalman filtering are studied. In addition to the temporal consistency provided by the filtering, extensions involving distributed filtering methods are considered in order to maintain spatial consistency. For the latter, locally optimal closed curve filtering

strategies involving curve velocities are explored. The introduction of a local, linear description for planar curve variation and curve uncertainty enables the derivation of a mechanism for estimating the optimal gain associated to the curve filtering process, given quantitative uncertainty levels.

Experiments on synthetic and real sequences of images validate the filtering designs. While the techniques presented in this thesis are applied to planar curves, they can be extended to deal with the 3D cases involving surfaces.

CHAPTER I

INTRODUCTION

This thesis considers the problem of accurate contour-based object tracking in the face of uncertainty caused by imaging noise and approximate segmentation models. Here, an object is a portion of a scene, captured by a visual sensor, with homogeneous properties (e.g., consistent appearance information such as the intensity, color, or texture) that differentiate it from the rest of the scene. Online tracking captures the evolving object by segmenting the individual frames extracted from a video sequence. In its simplest instantiation, tracking consists of a series of statically determined contour measurements for each frame. This static approach yields good results so long as the target is sufficiently differentiated from the background. Poor differentiation arises from approximate or inadequate segmentation models, imaging noise, or occlusions. In those cases, this scheme may lose track of the target or result in degraded performance.

Imposing temporal consistency on the measurements is one strategy to manage imperfect or noisy measurements. To impose temporal consistency, [61, 88] process the image sequence in a volumetric fashion; they consider the entire video sequence at once or several frames before and after the current frame, and solve a minimization problem whose solution guarantees temporal consistency and fitness to the segmentation model. The works in [79, 87] optimize in batch a parametrized model for the target and the scene. While such processing techniques successfully impose temporal consistency of the solution, they operate in a context where access to future measurements is allowed, or measurement delay is tolerated. This work considers tracking techniques that operate in a recursive setting. Furthermore, the specific focus is on methods that produce both a trackpoint and a contour encircling the target.

1.1 Prior Related Work

Contour-based approaches for online tracking typically consider objects to be regions bounded by closed, planar curves [12, 40]. The space of closed, planar curves forms an infinite-dimensional manifold [41]. In recent years, attention has been devoted to the geometry of the manifold of curves [47, 89] and, in particular, to geodesics of closed curves [15, 41]. Closed curve geodesics have utility within the contexts of shape comparison and shape analysis. The fundamental difficulty is that there is no unique metric for comparing curves [1]. Given these facts, there is no longer a unique method for generating curve comparisons for filtering, nor is it obvious which method is optimal for a given tracking objective. This has a profound implication as to how we can fuse two curves, say prediction and measurement, to get a final corrected curve during the filtering procedure. With regards to that aspect, this work constitutes an attempt to design suitable mechanisms to resolve curves.

The incorporation of shape constraints or of shape consistency benefits the tracking procedure. Many techniques that are robust to image-based disturbances utilize *a priori* shape information [21, 42]. A collection of shapes assumed to represent the shape space is typically analyzed using low-dimensional representation methods, e.g., principal component analysis (PCA) or kernel PCA (kPCA), yielding a finite-dimensional approximation for the actual infinite-dimensional shape space. The finite-dimensional representations factor into the segmentation algorithms so as to constrain the measurements. Extensions to shape-based methods include temporal modeling of the low-dimensional learned shape space [19, 20, 22]. Superb results are obtained using such techniques when the approximated low-dimensional space is suited to the video sequence at hand [22]. However, while it is plausible to assume sufficient prior knowledge of the target shape and movement for tracking certain sequences (rigid-body objects, cyclic shape deformation for a walking person from a single viewpoint), for many scenarios it is unrealistic to make any assumption about the geometry and movement of the target. In an unconstrained setting, the target is able to undergo any

[1] For example, in the case of curve evolution, the H^0 metric yields the classic active contours family among which we classify the geodesic active contours [12] while the H^1 metric yields the Sobolev active contours [76].

feasible deformation of its shape which the tracker should accurately recover.

The online tracking problem can be viewed as an estimation problem given temporally correlated measurements. A Markovian assumption simplifies the problem to one of recursive estimation, for which one solution involves the use of observers [28]. An observer reconciles a prediction of the current target state and a measurement of the current target state through a correction mechanism. Typically, the measurements are directly obtained from a sensor. In visual tracking, rather than coming from a sensor, the measurements are considered to be the output of a tracking algorithm.

Observers have been previously used to design visual tracking systems [23, 54, 68, 77, 84]. In [23], PCA is used to reduce the shape space, whereby unscented Kalman filtering is performed on the resulting finite-dimensional space. The reduced finite dimensional shape space is of great benefit: filtering strategies on finite-dimensional spaces (such as Kalman, extended Kalman, unscented Kalman, particle, etc.) may be used to filter the shapes [68, 84]. The drawback of such finite-dimensional shape approximations is related to the training step. In most cases, training requires carefully choosing the training set, performing the reduction analysis, and possibly learning the state evolution model in the reduced space, while facing the out-of-sample problem.

In contrast, the method in [59] derives an observer on the full infinite-dimensional manifold. The observer state implicitly encodes for the bounding contour through a signed distance function. However, on account of the state choice and measurement strategy, the observer has difficulty capturing rigid-body motion of objects. These limitations are overcome in [54, 77].

1.2 Contributions of this Thesis

This thesis considers the problem of accurate trackpoint and boundary tracking of an unconstrained object given an image sequence. The hypothesis of the work is that the appropriate construction of dynamic estimators for the track object's state, whose shape is determined by an implicit contour, will lead to improved tracking performance. The estimators provide a mechanism for the temporal history to influence the final segmentation of a given

frame. A consequence of the estimation is improved segmentations with more coherent and smooth target tracking. With regards to the estimators, object dynamics are decomposed into a group component and a shape component, with a subsequent decoupling of filtering strategies. The thesis will explore different instantiations of the state-space representation and modelled uncertainty.

We believe there are several benefits associated to this approach. First, the estimator design is agnostic of the measurement strategy, i.e. any segmentation algorithm can be used to generate the measurements. With respect to the measurements, this work presents a mechanism to augment existing segmentation-based tracking techniques rather than to replace them. Secondly, as opposed to techniques using low-dimensional representations for the shape space, we examine temporal consistency in an infinite-dimensional non-parametrized setting. In practice, this translates into the proposed strategy not requiring training steps and being immune to the out-of-sample problem, while still enforcing temporal consistency. Finally, defining robustness as the ability to track with minimal deviation from the truth, the approach presented here exhibits increased robustness. In particular, the estimators presented here enable accurate contour tracking that results in successful tracking over extended periods of time.

1.3 Organization of this Thesis

The thesis is organized as follows (see Figure 1):

- **Chapter 2** provides background material to better understand the contributions developed in the subsequent chapters of the thesis.

- **Chapter 3** presents a probabilistic contour observer for visual tracking. Using a sensible state-space representation and a probabilistic shape description, the observer design is explored. In particular, a dynamical prediction model and a novel geometric averaging update strategy are proposed.

- **Chapter 4** proposes a procedure to characterize segmentation-based visual tracking performances with respect to imaging noise. The chapter develops a methodology for

utilizing a contrast parameter to derive expected segmentation errors, and an empirical procedure for identifying the optimal filter gain given the measured contrast. The resulting methodology is used in conjunction with the observer presented in chapter 3 to obtain an adaptive shape filter.

- **Chapter 5** expands the work developed in chapter 3 to propose an optimal estimator. In contrast to the previous chapters, the optimal correction gain is now formally tied to quantitative uncertainty levels on the image sequence.

- **Chapter 6** provides an extension of chapter 5. A distributive filtering scheme is proposed in order to achieve spatial consistency.

- **Chapter 7** considers the task of closed curve filtering. The principal contribution of the chapter is the derivation of a mechanism for estimating the optimal gain associated to the curve filtering process for planar curves, given quantitative uncertainty levels.

- **Chapter 8** further extends the geometric averaging update model to other statistical methods in computer vision.

- **Chapter 9** concludes the thesis and discusses potential research directions.

The contributions listed in this thesis have been partially published in [49, 50, 51, 52, 53, 54, 55, 85].

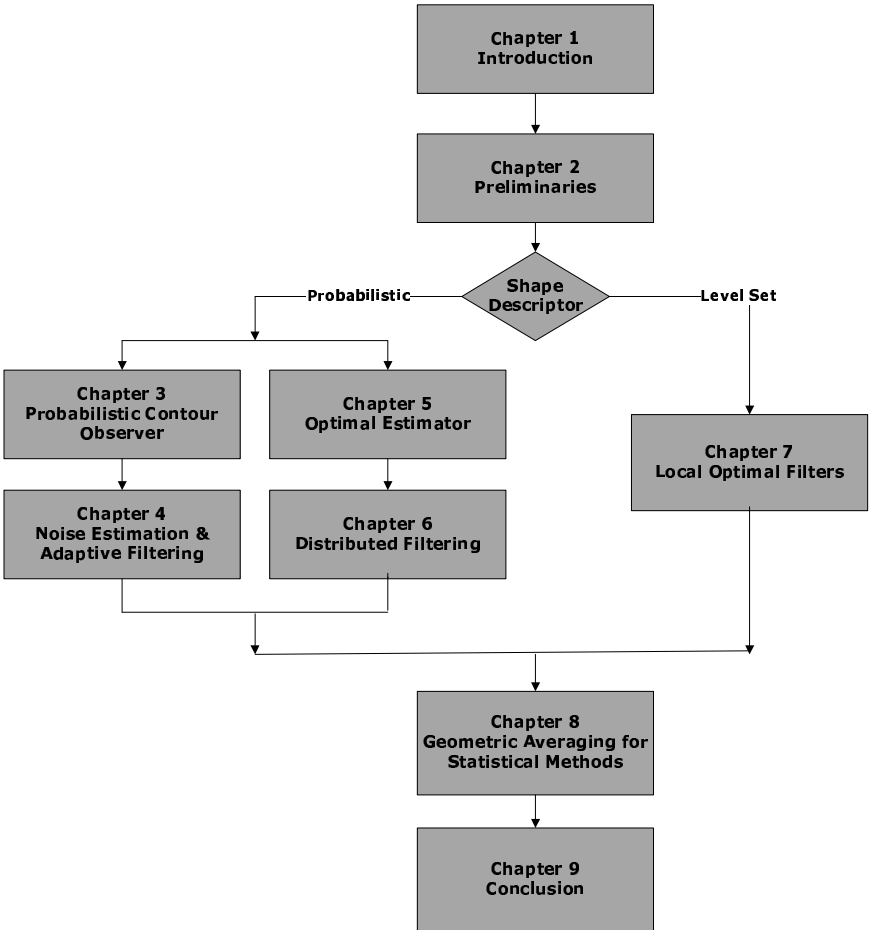


Figure 1: Outline of the thesis.

CHAPTER II

PRELIMINARIES

2.1 Shape/Curve Representations

Statistician David G. Kendall defined shape as all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. In other words, the shape of a set is all the geometrical information that is invariant to position (including rotation) and scale.

Historically, there have been many ways to represent shapes. *Yilmaz et al* [91] provide an accurate classification of shape models used in computer vision. Following their classification, we can distinguish:

1. *point representations* that describe objects using a point or set of (feature) points.
2. *primitive geometric shapes* that use rectangles, circles or ellipses to represent rigid body objects.
3. *contours and silhouettes* that define the boundary of an object and the region inside that boundary to describe complex non-rigid shapes.
4. *articulated shape models* that represent objects composed of body parts held together with joints.
5. *skeletal models* that describe shapes through objects skeletons.

This thesis considers curve (contour) models to describe object shapes. A planar curve is described as the image of a 1-parameter map into the plane. In the following, the curve and its parameter are denoted $\mathcal{C}$ and p respectively, with $\mathcal{C} : [0, 1] \to \mathbb{R}^2$. For a continuous curve, the map has to be continuous while smooth curves require the map to be infinitely differentiable. Closed curves have the additional property that $\mathcal{C}(0) = \mathcal{C}(1)$, i.e. the map is periodic. A particular choice of the map $\mathcal{C}(p)$ is called a parametrization of the curve. For example, consider the following parametrizations of the unit circle:

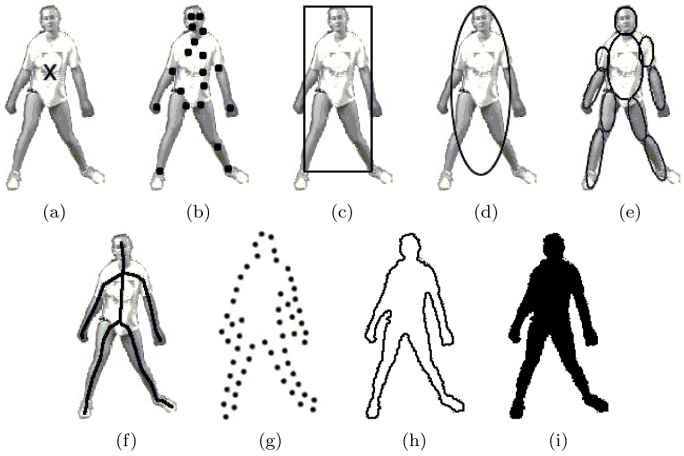


Figure 2: Shape representations: (a) centroid, (b) set of points, (c) rectangle, (d) ellipse, (e) part-based multiple patches, (f) skeleton, (g) discretized contour, (h) contour, (i) silhouette.

1. $\mathcal{C}(p) = (cos\,(2\pi p)\,, sin\,(2\pi p))$ with $p \in [0, 1]$.

2. $\mathcal{C}(p) = \left(cos\left(2\pi p^2\right), sin\left(2\pi p^2\right)\right)$ with $p \in [0, 1]$.

The previous example illustrates the fact that there is no unique parametrization of a given curve. However, geometric curve properties can be computed in a way that does not depend on the choice of parametrization. For instance, it is well known that the unit tangent $\mathcal{T}$ and unit normal $\mathcal{N}$ can be obtained from the expressions:

$$\begin{cases} \mathcal{T}(p) = \frac{\mathcal{C}'(p)}{||\mathcal{C}'(p)||} \\ \mathcal{N}(p) = J \cdot \mathcal{T}, \end{cases}$$

where J is the rotational operator defined by $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

In general, computer implementations for parametrized curves require discrete maps: the parameter p is chosen discrete. This is equivalent to discretizing the curve (Figure 2-(g)), using a finite set of points in a vector space ($\mathbb{R}^2$ for planar curves) [6, 25, 40]. Defining equivalences between curves with respect to invariant transformations (i.e. translations, rotations, and scalings) leads to a finite-dimensional shape manifold. To add a measure

of distance between curves, a metric[1] is further defined on the manifold, yielding a finite-dimensional Riemannian manifold. Note that the accuracy of these curve representations relies heavily upon the sampling/discretization, interpolation schemes (linear, polynomial, spline, etc.) being generally used to reconstruct the curve. Alternatively, polygonal approximations [81, 82] can be used to facilitate representation and modeling of the curve. The main advantage of these finite-dimensional parametric curve representation models is the simplicity of use and low computational complexity. However, they do not fully capture the continuous nature of the curve and related implementations depend on the choice of parametrization.

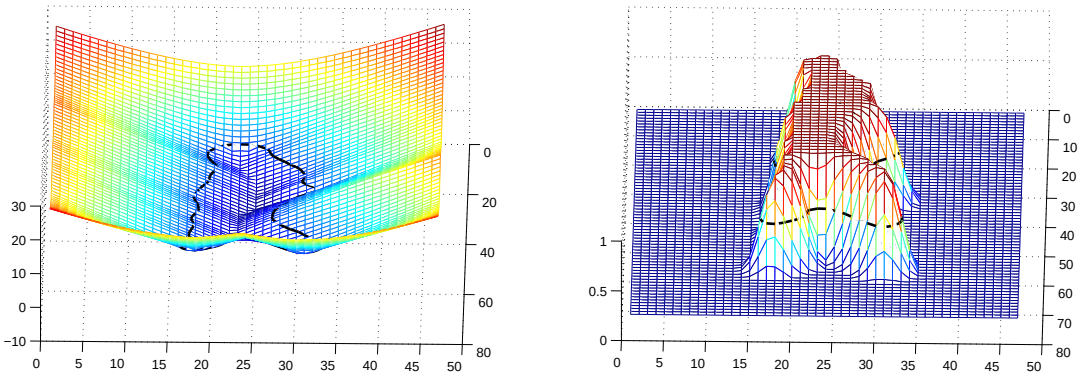
On the other hand, there exists implicit curve representations that enable geometric segmentation/tracking models independent of parametrization choices. The curve is then described in the continuum (Figure 2-(h)) and is formally defined as an element of an infinite-dimensional differentiable manifold [41]. Curves are invariant by action of low-dimensional Lie groups, e.g. $SE(2)$, while high-dimensional diffeomorphisms smoothly map one curve into another [15, 41]. Often, these implicit curve representations use a real-valued function defined over the plane: $\Psi : \mathbb{R}^2 \to \mathbb{R}$. The curve $\mathcal{C}$ is then given by the set $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 \mid \Psi(x, y) = \alpha\}$, where α is a fixed constant. For example, consider the following implicit representations for the unit circle:

1. $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 \mid \Psi(x, y) = 1\}$, with $\Psi(x, y) = x^2 + y^2$.

2. $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 \mid \Psi(x, y) = 0\}$, with $\Psi(x, y) = (x^2 + y^2)^2 - 1$.

The choice of the embedding function is not unique. The most popular among such infinite-dimensional representations is certainly the level set model where the curve $\mathcal{C}$ is embedded as the zero level set of a signed distance function Ψ, with Ψ satisfying certain properties,

[1] The Procrustes distance [24, 25] is an example of shape metrics.



(a) Level set representation (b) Probabilistic representation

Figure 3: Implicit curve representations.

e.g.:

$$\begin{cases} ||\nabla\Psi|| = 1 \text{ everywhere,} \\ \Psi < 0, \text{ for region inside the curve,} \\ \Psi > 0, \text{ for region outside the curve.} \end{cases}$$

In addition to lifting the dependence on parametrization choices, level set representations handle naturally changes of topology (splittings and mergings), allow the computation of geometric curve properties (e.g. local curvature, normals) and remain simple to implement. We refer the reader to [71] for more details on level sets.

This work will also use a probability field defined over the image domain, $P : \mathbb{R}^2 \rightarrow [0, 1]$. The implicitly defined curve $\mathcal{C}$ is obtained through the set $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 \mid P(x, y) = \rho\}$, where the value 0.5 is chosen for ρ. Figure 3 depicts a curve embedded implicitly with the probabilistic and level set functions.

2.2 Image Segmentation

The goal of image segmentation is to partition a given image into multiple regions in order to provide a meaningful image representation for further analysis and interpretation. Many algorithms have been proposed to perform image segmentation. In the following, three techniques among the most popular will be briefly presented.

2.2.1 Active Contour Segmentation

Active contour techniques are based on the temporal deformation of a contour according to the variations of an energy functional. The energy is defined with respect to the evolving contour and its minimum is attained when segmentation is achieved. In this framework, the segmentation task is reformulated as an energy minimization problem solved with the calculus of variations. Typically, the energy contains a so-called external component that attracts the contour towards the features of interest, e.g. edges, and an internal component that imposes smoothness of the contour. The contour is commonly embedded as the zero levelset of a higher-dimensional function, e.g. a signed distance function, and iteratively deformed until the energy reaches its minimum. The levelset framework [71] is widely used for active contour implementations, mainly because of its ability to handle changes of topology (splittings and mergings) that are otherwise very difficult to deal with.

From a different perspective, it has been shown that global minimizers for geometric models of active contour correspond to geodesic curves in a Riemannian space (with a metric derived from the image). This explains why the geodesic active contour model is also referred to as the Euclidean curve shortening flow. Incidentally, approaching the segmentation problem from this angle (minimal path extraction) has resulted in robust active contours methods [17]. For more details on active contour methods, we refer to [6, 13, 14, 40, 70].

2.2.2 Graph Cut Segmentation

Graph-based image segmentation techniques represent the problem in terms of a graph $G = (V, E)$. The nodes (vertices) $v_i \in V$ correspond to image pixels and the model assumes the existence of two terminal nodes (the source and the sink) respectively representing the "object" and "background" labels. There are two types of edges: *n-links* connecting neighboring pixels and *t-links* connecting pixels to the terminal nodes. Each n-link is assigned a non-negative weight that is a measure of similarity between the neighboring elements it connects, e.g. a decreasing function of the difference in intensity/color values or locations. Edges between two vertices of the same class should then have high weight while

edges between vertices from different classes should have lower weights. Similarly, each t-link is assigned a weight representing the penalty of assigning the corresponding node and terminal. The graph is partitioned into disjoint classes by removing the edges connecting classes. Given this setup, the segmentation task becomes equivalent to finding the min-cut of the graph. Additional information on graph cut segmentation methods can be found in [7, 8] and references therein.

2.2.3 Bayesian Segmentation

The Bayesian image segmentation algorithm relies on statistical analysis of the image sequence with classification done through a *maximum a posteriori* approach; it is sometimes referred to as knowledge-based segmentation. The *maximum a posteriori* (MAP) segmentation algorithm with Bayesian update, as implemented for image processing, is an adaptive thresholding algorithm that has found much success in processing and quantizing noise corrupted imagery [31, 32].

The Bayesian segmentation algorithm interprets an image to be the composition of several layers, each of which is described by a class $c \in C$ from a collection of classes. Each class has associated to it a distribution describing the expected data values v of the class, $\Pr(v|c)$. Such distributions, also called likelihoods, are commonly assumed to be Gaussian. Lastly, for each class c, there is an *a priori* probability of a pixel i being assigned to that particular class $\Pr(c_i = c)$. The Bayesian classifier selects the most likely class for a given pixel based on the probability $\Pr(c_i = c \,|\, v_i = v)$ that a given pixel value v_i is associated to the class. Classification probabilities are obtained using Bayes' rule

$$\Pr(c_i = c|v_i = v) = \frac{\Pr(v_i = v|c_i = c)\Pr(c_i = c)}{\sum_\gamma \Pr(v_i = v|c_i = \gamma)\Pr(c_i = \gamma)}.$$

The 50% probability contour associated with the target class generates the segmentation. The Bayesian segmentation procedure described here can be extended to vector-valued images in order to achieve color image segmentation. More importantly, when the unimodal Gaussian model used to generate the likelihoods does not fit with the image intensity or color distribution, one can use Gaussian mixtures and update the components of such mixtures using simple expectation-maximization algorithms. Non-parametric distributions can also

be used to better describe the image. For more details on Bayesian segmentation, we refer to [31, 32].

2.3 Online Visual Tracking: A Recursive Bayesian Filtering Problem

The purpose of estimation theory [46, 72] is to accurately infer the true state of a dynamical system in real time, given noisy measurements of the system's state. The system is presumed to evolve according to an *equation of state* while (state) measurements are obtained with a mapping from the true state space to the observed space. State evolution and measurements are formally described by the set of equations:

$$\begin{cases} \mathbf{x}(t) = f(\mathbf{x}_{t-1}, \tau_{t-1}) \\ \mathbf{z}(t) = h(\mathbf{x}_t, \eta_t) \end{cases}$$

where $\mathbf{x}$, $\mathbf{z}$, f and h represent the actual state, the measurement, the state transition model, and the observation function respectively. The process and observation noises are generally assumed to be independent; here they are denoted by $\tau(t)$ and $\eta(t)$ respectively. Using the law of total probability, Bayes' rule of probability, and a Markovian assumption makes the problem tractable and enables the construction of two-step filtering solutions commonly known as recursive Bayesian filters. The assumptions of linearity for the observation and state transition functions, and normal distributions for the observation noise, the process noise, and the prior distribution of the state lead to the popular Kalman filter [39]. Weakening the linearity assumption results in generalizations of the previous filtering scheme such as the extended (EKF) [74] and unscented Kalman filters (UKF) [37, 38]. Particle filters [1] constitute a more general class of solutions when further dropping the assumption of unimodal Gaussian distribution for the evolving state density. For detailed information about the derivation of recursive Bayesian filters, see [46, 72].

In this thesis, the online visual tracking problem is examined from the estimation perspective and recursive Bayesian filters are proposed. Computer vision algorithms are typically used to interpret the raw data obtained by the visual sensors in order to provide measurements. In a classical computer vision system, measurements cannot be presumed free of disturbances. The estimators presented here account for the measurement noise to

produce more accurate estimates. In particular, the online visual tracking problem is construed as an optimal filter design problem for systems with continuous-time dynamics and discrete-time measurements.

CHAPTER III

A PROBABILISTIC CONTOUR OBSERVER FOR ONLINE VISUAL TRACKING

3.1 Introduction

This chapter considers the problem of faithful contour-based object tracking under imaging noise and approximate target/background models. Approximate segmentation models arise from the use of image formation models that are simple relative to the true 3D scene being imaged. There are two main sources of noise, camera noise and image noise. Camera noise can be modelled as Gaussian and arises from the actual sensing procedure [33]. Image noise arises from the image handling technique; sometimes the images arrive with missing information. For example, TCP/IP-based cameras are being sold for video surveillance and tracking applications. These cameras utilize lossy image compression algorithms to stream video. One artifact of these cameras is that the reconstructed signal will no longer be pixel-wise i.i.d. due to the block-based nature of most lossy compression algorithms. Within the vicinity of the object to track, the target and background image data will be correlated. Ultimately, we are dealing with a target contour estimation problem in the face of corrupted measurements.

3.1.1 Prior Related Work

Through the definition of a parametrized model and an associated monolithic probability for the temporally related measurements, a gradient descent solution to the tracking problem is found. The solution typically relies on expectation maximization, generalized expectation maximization, or a related gradient descent procedure obtained by minimizing the negative log likelihood of the probability [79, 87]. Alternatively, temporal consistency has been used to update the initialization of the detection algorithm at subsequent times [4] or process a sequence volumetrically. The latter makes sense when the objective is to process offline a pre-existing video sequence; the minimization problem is solved over the entire sequence at

once or several frames before and after the current one, as though it were a volume [60, 88]. However, such an acausal approach is not well-suited to the problem of online, recursive estimation.

Many of the techniques that are robust to noise or false positives utilize *a priori* shape information. More recent methods have incorporated temporal modeling and filtering of the *low-dimensional* learned shape space [19, 23]. However, such techniques are unrealistic in semi-constrained and unconstrained settings. Our test video sequences are constrained to people and fish, but the view angle, the object posture, and the body shape are unconstrained variables that result in a high-dimensional shape space. Relying on low-dimensional representations may no longer be feasible.

3.1.2 Contributions

The work here considers the unconstrained segmentation and tracking problem. Instead of relying on shape information to constrain the segmentation, temporal consistency will be imposed on the segmentations and track points. Temporal consistency is obtained by describing an observer for the generated measurements and using the observer states as the estimated state, rather than the measurements. Related work includes [9, 59, 77], which examined temporal consistency in an infinite-dimensional non-parametrized setting. Alternatively, the overall object motion can be decomposed into a principal fiber, consisting of a group component (the rigid pose) and a shape component, to be filtered over as in [36]. Instead of performing a joint minimization procedure as done in [36], we decouple the filtering strategies on each component of the state, group and shape, yielding filtered contour and trackpoint signals that compose to form the trackstate. A major limitation of the tracking system presented in [36] is the proposed joint minimization procedure and the choice of shape averaging as the correction method, which restricts potential targets to rigid or slowly deformable objects. Moreover, the observer components are fully embedded and described by the internal model. Here, by recasting the tracking problem as an estimation problem broken down into observer components, a more modular structure is derived. We propose a second-order model that incorporates dynamics for capturing rapid or large deformations

in shape. We also propose a probabilistic model for the shape space, which better retains information about the classification of a given pixel through time compared to a level set model where such information is unequivocally determined by proximity to the zero level set. In online tracking, relying on past information has practical utility. Thus, by choosing to use a pixel-wise probabilistic description for the internal state, overall performance is improved. Finally, we present a correction strategy adapted to the shape space description and motivate its use for tracking.

Principal contributions include: **(1)** the formulation of the tracking problem as an observer design problem on the group and shape; **(2)** the incorporation of a dynamical model for the probabilistic shape space; **(3)** the definition of a novel correction method suited to the probabilistic shape space description; and **(4)** the quantitative validation of the system's performance.

3.1.3 Organization

The chapter is organized as follows. Sections 3.2 and 3.3 respectively describe the general structure of the observer and the observer components. A demonstration of the tracking improvement is given in Section 3.4. Section 3.5 follows with a short conclusion.

3.2 Observer Setup for Visual Tracking

In control theory, *observers* or *estimators* are used to reconstruct the state of a system from (possibly incomplete) measurements [28]. In addition to estimating unmeasured state variables, an observer is capable of filtering noisy measurements. Within the context of visual tracking, the proposed observer will serve to do both: estimate unmeasured state quantities and filter noisy measurements. The structure of the proposed observer for visual tracking is given in Figure 4. It reflects the fact that filtering will be performed on the output of a visual tracking strategy rather than on the raw visual sensor data. Thus a state space must be defined for the internal states of the observer.

State description: The state of a deformable moving object comprises of a group motion (pose) and a shape component describing the rigid motion and the non-rigid deformations of the object, respectively [90]. The group motion can be described by $SE2$ or its subgroup

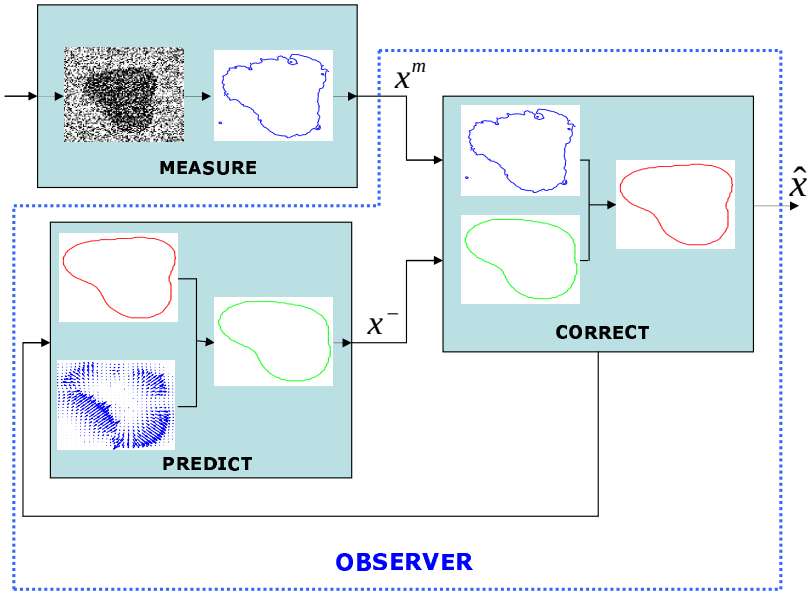


Figure 4: Structure of an observer for a visual tracking system.

$E2$ [34]. The Special Euclidean group in two dimensions $SE2$ is a Lie group formed by the set of rigid body motions in the plane. Its subgroup $E2$, the Euclidean group in two dimensions, is the set of translations in the plane. The non-rigid or shape component is given by a closed curve, meaning that the shape space is the space of closed curves. The closed curve description serves to partition the image domain into target and background regions.

A standard representation for the closed curve $\mathcal{C}$ is achieved through its embedding into a higher-dimensional space, e.g., as the level-set of a signed distance function [71]. In practice, any function capable of implicitly describing a shape through the selection of an iso-contour will suffice. This work proposes a probability field defined over the image domain, $P : D \to [0, 1]$ where $D \subset \mathbb{R}^2$ compact. The implicitly defined contour $\mathcal{C}$ is recovered through the set $\mathcal{C} = \{r \in D \mid P(r) = \rho\}$ where $\rho \in (0, 1)$. Pixels with probabilities higher than or equal to ρ are presumed to belong to the target and those with lower probabilities are presumed to belong to the background (we use the value $\rho = 0.5$). Consequently, the shape space for the work described herein is the space of probability fields defined over the image domain, an element of which will be denoted by P. We will also require the

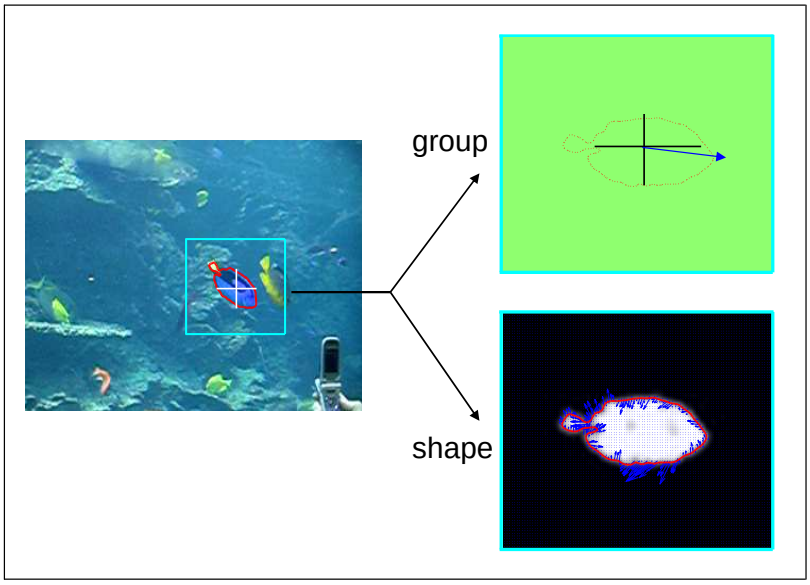


Figure 5: Measurements of the group and shape. The description of a moving object is given by a group component that encodes for gross movement in the image frame and a shape component that encodes for non-rigid movement. To the left, the segmented object is displayed within the image, with the group component determining the origin of the velocity vector (the top right) and the coordinate axis of the implicit shape description (the bottom right). The shape description depicts both the current shape and the shape velocities.

velocity field associated with P, here denoted by Θ. Thus, the state of the tracking system is described by the group variable g and the probability field P in conjunction with their associated velocities (g, P, ξ, Θ), see Figure 5.

3.3 Observer Components

Observer implementation requires the definition of a prediction model for the state and a correction strategy for the internal model given state measurements. State measurements serve as inputs to the observer, while the complete estimated state of the system is typically the output. The correction model is crucial to estimation performance: from a prediction and a measurement, it generates an updated estimate of the internal state of the observer. For the finite-dimensional group component, the correction is relatively straightforward. In comparison there is not a unique method to perform correction on the shape component, due to the manifold nature of the space of closed curves.

Consider a linear finite-dimensional system in state space representation. In what follows, let the subscripts and superscripts m, - and + respectively denote measurement, prediction and correction. The update equation of a finite-dimensional observer, with discrete correction times, is given by:

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix}^{+} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}^{-} + \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \cdot \begin{bmatrix} err(x_m, x^{-}) \\ err(\dot{x}_m, \dot{x}^{-}) \end{bmatrix}$$

where $err(x_m, x^{-})$ and $err(\dot{x}_m, \dot{x}^{-})$ are the residuals in "position" and "velocity". The gain matrix $K = [K_{ij}]$ for $i, j = 1, 2$, is such that all entries lie in the range $[0, 1]$.

In the present case, an ideal correction scheme on the shape of the target would be to compensate the predicted shape by using the residuals in 'position' and 'velocity' appropriately weighted with a gain matrix K. Because of the nature of the shape space and its probabilistic model, this scheme is infeasible. We propose instead a correction method adapted to the probabilistic shape description. The following sections further detail the observer inputs and components with regards to the visual tracking problem defined in §3.1.

3.3.1 Prediction Model

The prediction model uses the state estimate from the previous frame to produce an estimate at the current frame. It is derived from *a priori* knowledge about the target evolution. With strong priors about the state dynamics, this framework allows the use of a detailed motion model to predict state estimates at future times from present and past state estimates. In some cases, generic motion models exist that can sufficiently capture the system dynamics. For more specific motion models, one can appeal to available literature or otherwise derive a motion model from first principles.

Here, we propose two general purpose dynamic prediction models for the shape probability field. The motion models are summarized in Table 1, in addition to the simpler general models currently found in the literature (static and constant group velocity). The constant velocity models in Table 1 push forward the probability field according to estimated velocities of the target domain. The second model is differentiated from the first

Table 1: State Motion Models.

Static prior	$\begin{cases} \dot{g} = 0 \\ \dot{P} = 0 \end{cases}$
Constant group velocity	$\begin{cases} \dot{g} = \xi, \quad \dot{\xi} = 0 \\ \dot{P} = 0 \end{cases}$
Constant velocity (1)	$\begin{cases} \dot{g} = \xi, & \dot{\xi} = 0 \\ \dot{P} + \nabla P \cdot \Theta = 0, & \dot{\Theta} = 0 \end{cases}$
Constant velocity (2)	$\begin{cases} \dot{g} = \xi, & \dot{\xi} = 0 \\ \dot{P} + \nabla P \cdot \Theta = 0, & \dot{\Theta} + \nabla\Theta \cdot \Theta = 0 \end{cases}$

through the advection of the velocity field with the probabilities. The incorporation of second order dynamics will more effectively predict the motion of deforming targets when compared to static shape models, such as the static prior and the constant group velocity prior (as used in [90]). The transport equations for the shape and associated velocities given in Table 1 are simple partial differential equations involving the partial derivatives in time and space, respectively denoted as $\dot{X}$ and ∇X when applied to the variable X. Real-time implementations for the motion models exist [75].

3.3.2 Measurement

As depicted in Figure 4, the measurement block is not part of the observer even though it is an important component of the visual tracking system. The state measurements occur external to the observer and provide it with the input. The measurement module involves, at most, the determination of the four sub-states associated with the tracker, $(g_m, \xi_m, P_m, \Theta_m)$. In practice, the group velocity ξ_m is not normally available for measurement, as it is not directly measurable from the image, and must be estimated. While Θ_m can be approximately measured from the image sequence, doing so is optional.

3.3.2.1 Group - Probability Field

Measurement of the target pose can be achieved through any target localization strategy [18, 63, 73], or can be omitted if the segmentation procedure works sufficiently well without an initial localization. Measurement of the target shape can be achieved through any

segmentation algorithm applied to the current image, so long as the segmentation is converted to the implicit probability field description. Candidate algorithms include Bayesian segmentation [32], active contours [14], graph cuts [43], etc.

3.3.2.2 Velocity Field

The velocity field Θ_m can be measured by computing the optical flow [35] between two subsequent aligned images I_{n-1} and I_n. Since direct implementation of the optical flow field might not allow to capture large displacements, one may need to refine the algorithm in order to fully capture the field [3]. One example of such refinement would be to consider pyramidal implementations. Alternatively, rather than using the velocity field traditionally computed by the optical flow, one can use another variant based on the displacement field. The displacement field minimizes the energy functional:

$$E(u,v) = \int_D \left[I_n(x,y) - I_{n-1}(x+u, y+v)\right]^2 dx\, dy \\ + \lambda \cdot \int_D \left[||\nabla u(x,y)||^2 + ||\nabla v(x,y)||^2 \right] dx\, dy$$

The minimization of the functional can be performed through a simple iterative Gauss-Seidel procedure:

$$\begin{cases} u_{\tau+1} = u_\tau + d\tau \cdot (\lambda \Delta u + I_x \cdot [I_n - I_{n-1}(x+u, y+v)]) \\ v_{\tau+1} = v_\tau + d\tau \cdot (\lambda \Delta v + I_y \cdot [I_n - I_{n-1}(x+u, y+v)]) \end{cases}$$

where Δ denotes the Laplacian operator, τ is an artificial time parameter, $d\tau$ is the time step, (u_τ, v_τ) represents the displacement field at time τ and (I_x, I_y) is the gradient in space of the image I_{n-1}. The resulting error vector field $\Theta_{err} = (u, v)$ is able to capture large disparities between the images I_{n-1} and I_n.

3.3.2.3 Regarding the Shape Measurement

Recall that the observer is agnostic to the measurement strategy. If the segmentation algorithm does not automatically generate a probability field, then conversion to implicit probability field form is required. For example, consider an active contour implicitly represented by a signed-distance function, $\Psi : D \to \mathbb{R}$. Conversion to a probabilistic description

is realized using the regularized Heaviside function [14] whose values lie in the range $[0, 1]$,

$$P(\cdot) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\Psi(\cdot)}{\sigma_r}\right)\right),$$

or by applying the cumulative density function of the normal distribution (with zero mean) to the negative signed-distance function,

$$P(\cdot) = \mathrm{cdf}(-\Psi(\cdot); \sigma_r) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\Psi(\cdot)}{\sigma_r\sqrt{2}}\right)\right),$$

where σ_r is the standard deviation and erf denotes the Gauss error function defined by $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$. In both cases, σ_r is a regularization parameter.

3.3.3 Correction

Given state measurements, the correction model refines the current estimate of the state. Due to the non-uniqueness of the group and shape decomposition, the measurement procedure may not properly determine the necessary target state measurements for comparison against the predicted state. A registration step is required to describe the predicted and measured shapes with respect to the same coordinate frame [90]. Thus, once localization and segmentation are performed on the current image, a registration procedure is applied to match the resulting measured probability field with the predicted probability field. The registration procedure yields the group error, which is also the necessary transformation to describe the two shapes within the same coordinate system. Table 2 summarizes the correction model as detailed in the remainder of this section.

3.3.3.1 Group

In this work, the group motion is described by $SE2$, or its subgroup $E2$. It is given by a translation in the x-axis, a translation in the y-axis, and possibly a rotation of the axes. Correction on the group motion can then be done through classic filtering using a three-state vector (appropriately augmented). If a linear motion model such as the one presented in Table 1 is used, Kalman filtering is recommended. When the motion model is believed to be nonlinear, nonlinear filtering methods such as extended or unscented Kalman filtering should be considered. Detailed information about Kalman filtering can be found in [46].

3.3.3.2 Shape

This paragraph motivates the use of geometric averaging for the correction scheme through an analysis of the effect of additive noise on the shape density when using Bayesian classification to identify foreground and background regions.

From additive imaging noise to multiplicative segmentation uncertainty: Consider an image I defined over a compact domain of the plane and taking values in $\mathbb{R}$. Further, assume that measurement of the pixel intensities has been corrupted by independent, identically distributed additive Gaussian noise ν with zero mean and variance σ_ν^2. Classification is performed through Bayesian segmentation [32] with two classes: foreground and background. The two classes are modelled with a Gaussian distribution for the pixel intensities. Assuming uniform priors and a normal distribution $\mathcal{N}(\mu_F, \sigma_F^2)$ for the foreground pixels, then the measured likelihood for the corrupted pixel $I(r)$ to be classified as foreground is given by:

$$\zeta_F(r) = \sqrt{\delta} \cdot e^{-\frac{1}{2}\left(\frac{I(r)+\nu(r)-\mu_F}{\sigma_F}\right)^2},$$

where δ is a positive normalizing factor. The expression for the measured likelihood can be expanded further:

$$\zeta_F(r) = \sqrt{\delta} \cdot e^{-\frac{1}{2}\left(\frac{I(r)-\mu_F}{\sigma_F}\right)^2} \cdot e^{-\frac{1}{2}\left(\frac{\nu(r)}{\sigma_F}\right)^2} \cdot e^{-\left(\frac{\nu(r)\,(I(r)-\mu_F)}{\sigma_F^2}\right)}, \tag{1}$$

which can be rewritten as

$$\zeta_F(r) = P_F(r) \cdot \eta(r; \mu_F, \sigma_F),$$

where $P_F(r)$ consisting of the first two terms from (1) is the true classification likelihood, and $\eta(r; \mu_F, \sigma_F)$ consisting of the remaining two terms is the class measurement noise. A similar derivation holds for the background classification densities. Thus, corruption by additive noise on the image data results in multiplicative uncertainty for the foreground/background likelihoods. Further, the extension to vector valued imagery follows naturally.

Estimation with multiplicative noise: In the log-space associated to the probabilities,

$$\log \zeta_F(r) = \log P_F(r) + \log \eta(r; \mu_F, \sigma_F),$$

the uncertainty associated to the additive image noise becomes additive (but nonlinear). Now, suppose that the current estimate for the foreground probability field is $\widehat{P}_F^-$ while the current measurement of the probability field is ζ_F. Applying point-wise a constant gain, linear filtering strategy to filter the noise leads to the corrected estimate of the probability field,

$$\log(\widehat{P}_F^+) = \log(\widehat{P}_F^-) + K \left[\log(\zeta_F) - \log(\widehat{P}^-)\right].$$

Rearranging the terms,

$$\log(\widehat{P}_F^+) = (1-K)\log(\widehat{P}_F^-) + K\log(\zeta_F).$$

Returning to the density space by applying the exponential gives

$$\widehat{P}_F^+ = \left(\widehat{P}_F^-\right)^{1-K} (\zeta_F)^K .$$

Thus, a sub-optimal filtering strategy for managing uncertainty is to perform geometric averaging of the segmentation probabilities. The filtering applies to both the foreground and background probability fields.

Application to Visual Tracking: Correction on the shape will be performed using geometric averaging. The geometric averaging correction strategy for the shape probability field is applied point-wise,

$$\widehat{P}^+(r) = \left(\widehat{P}^-(r)\right)^{1-K_{11}(r)} \left(\widehat{P}_m(r)\right)^{K_{11}(r)}, \qquad \forall r \in D.$$

The parameter K_{11} is defined by the user according to the measurement noise. Low K_{11} is for high measurement noise, since the correction is biased towards the prediction. High K_{11} is for low measurement noise since the correction is biased towards the measurement. Here, the gain K_{11} is constant temporally but may vary spatially. For example, since the shape is always centered, low gain values may be applied near the image boundaries to forbid shape update where the target is known to be absent. Lastly, the gain K_{12} is set to zero due to the structure of the shape space.

3.3.3.3 Shape Velocities

The shape velocities live in the tangent space to the shape probability field, meaning that shape velocities lie in the space of 2D vector fields defined on D. Corrections on the velocity field occur through an error in the measured probability fields, Θ_{err}, and an error in the measured velocities:

$$\widehat{\Theta}^{+} = \widehat{\Theta}^{-} + K_{21} \cdot \Theta_{err}(P_m, \widehat{P}^{-}) + K_{22} \cdot (\Theta_m - \widehat{\Theta}^{-}).$$

The error vector field Θ_{err} is not uniquely defined and will vary according to the notion of geodesy as discussed in the Introduction. One method for generating the error Θ_{err} is to compute the optical flow between the measured and predicted probability fields, while another is to utilize optimal mass transport [64].

Table 2: State Correction Model.

State Component	Correction Description
Group and group velocity	Correction according to the update equations of the finite-dimensional filter adopted (Kalman, EKF or UKF).
Shape	$\widehat{P}^{+} = (\widehat{P}^{-})^{1-K_{11}} \cdot (P_m)^{K_{11}}$
Shape velocity	$\widehat{\Theta}^{+} = \widehat{\Theta}^{-} + K_{21} \cdot \Theta_{err}(P_m, \widehat{P}^{-}) + K_{22} \cdot (\Theta_m - \widehat{\Theta}^{-})$

3.4 Experiments and Results

3.4.1 Setup

The observer-based tracking system was tested on a body of construction footage and aquarium sequences. The basic measurement strategy chosen was Bayesian segmentation [32]. In addition to tracking with Bayesian segmentation, several other methods were tested on the sequences. The third algorithm used is from [36], which is a joint filtering strategy for both the group and the shape, and is labelled Deformotion. An active contour technique [70] has also been applied based on the same distributions used by the Bayesian algorithm; it is labelled AC. A standard, additional regularizing term is present in the active contour energy

whose relative weight with respect to the probability competition terms has been manually tuned for each sequence to optimize performance. The parameter regulates convergence to the object (external energy) while penalizing irregularities of the contour (internal energy) [11]. The fifth comparison technique is a shape-based method derived from [80], similar to [20] and labelled Shape. The shape-based segmentation strategy uses the same distributions as the Bayesian algorithm. Often in the literature, the shape-based trackers utilize the learned shape priors from the same image sequence to be tracked. In more realistic scenarios, one cannot presume the existence of shape information for all possible persons/objects to be tracked. Consequently, we obtained 67 sample shapes from our construction imagery database and the ten dominant eigenmodes were kept. For the aquarium sequence, 7 shapes were considered and the five dominant eigenmodes were retained. For all methods except deformotion, we applied the same filter to the group space, while the shape-based method also utilized a Kalman filter on the finite-dimensional shape parameters.

For the noisy infra-red sequence, we used a sixth comparison algorithm based on a finite dimensional approximation of the shape space. From a collection of 30 shapes extracted from the infra-red sequence, we used an autoregressive model [56] to learn the state evolution in the PCA-reduced shape space as in [19]. At each frame, the Kalman tracker operates by projecting the shape obtained through the Bayesian algorithm on the PCA space, the resulting vector is then applied to a Kalman filter and reconstructed back in the shape space for display and comparison purposes.

3.4.2 Validation

In order to validate the effectiveness of the observer, the output signal (group + shape) was compared against a ground truth signal using a variety of error metrics. Hand-segmentations of all sequences served as the ground truth. For the group signal, we used the L_2 and L_∞ errors with respect to the ground truth. For the shape signal, we used the number of misclassified pixels, the Hausdorff distance, and the Sobolev distance [15, 45]. For the infra-red sequence, we also used the mean and maximum Laplace error [62]. For more information about shape metrics, see [15, 29, 41, 92] and references therein. The performance summary

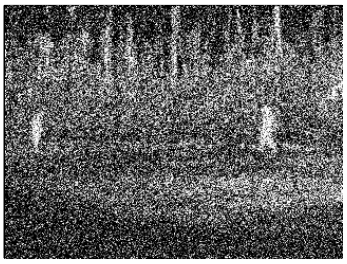
tables give the median value of the shape error computed for all frames successfully tracked by the given method, and also the maximum shape error throughout the sequence. The median value gives an indication of how well on average a given technique is performing, while the maximum value highlights the worst behavior.

The sample tracking snapshots display three chosen frames for each tracking algorithm: the left frame depicts the lowest shape error, the middle frame illustrates a sample shape error at the median, and the right frame shows the largest shape error. Strikeouts in the text indicate a loss of track.

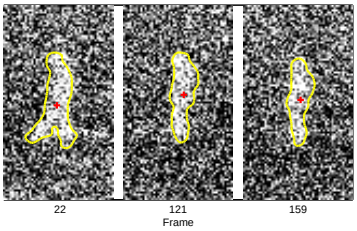
3.4.3 Comments

The first set of observations will be made on the results obtained on the noisy infra-red sequence. The Bayesian measurement strategy results in the worst performance with relatively large deviations. The quality of the segmentations is shown in Figure 6. The active contour better manages the noise on account of the regularization term, but has a tendency to undersegment. The Deformotion tracking, through its shape averaging correction, attenuates the noise, but also oversmooths as reflected by the number of misclassified pixels and the Sobolev distance. Listing the algorithms as follows {Bayesian - AC, Deformotion, Shape, Kalman}, orders them according to the quantity of prior information embedded into the algorithms (from least to most). It is therefore to be expected for the metrics to indicate a ranking of performances in that order. The Bayesian observer, lying between the Deformotion and Shape methods in term of information requirements, exhibits an ability to filter out the noise while still matching the segmented shapes to the ground truth. When compared to the other algorithms (Figure 6), the Bayesian observer has the best overall performance after the Kalman tracker. The Kalman tracker in this instance represents the best possible tracker. The dynamics of the trackstate have been fully analyzed and described in the dynamical model, plus the learned shapes were obtained directly from the image sequence. In most scenarios, it will be unlikely to have available a tracker with such prior knowledge and with the guarantee that the parametrized shape space will perfectly describe the target movement.

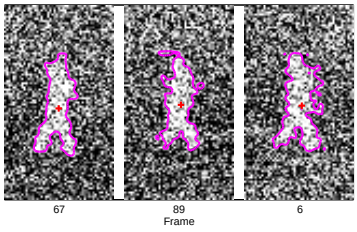
(a) Original sequence sample.



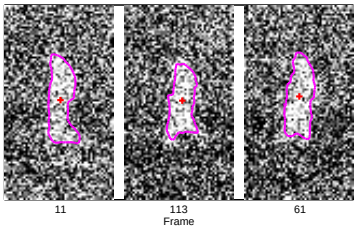
(b) Sample from noisy sequence.



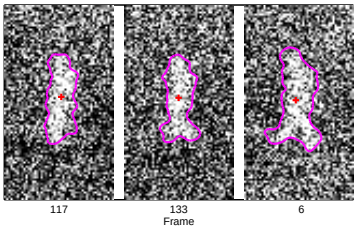
(c) Sample ground truth.



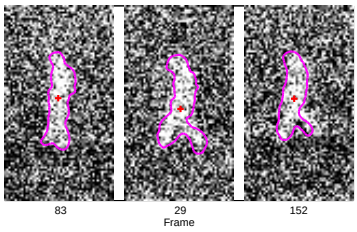
(d) Bayesian Tracker



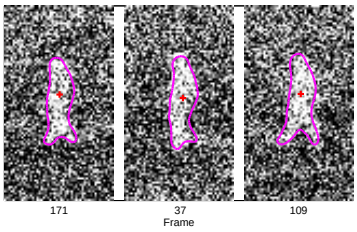
(e) Active Contour



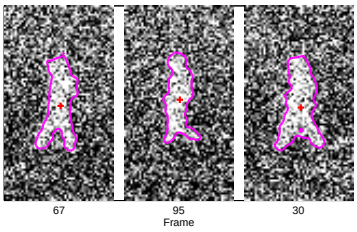
(f) Deformotion Tracker



(g) Shape-based



(h) Kalman-based Tracker



(i) Observer-based Tracker

Metric \ Algorithm	Bayesian	AC	Deformotion	Shape	Kalman	Observer
Trackpt error (L_2/L_∞)	1.8 / 4.4	1.4 / 3.9	1.2 / 3.4	3.0 / 10.5	1.4 / 5.8	1.2 / 4.4
NMP (med - max)	129 / 242	91 / 160	116 / 211	105 / 199	70 / 146	90 / 155
Hausdorff (med - max)	6.2 / 13.5	4.5 / 9.5	4.0 / 6.7	3.9 / 7.7	3.0 / 6.7	3.5 / 6.6
Sobolev (med - max)	3.2 / 10.5	2.4 / 6.9	1.5 / 3.6	1.5 / 5.4	0.8 / 8.8	1.2 / 3.3
Mean Laplace (med - max)	1.9 / 9.8	0.7 / 2.4	1.1 / 1.8	1.0 / 4.7	0.6 / 2.3	0.8 / 1.5
Max Laplace (med - max)	5.5 / 13.0	2.3 / 3.5	3.2 / 6.4	3.3 / 8.0	2.2 / 4.9	2.8 / 6.2
# Frames tracked	180	180	180	180	180	180

(j) Table summarizing the statistics

Figure 6: Noisy infra-red sequence. The left frame shows the best frame tracked, the middle frame shows how the technique operates on average and the right frame displays the worst segmentation obtained. The table displays quantitative error metrics: the median value and the maximum value are given.

The remainder of the discussion will revolve around the video sequences whose performance evaluation is depicted in Table 3. Sample frames are given in Figure 8. The table summarizes the performance results obtained using various performance measures. For each of these metrics and each algorithm, we provide the median value and the maximum value

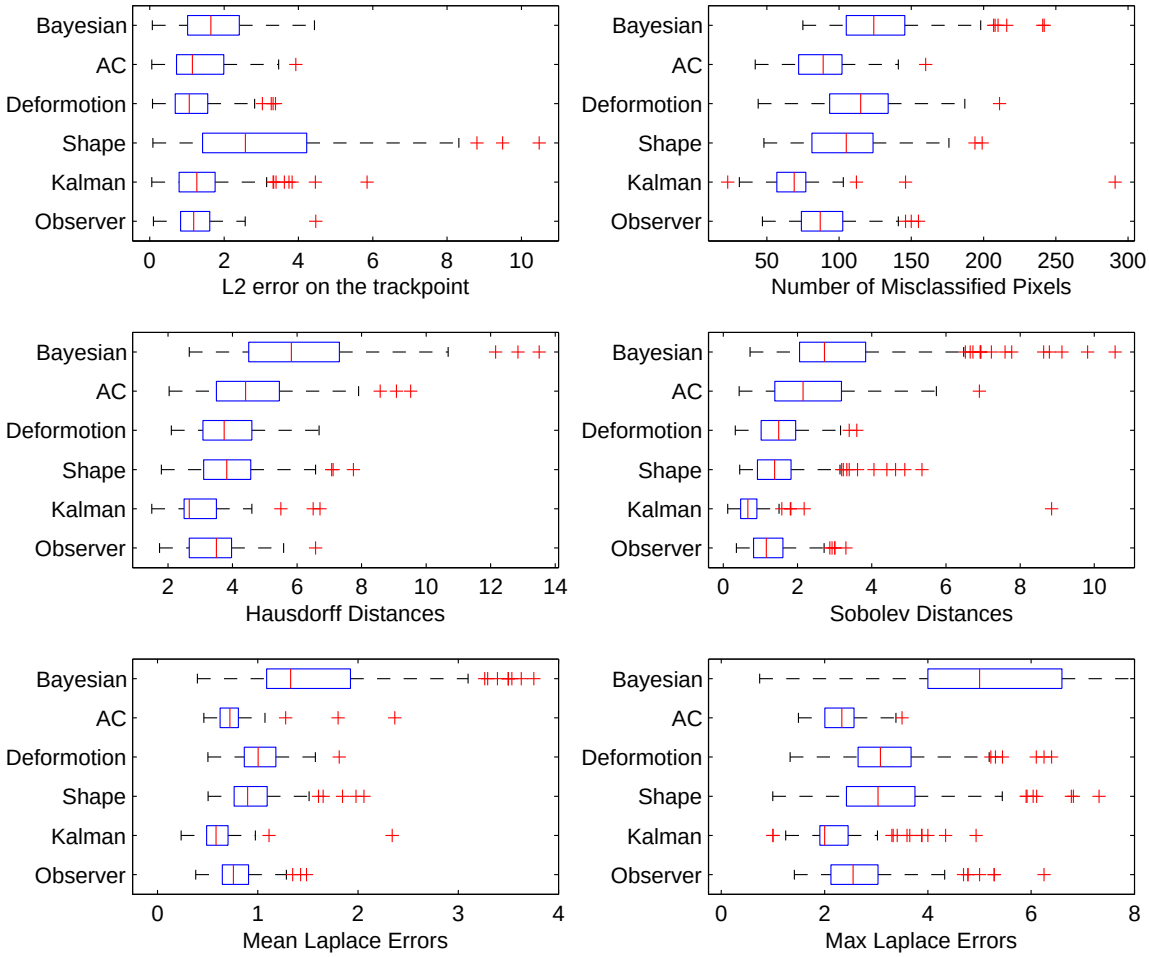


Figure 7: Boxplots of the comparative statistics on the infra-red sequence. For each metric/algorithm pair, the corresponding box represents the error distribution with lines at the lower quartile, median, and upper quartile values. The extent of the remaining data is displayed by the lines extending out from each box. Outliers are denoted by the + symbol.

of the corresponding error throughout the sequence. When a given technique fails to track the target throughout the entire sequence, the corresponding column is stricken out.

The first construction sequence (Figure 9) contains a worker that is bending at some point in the video (frame 57 to frame 162). Even though a couple of the bending segmentations were used at the training step, they did not factor into the main eigenmodes. Consequently, the shape-based technique is unable to find the correct shape for the portion of the video where the man is bending. It may be possible to improve the segmentation with the shape-based by allowing for more eigenmodes. However, to consider all possible shapes would increase the computational complexity of the technique and significantly

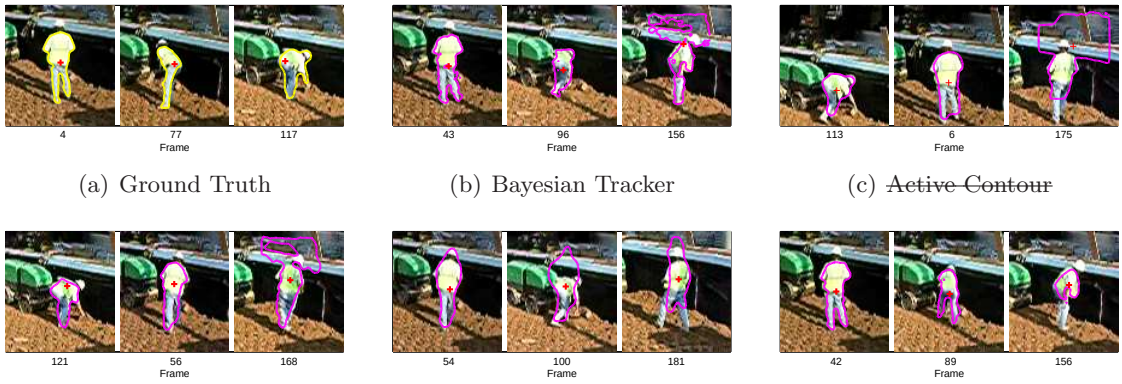
(a) Sequence 1 (b) Sequence 2 (c) Sequence 3 (d) Sequence 4

Figure 8: Sample frames from construction video sequences.

increase the size of the training set. A further difficulty of the sequence is the fact that the target and background distributions overlap somewhat, which leads to some ambiguity regarding the proper segmentation. The depiction of the worst shape error indicates that all of the methods, with the exception of the Bayesian observer, were not able to reject the segmentation disturbance. By reducing the measurement information to the segmentation boundary, the Deformotion method effectively ignores any temporal history associated to the interior target regions. The Bayesian observer maintains a history of the segmentation probabilities pixelwise and can thus more effectively reject spurious segmentation leakages that occur in individual frames. The worst case scenario for the Bayesian observer indicates a more conservative estimation of the shape than the other filter-based methods.

Sequence 2 (Figure 10) features a construction worker going down a slope. The Bayesian segmentation method performed well as reflected by the quality metrics given in Table 3. This example illustrates that by applying the Bayesian observer, one can still improve the quality of the segmentations. Not only do we enforce temporal consistency, but thanks to the second order model that incorporates dynamics, we are able to maintain a coarse definition of the contour around the legs of the man. In this case, due to the shape averaging, the Deformotion consistently fails to capture the legs as can be seen in the middle frame (Figure 10).

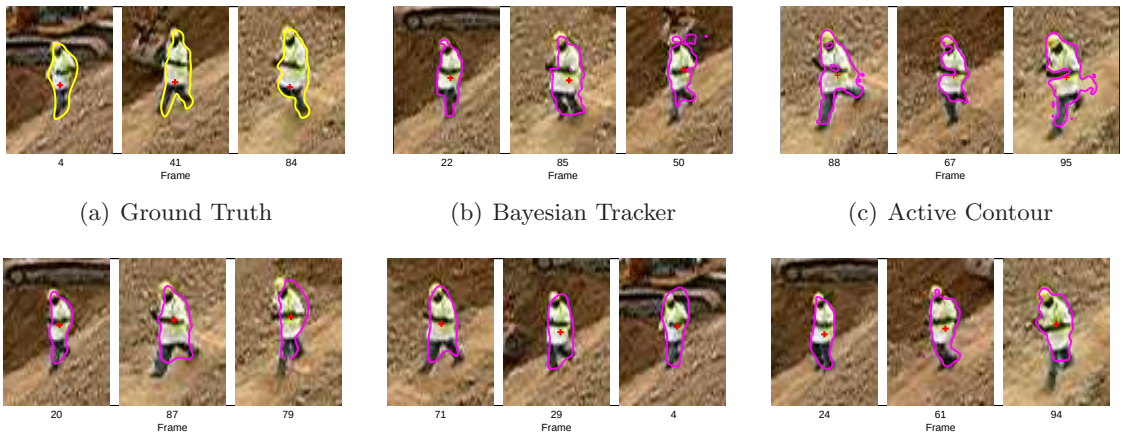
Sequence 3 (Figure 11) features a working man on a construction site that approaches another man with similar colors/distribution. Since the Bayesian tracker and the active contour obtain a trackpoint by computation of the segmentation centroid, whenever the



(a) Ground Truth (b) Bayesian Tracker (c) ~~Active Contour~~

(d) Deformotion Tracker (e) Shape-based (f) Observer-based Tracker

Figure 9: Snapshots of construction sequence 1 (strikeouts indicate loss of track).



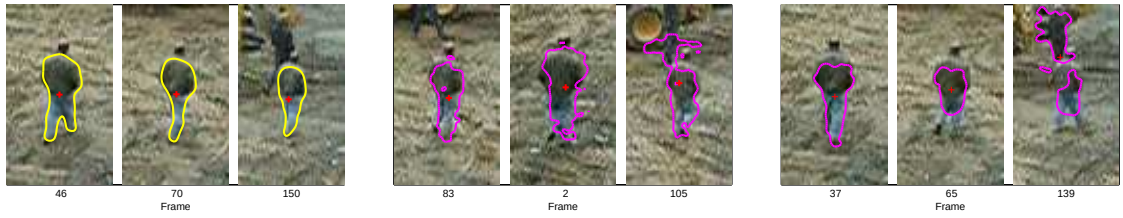
(a) Ground Truth (b) Bayesian Tracker (c) Active Contour

(d) Deformotion Tracker (e) Shape-based (f) Observer-based Tracker

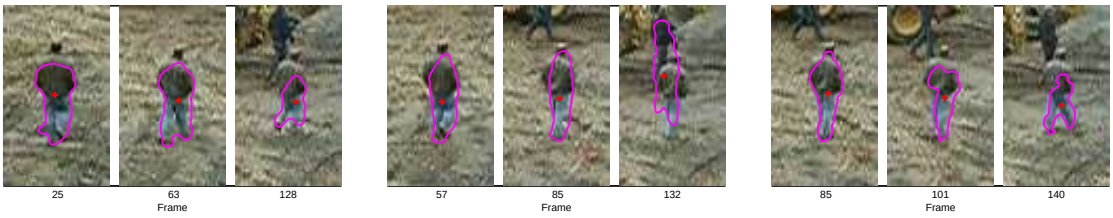
Figure 10: Snapshots of construction sequence 2.

segmentation fails and grabs portions of the background, their trackpoints deviate. Moreover, the trackpoints fluctuate due to their dependency on shape. Even though the Bayesian observer relies on the segmentation, it uses a registration procedure to decompose the track-state into group and shape and the resulting trackpoint is immune to segmentation fluctuations: the trackpoint remains anchored even in the presence of perturbations.

The last construction sequence (Figure 12) shows a worker walking around an excavator. Difficulties associated to the sequence include the small size of the target compared to the image dimensions, significant clutter, noise and partial occlusion. The Bayesian observer proves its robustness by tracking the whole sequence (1010 frames) with good performance.



(a) Ground Truth (b) Bayesian Tracker (c) ~~Active Contour~~



(d) Deformotion Tracker (e) ~~Shape-based~~ (f) Observer-based Tracker

Figure 11: Snapshots of the Construction Sequence 3 (strikeouts indicate loss of track).

By comparison, all other techniques fail to track up to frame #700 (Table 3).

The aquarium sequence (Figure 13) is an example of a moving object that undergoes large shape deformation during motion reversals. Again, there is some overlap of the target and background distributions. Visual inspection of the tracked videos show that the observer better enforces temporal consistency on the contour, with Deformotion a close second. The results are confirmed by the performance metrics.

Application to Other Measurement Methods: To demonstrate the fact that other segmentation-based tracking methods would benefit from the use of the probabilistic observer, the observer strategy was applied to tracking methods relying on active contour and graph-cut segmentation. Table 4 shows the results for two of the sample sequences. Results obtained with the segmentation methods (active contour [70] and graph cuts [8]) are improved by the observer. Originally, the active contour alone could not track the complete sequence (see Table 3-(a)). With the addition of the observer complete tracking was achieved.

The Bayesian observer compares favorably with the other techniques used here. In the face of a large variety of disturbances to the segmentation process, the observer preserves target track and improves temporal consistency of the track signal. Using a range of metrics,

Table 3: Comparative performance of applied algorithms.

(a) Construction Sequence 1

Metric \ Algorithm	Bayesian	AC	Deformotion	Shape	Filtered Bayesian
Trackpt error (L_2/L_∞)	16.6 / 24.4	~~11.5 / 52.3~~	7.9 / 16.0	5.4 / 12.3	8.0 / 15.5
NMP (med/max)	253 / 1420	~~288 / 1328~~	202 / 755	299 / 536	171 / 508
Hausdorff (med/max)	10.2 / 35.0	~~30.0 / ∞~~	7.8 / 26.2	10.9 / 25.8	7.7 / 27.4
Sobolev (med/max)	8.2 / 70.6	~~100.0 / ∞~~	5.8 / 35.3	11.7 / 38.1	6.5 / 81.8
# Frames tracked	200	~~150~~	200	200	200

(b) Construction Sequence 2

Metric \ Algorithm	Bayesian	AC	Deformotion	Shape	Filtered Bayesian
Trackpt error (L_2/L_∞)	11.5 / 13.3	8.6 / 12.4	8.3 / 12.4	6.5 / 9.1	8.3 / 12.4
NMP (med/max)	84 / 183	98 / 194	94 / 146	264 / 354	81 / 132
Hausdorff (med/max)	3.9 / 7.5	5.7 / 12.0	4.7 / 9.2	8.9 / 13.2	3.9 / 8.1
Sobolev (med/max)	1.6 / 5.0	4.5 / 33.3	3.1 / 10.8	14.0 / 23.8	2.2 / 9.2
# Frames tracked	100	100	100	100	100

(c) Construction Sequence 3

Metric \ Algorithm	AC	Deformotion	Shape	Observer
Trackpt error (L_2/L_∞)	2.7/ 6.5	2.7 / 6.5	4.5 / 20.5	2.5 / 9.5
NMP (avg/max)	131 / 276	130 / 290	162 / 330	113 / 180
Mean Laplace (avg/max)	1.2 / 2.7	1.1 / 2.8	1.9 / 4.8	1.0 / 2.1
Max Laplace (avg/max)	3.6 / 7.3	2.3 / 7.93	7.1 / 16.5	3.6 / 8.3
# Frames tracked	150	150	150	150

(d) Construction Sequence 4

Metric \ Algorithm	AC	Deformotion	Shape	Observer
Trackpt error (L_2/L_∞)	4.4/ 9.4	4.3 / 9.3	~~3.3 / 9.7~~	2.3 / 5.5
NMP (avg/max)	52 / 149	49 / 125	~~98 / 195~~	48 / 95
Mean Laplace (avg/max)	1.2 / 14.9	1.1 / 18.4	~~2.7 / 8.4~~	0.8 / 2.4
Max Laplace (avg/max)	3.0 / 18.3	2.7 / 21.5	~~8.6 / 18.9~~	2.3 / 6.4
# Frames tracked	1014	1014	~~430~~	1014

(e) Aquarium Sequence

Metric \ Algorithm	Bayesian	AC	Deformotion	Shape	Filtered Bayesian
Trackpt error (L_2/L_∞)	8.6 / 13.2	2.8 / 7.0	2.6 / 12.3	5.6 / 15.8	2.7 / 5.8
NMP (med/max)	251 / 969	244 / 549	248 / 769	575 / 833	279 / 478
Hausdorff (med/max)	10.9 / 18.4	11.1 / 19.2	12.3 / 19.7	12.0 / 22.5	14.6 / 20.7
Sobolev (med/max)	8.2 / 52.9	12.9 / 95.8	11.9 / 46.7	13.2 / 43.9	12.9 / 26.9
# Frames tracked	477	478	477	475	478

we have compared it objectively to several tracking techniques of the same family. The results of these investigations show that the observer results in equivalent performance in the absence of disturbances. When perturbation occurs, the observer eliminates or minimizes the resulting undesired segmentation effects. Even when it is not able to completely correct poor measurements, temporal consistency is maintained and the perturbations manifest as

(a) Ground Truth

(b) ~~Bayesian Tracker~~

(c) ~~Active Contour~~

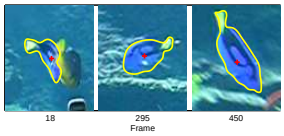
(d) ~~Deformotion Tracker~~

(e) ~~Shape-based~~

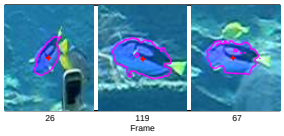
(f) Observer-based Tracker

Figure 12: Snapshots of the Construction Sequence 4 (strikeouts indicate loss of track).

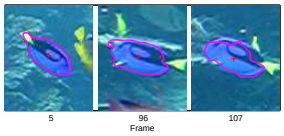
(a) Sample of Aquarium



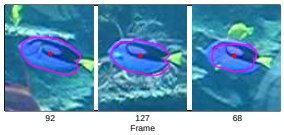
(b) Ground Truth



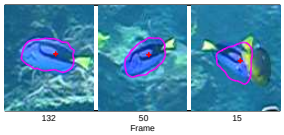
(c) Bayesian Tracker



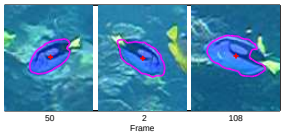
(d) Active Contour



(e) Deformotion Tracker



(f) Shape-based



(g) Observer-based Tracker

Figure 13: Sample frame and snapshots of the aquarium sequence.

conservative segmentations (Figures 10 and 12), rather than a noisy shape. The technique has minimal computational complexity and can easily run at frame rate (in a Matlab implementation, we can achieve $2-3$ frames per second or more, depending on the target size.)

Table 4: Error metrics (for observer) when using measurements obtained from an active contour and graph cut segmentations.

(a) Noisy infra-red sequence

Metric \ Algorithm		Filtered AC	Graph Cut	Filtered Graph Cut
Trackpt error	(L_2/L_∞)	2.5 / 5.6	2.4 / 9.8	1.6 / 4.3
NMP	(med/max)	93 / 133	293 / 480	105 / 178
Hausdorff	(med/max)	4.2 / 8.7	14.1 / 16.5	4.2 / 7.9
Sobolev	(med/max)	1.6 / 4.4	23.0 / 37.6	1.7 / 6.3
Mean Laplace	(med/max)	0.9 / 4.6	9.2 / 16.1	0.9 / 2.0
Max Laplace	(med/max)	2.8 / 4.9	11.1 / 16.3	3.7 / 7.8
# Frames tracked		180	180	180

(b) Sequence 1

Metric \ Algorithm		Filtered AC	Graph Cut	Filtered Graph Cut
Trackpt error	(L_2/L_∞)	6.5 / 22.1	7.9 / 31.1	6.9 / 27.4
NMP	(med/max)	192/ 663	288 / 1014	219 /457
Hausdorff	(med/max)	8.3 / 25.4	12.8 / 32.0	11.7/ 25.9
Sobolev	(med/max)	6.2 / 35.8	8.3 / 70.8	10.2/ 80.1
# Frames tracked		200	200	200

3.5 Conclusion

This chapter presented the design of an observer with a probabilistic shape representation for the purpose of improving tracking performance of segmentation-based tracking algorithms. As part of the observer, generic dynamic prediction models are given and a local shape correction adapted to the implicit probabilistic representation is described. Experiments conducted on recorded color videos, together with objective error metrics, demonstrate that the proposed observer improves both temporal consistency and tracking robustness. In particular, the quantitative results indicate that the technique is as effective if not more than more sophisticated algorithms in an online, recursive estimation setting.

The correction algorithm currently utilizes fixed gains. An essential step to improve the technique towards adaptive filtering will involve the derivation of an optimal gain strategy for the observer based on known uncertainty levels associated to the state estimate, the prediction, and the measurement.

CHAPTER IV

NOISE ESTIMATION AND ADAPTIVE FILTERING DURING VISUAL TRACKING

This chapter proposes a procedure to characterize segmentation-based visual tracking performances with respect to imaging noise. Segmentation techniques have not only different tolerances to noise level, but also degrade differently in presence of such perturbations. This work aims to identify how imaging noise affects the target segmentation as measured through local shape metrics (Sobolev and Laplace metrics). Such a procedure would be an important calibration step prior to implementing a segmentation-based tracking system. We utilize the Bhattacharyya coefficient between the target and background intensity distributions as a proxy to estimate the segmentation error. An empirical study is conducted to establish a correspondence between the Bhattacharyya coefficient and the segmentation error. The correspondence is used to adaptively filter temporally correlated segmentations. Preliminary results are very promising and show improved performances when compared to arbitrarily fixed gains.

4.1 Introduction

For image sequences corrupted by imaging noise, the introduction of filtering schemes [23] mitigates the results of imperfect segmentations. The temporal correlation between consecutive images provides sufficient information to remedy poor segmentations that cannot be handled by optimizing the individual segmentations. This work provides an analysis of noise on the segmentation procedure and determines its effects through the use of curve comparison metrics [45, 62]. In the context of filtering, it is then possible to infer a corrective gain to handle the expected segmentation error arising from imaging noise. In order to properly incorporate the expected error into the filtering procedure, the functional relationship between the image data and the segmentation quality must be determined.

The principal contributions of the work include a methodology for utilizing a proven

contrast parameter to derive expected segmentation errors, an empirical procedure for identifying the optimal filter gain given the measured contrast, and the use of the optimal gain for probabilistic shape filtering.

4.2 Quantification of Segmentation Error Through a Contrast Parameter

Suppose that the target is a single connected object in the image to process. Let P_{in} and P_{out} be the intensity probability distribution functions (pdfs) of pixels inside and outside the object, respectively. Local to the object, an algorithm's ability to segment is directly related to the interior and exterior pdfs. Segmentation ability is related to how distinct the distributions P_{in} and P_{out} are (see Figure 14). When there is significant overlap between the target and background distributions, the segmentation is prone to errors. Conversely, when the distributions are distinct, the segmentation is reliable.

Distance between pdf's: The Bhattacharyya coefficient between two distributions p and q is defined as

$$\beta(p, q) = \int \sqrt{P_{in}(x) \cdot P_{out}(x)}\, dx.$$

It is a similarity measure between pdf's that varies in the range $(0, 1)$. High values of β indicate overlapping pdf's (and suspect segmentations), while low values indicate distinct pdf's (and reliable segmentations).

Distances between curves: Several metrics [15] exist to quantify the result of a segmentation given ground truth. While [44] utilizes the number of missclassified pixels, this work utilizes curve metrics. The Sobolev distance [45] is a shape metric for curves implicitly defined by a signed distance function; it computes pointwise errors between the two curves' signed distances. The Sobolev distance provides a local measure of curve mismatch. The Laplace distance [62] is a metric on the space of curves that locally provides the distance between curves, by computing the length of unique correspondence trajectories between the two curves.

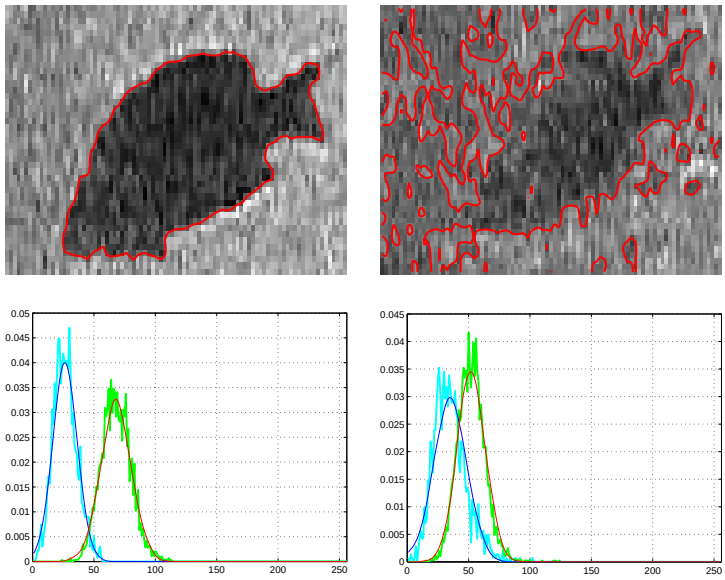


Figure 14: Image samples and corresponding target/background distributions. The left column represents a scenario where the pdf's are clearly separated ($\beta = 0.12$). The right column represents a scenario where the pdf's overlap significantly ($\beta = 0.75$). The true pdf's are given by thick lines while Gaussian-fitted pdf's are shown with fine lines.

Segmentation error vs contrast parameter: Here, we study the influence of noise on the segmentation process and use the Bhattacharyya distance in order to predict the segmentation error. Each sensor and visual tracking application will have different noise level characteristics and tolerances. The process presented here should be viewed as an important calibration step to perform before using a segmentation algorithm for tracking: it characterizes the nominal performance and response to imaging noise. The empirical uncertainty calibration is described in what follows, where we use Bayesian segmentation [30, 32].

Protocol: First, begin with a collection of shapes that will form the ground truth (preferably from an existing video sequence). Select the interior and exterior distributions to be Gaussian, $P_{in,out} = \mathcal{N}(\cdot; \mu_{in,out}; \sigma_{in,out})$. with μ_{in}/μ_{out} and σ_{in}/σ_{out} the interior/exterior Gaussian parameters. Add zero mean Gaussian noise with standard deviation $\sigma_{noise} > 0$ to the images. For each choice of σ_{noise}, generate a set of corrupted images.

Perform segmentation to yield curves partitioning the images into target and background regions. Determine the contrast coefficient, as given by the Bhattacharya distance between the interior and exterior distributions, using ground truth. Compute the curve estimation

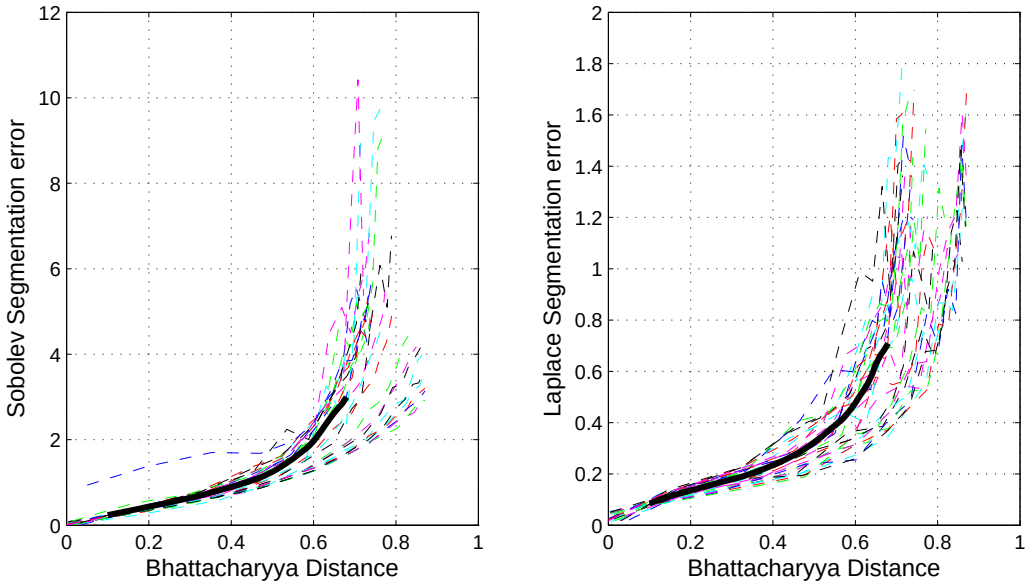


Figure 15: Segmentation error vs Bhattacharyya coefficient between target and background distributions.

error using the Sobolev and Laplace shape metrics. With these measurements, derive the expected segmentation error as a function of the Bhattacharyya distance.

Experiment: We used a collection of 36 different shapes both artificially generated, and hand-segmented from real images. The collection of shapes considered included circles of different radii, walking people, and fishes. For each noise level, σ_{noise}, 180 realizations of noisy images were generated.

Figure 15 depicts the experimental curves giving the segmentation error as a function of the contrast parameter. The mean curve is given by the thick line curve, and serves as a first approximation to the segmentation error given the Bhattacharyya coefficient. Using the fit, the Bhattacharya measure will map to the expected segmentation error.

When the target and background are clearly separable ($\beta < 0.4$), the error dependence on β is independent of shape. Due to the clear separation, the segmentations have low error. A Bhattacharyya coefficient between 0.4 and 0.7 represents the transition region from moderate to poor separation of target and background. In this range, the spread between the curves is larger. Still, the mean curve provides a consistent measure of the expected segmentation error given the Bhattacharya measure. The wider spread is due to

an increased dependency on the shape. For sufficiently low noise levels, the segmentation error is fairly independent of target shape. For significant noise levels, the segmentation algorithm performance also depends on the local curvature of the shape. Such behavior is expected given that many segmentation methods utilize curve smoothing priors during the optimization process. The error spread for high β values reflects the dependence of the error on the local shape curvature and the influence of the curve smoothing terms in the segmentation algorithm. Above a certain noise level ($\beta > 0.7$), target and background are no longer separable; the segmentation results are meaningless (the smoothing terms dominate).

4.3 An Adaptive Probabilistic Filter

This work uses the probabilistic filtering strategy described in chapter 3. Instead of the fixed gain used earlier, an adaptive gain is now sought. The error characterization process described above assessed the measurement uncertainty asociated to imaging noise. Due to the nonlinear nature of shape, there may be a nonlinear relationship between the expected segmentation error (via Sobolev of Laplace metrics) and the optimal shape correction gain during tracking. A mapping of the measurement uncertainty to the optimal gain is introduced here. In what follows, we describe an empirical evaluation of the optimal gain.

Protocol: Take video sequences with ground truth and inject a known amount of noise, σ_{noise} into the sequences. For gain values K in the range $(0, 1)$, perform the experiment at each realization of the noise level (with fixed K_{vx} and K_{vv}). Quantify, via the Sobolev or Laplace metrics, the tracking performance of the filter at the different gain levels. Collect the results together to obtain the optimal gain as a function of the Bhattacharya measure.

Experiment: The protocol was followed for a single image sequence. There were 240 different configurations for the inside/outside distributions, and the gain sweep went in increments of 0.05. The resulting functional dependence is given in Figure 16. The optimal gain for the Sobolev metric gives an almost linear dependence for about $\beta > 0.15$. The optimal gain for the Laplace metric has slight nonlinear dependence, but approximately follows the trend of the Sobolev metric optimal gain.

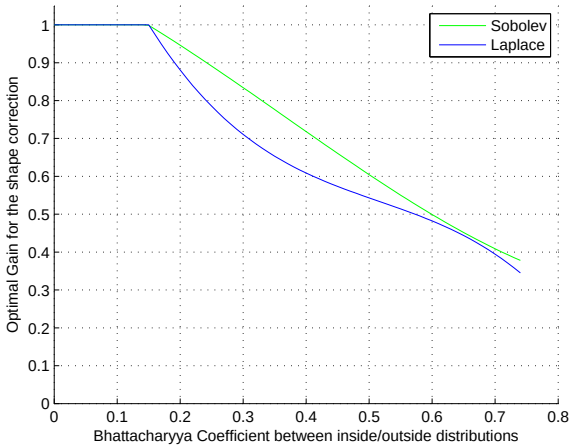
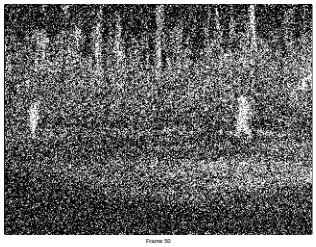


Figure 16: Optimal gain curve.

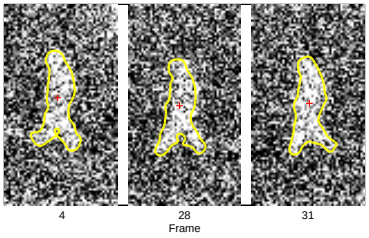
4.4 Experiments and Results.

For the experiments, we used a synthetically corrupted infrared sequence from OTCBVS and a naturally noisy aquarium sequence. The sequences were tracked with constant filter gains. Then we used the Bhattacharyya distance to adjust the gain. Because it is unrealistic to assume ground truth is available, we computed the Bhattacharyya distance with the interior/exterior distributions generated by the segmentations. Performance evaluation used the Laplace metric in conjunction with hand-segmentations (ground truth) of the sequences.

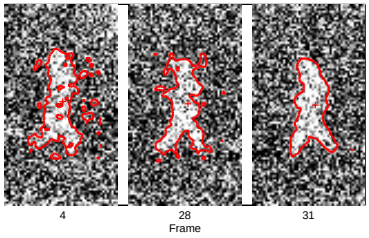
Figure 17 shows the results obtained on the infrared sequences for different noise levels. The adaptive gain has good overall performance compared to fixed gains. For naturally noisy sequences with variable noise as in the aquarium sequence (sample depicted in Figure 14), the Bhattacharyya coefficient proves to be efficient at triggering shape correction when necessary and assessing the extent to which such correction needs to be performed. Figure 18 shows the Laplace error as a function of time. The adaptive filter tracks well throughout.



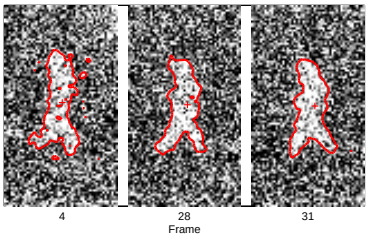
(a) Sample frame (320x240)



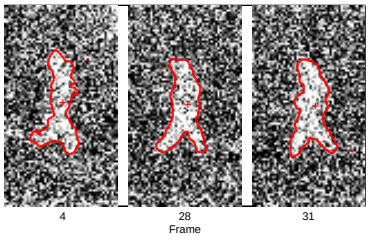
(b) Ground Truth



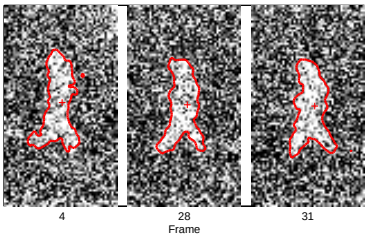
(c) Fixed Gain K = 1



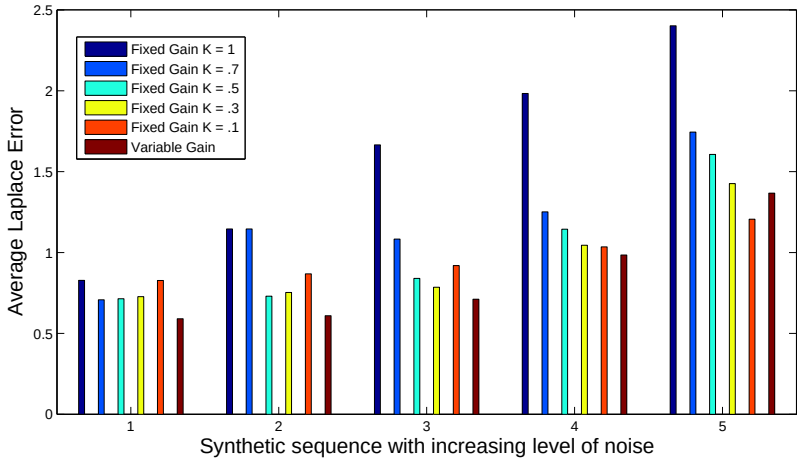
(d) Fixed Gain K = 0.7



(e) Fixed Gain K = 0.4



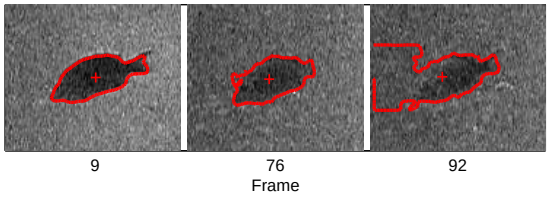
(f) Variable Gain



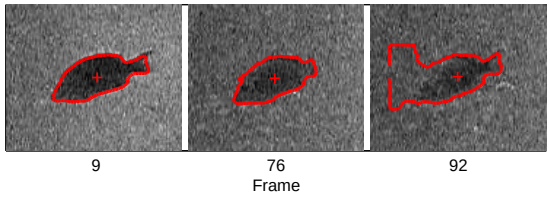
(g) Laplace error

Figure 17: Adaptive filtering results for different levels of noise curruption. For one level of noise corruption, image sample and segmentations obtained at given times for different values of the gain. For five levels of noise corruption, the average Laplace errors throughout the sequences are displayed.

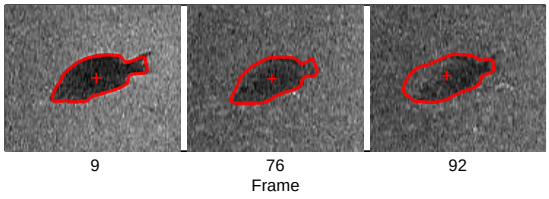
(a) Frame 1 (b) Frame 60 (c) Frame 97



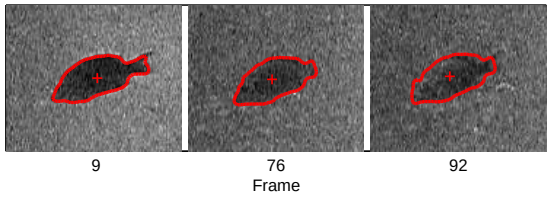
(d) Bayesian Measurements



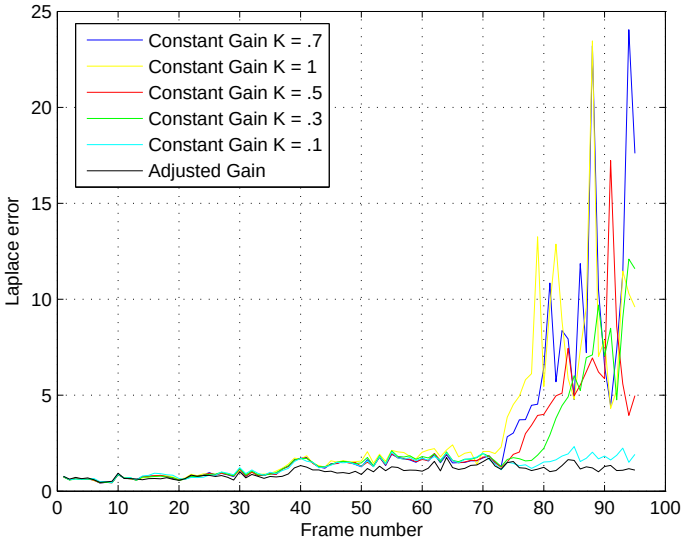
(e) Fixed Gain K = 0.5



(f) Fixed Gain K = 0.1



(g) Variable Gain



(h) Laplace error vs. time

Figure 18: Adaptive filtering results on a real sequence.

4.5 Conclusion

This chapter presented a procedure to characterize the behavior of segmentation algorithms in the presence of noise. The Bhattacharyya coefficient between target and background distributions proved to be useful for assessing segmentation errors. While such error characterization led to an effective empirical method for adaptive shape filtering, optimal filtering remained elusive.

CHAPTER V

AN OPTIMAL ESTIMATOR FOR VISUAL CONTOUR TRACKING

5.1 Introduction

This work considers the problem of accurate contour-based tracking in the presence of perturbations caused by imaging noise. Given a sequence of corrupted images $\{I_1, I_2, ... : \Omega \longmapsto \mathbb{R}\}$, performing a series of individual segmentations [6, 32] generates noisy, incorrect contour measurements that may lead, over time, to loss of track. Typical solutions to the problem of noise uncertainty include filtering the image data prior to performing visual tracking, enforcing shape constraints [19, 20, 22] or considering more complex segmentation algorithms [27, 76]. While such actions may be appropriate in certain tracking scenarios, some drawbacks mitigate the effects of these remedies and narrow their scope of use. Image filtering suffers from edge blurring that generally causes loss of accuracy for tracking applications. The incorporation of shape information can be imagined only in sufficiently constrained settings where strong assumptions about the target shape are made. Moreover, the performance of the shape-constrained techniques usually relies heavily upon the selection and modeling of a training database.

Another class of solutions is given by estimation techniques [46, 72]. Often, these approaches formalize the problem as a nonlinear estimation problem, with the intensity of each pixel being an observation. Most of the related techniques use a top-down approach, featured most prominently in [59], where the general block structure of an observer is proposed and each component is subsequently specified. There are many difficulties associated to these techniques. First, the contour to be estimated resides in an infinite-dimensional manifold space [41] requiring infinite-dimensional filtering design [36, 54, 60, 77]. This leads in general to complex filtering design and possibly large computational cost [68, 83]. Recently, there have been successful attempts to approximate the shape space through reduction methods such as principal component analysis (PCA), kernel PCA (kPCA), locally

linear embedding (LLE) and kernel LLE (kLLE) [67] in order to simplify the problem. However, derived systems [23] are then prone to out-of-sample whenever a shape outside the training set arises during tracking. Secondly, while the selection of a gain parameter is often crucial for the performance, most estimators currently proposed require manual gain selection [36, 50, 59] or quantitative uncertainty levels such as the measurement error covariance matrix [23]. Consequently, these are often chosen by the external user based on a subjective perception of visual perturbations in the sequence of images.

5.1.1 Contributions

This chapter proposes the derivation of an optimal estimator for online visual contour tracking. Instead of a top-down approach, we utilize a bottom-up approach starting from the definition of the measurement strategy, then consider the effects of noise. Bayesian segmentation [31, 32, 86] is chosen as the measurement process. In this setting, the contour encircling the target is given implicitly as the 50% iso-contour of a scalar field describing at each pixel the probability that said pixel belongs to the foreground [22]. Following up on the work presented in chapter 3, we reexamine how the hypothesis of additive imaging noise affects the classification probabilities, infer the proper update law to be applied under such hypothesis, and derive the resulting optimal filtering scheme. Benefits of this approach include the simplification of a filtering problem on the infinite-dimensional space of closed curves into a series of point-wise filtering tasks. Also, this framework allows the computation of the optimal gain, given knowledge of the uncertainty level on the image data, e.g. the noise variance. Principal contributions of the work include: **(1)** the formulation of the visual tracking task as a bottom-up filtering design problem, **(2)** the derivation of an optimal filter, and **(3)** the quantitative validation of the filter's performance.

5.1.2 Organization

The chapter is organized as follows. Sections 5.2, 5.3 and 5.4 discuss the filtering design for grayscale and vector-valued images and provide the algorithmic description for optimal estimation. Extensions to handle complex target/background appearances and multiple-object tracking follow in section 5.5. The experiments are reported in section 5.6. Section

5.7 concludes the chapter.

5.2 Optimal Estimator Design

This section provides the derivation of an optimal estimator for visual tracking on sequences of grayscale images. First, it is recalled that corruption by additive imaging noise results in multiplicative uncertainty when the measurement strategy is Bayesian segmentation. The use of a geometric averaging update model is then justified in the context of filtering with multiplicative noise. Subsequently, the noise statistics are estimated and the filtering equations and optimal gain are derived.

5.2.1 The Recursive Filter Structure

Chapter 3 established that, when using Bayesian classification to identify foreground and background regions, corruption by additive noise on the image data results in multiplicative uncertainty for the foreground/background likelihoods. We are then interested by the filtering problem [46, 72] of a process given measurements corrupted with multiplicative noise.

Here, consider a state $\rho \in (0, 1)$ to be estimated, given a collection of measurements ζ. The measurements are corrupted by multiplicative noise, i.e. $\zeta = \rho \cdot \eta$. A recursive filter following a predictor-corrector structure is proposed. Conventional estimation notation (involving hats) is used. In particular, the superscripts $-$ and $+$ denote prediction and update respectively.

5.2.1.1 Prediction

The prediction step of the filtering scheme can be chosen to be static (propagation of the previous state estimate) or dynamic, given prior knowledge on the state evolution:

$$\hat{\rho}_t^- = f(\hat{\rho}_{t-1}^+),$$

where f represents the state transition function. It is obtained from *a priori* knowledge of the state evolution. When no sufficient prior information of the state evolution is known, the generic static prediction model can be used ($f = id$).

5.2.1.2 Update

Given state measurements, the update model refines the current estimate of the state. The update step of the filtering is described by:

$$\hat{\rho}_t^+ = \left(\hat{\rho}_t^-\right)^{1-K_t} \cdot (\zeta_t)^{K_t},$$

The previous update equation can be justified by considering the logarithm of the expression:

$$\log(\hat{\rho}_t^+) = (1 - K_t) \cdot \log(\hat{\rho}_t^-) + K_t \cdot \log(\zeta_t).$$

Rearranging the terms of the previous equation yields:

$$\log(\hat{\rho}_t^+) = \log(\hat{\rho}_t^-) + K_t \cdot \left[\log(\zeta_t) - \log(\rho_t^-)\right],$$

which is the standard linear approach to filtering. The combination of the geometric averaging update model with a prediction model and the Bayesian segmentation as a measurement strategy results in a recursive filter estimating the likelihood that a pixel belong to a given class. The filtering applies to both the foreground and background.

5.2.2 Noise Statistics Estimation

In the log-space associated to the likelihoods, the foreground measurement noise has mean $-\frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^2$. The parameter σ_F is known, being defined by the Gaussian distribution used to model the foreground intensity distribution during the segmentation measurement. The variance of the imaging noise σ_ν can be estimated prior to visual tracking [65]. Given knowledge of the quantitites σ_ν and σ_F, centering of the measurement noise is done by adding the constant factor $\frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^2$ to the measurements in the log-space, which corresponds to multiplying the measured segmentation probabilities by $e^{\frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^2}$. The foreground measurement noise is now expressed as:

$$\eta_F = e^{\frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^2} \cdot e^{-\frac{1}{2}\left(\frac{\nu(r)}{\sigma_F}\right)^2} \cdot e^{-\left(\frac{\nu(r)\,(I(r)-\mu_F)}{\sigma_F^2}\right)}.$$

Using independence of the imaging noise ν from the image I, the correlation between the foreground likelihood and measurement noise vanishes:

$$S = E\left(\log\left(\rho_F\right) \cdot \log\left(\eta_F\right)\right) = 0.$$

The second order moment of the foreground measurement noise is:

$$\begin{aligned} R &= E\left(\left[\log(\eta_F)\right]^2\right) \\ &= \frac{1}{4}\left(\frac{\sigma_\nu}{\sigma_F}\right)^4 + \frac{3}{4}\left(\frac{\sigma_\nu}{\sigma_F}\right)^4 \\ &\quad + E\left(\left(\frac{\nu(r)}{\sigma_F}\right)^2\right) \cdot E\left(\left(\frac{I(r)-\mu_F}{\sigma_F}\right)^2\right) \\ &\quad - \frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^2 E\left(\left(\frac{\nu(r)}{\sigma_F}\right)^2\right). \end{aligned}$$

Since $\frac{I(r)-\mu_F}{\sigma_F}$ and $\nu(r)$ follow normal distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(0,\sigma_\nu^2)$ respectively:

$$R = \frac{1}{2}\left(\frac{\sigma_\nu}{\sigma_F}\right)^4 + \left(\frac{\sigma_\nu}{\sigma_F}\right)^2.$$

The moment R is a function of the ratio $\frac{\sigma_\nu}{\sigma_F}$. A similar analysis is valid for the background measurement noise.

5.2.3 Gain Computation

Assume a static prediction model with multiplicative process noise τ, i.e. $(\hat{\rho}_t^- = \hat{\rho}_{t-1}^+ \cdot \tau_t)$ with $\rho \in \{\rho_F, \rho_B\}$. The objective is to find the optimal value of the gain K_t that minimizes the mean squared logarithmic error $E\left(\left[\log(\rho_t) - \log(\hat{\rho}_t^+)\right]^2\right)$. This is a measure of the accuracy of the estimate $\hat{\rho}_t^+$, it is denoted $\hat{P}_t^+$:

$$\begin{aligned} \hat{P}_t^+ &= E\left(\left[\log(\rho_t) - \log(\hat{\rho}_t^+)\right]^2\right) \\ &= E\left(\log(\rho_t) - \log(\hat{\rho}_t^-) - K_t \cdot \left[\log(\zeta_t) - \log(\hat{\rho}_t^-)\right]^2\right). \end{aligned}$$

Determining the optimal gain K that minimizes the mean error is equivalent to finding the value of K for which the derivative of $\hat{P}^+$ vanishes $(\frac{\partial \hat{P}^+}{\partial K} = 0)$.

$$\begin{aligned} \frac{\partial \hat{P}^+}{\partial K} &= 2 \cdot E\left(\left[\log(\rho) - \log(\hat{\rho}^-) - K \cdot \left(\log(\zeta) - \log(\hat{\rho}^-)\right)\right] \cdot \left[\log(\hat{\rho}^-) - \log(\zeta)\right]\right) \\ &= 2 \cdot E\left(-log(\rho) \cdot \log(\zeta) + log(\rho) \cdot \log(\hat{\rho}^-) + \log(\hat{\rho}^-) \cdot \log(\zeta) - \left[\log(\hat{\rho}^-)\right]^2\right) \\ &\quad + K \cdot \left(\left[\log(\zeta)\right]^2 + \left[\log(\hat{\rho}^-)\right]^2 - 2 \cdot \log(\hat{\rho}^-) \cdot \log(\zeta)\right) \end{aligned}$$

The resulting optimal gain is then given by:

$$K = \frac{E\left(\log(\rho) \cdot \log(\zeta) - log(\rho) \cdot \log(\hat{\rho}^-) - \log(\hat{\rho}^-) \cdot \log(\zeta) + [\log(\hat{\rho}^-)]^2\right)}{E\left([\log(\zeta)]^2 + [\log(\hat{\rho}^-)]^2 - 2 \cdot \log(\hat{\rho}^-) \cdot \log(\zeta)\right)} = \frac{K_{num}}{K_{den}}$$

The prediction at the current time step $\hat{\rho}^-$ is independent from the measurement noise η, which implies $E\left(\log(\hat{\rho}^-) \cdot \log(\eta)\right) = 0$. Substituting ζ by $\rho \cdot \eta$, regrouping terms and rearranging yield the numerator:

$$\begin{aligned}
K_{num} &= E\left(\log(\rho) \cdot \log(\rho \cdot \eta) - \log(\rho) \cdot \log(\hat{\rho}^-) - \log(\hat{\rho}^-) \cdot \log(\rho \cdot \eta) + \left[\log(\hat{\rho}^-)\right]^2\right) \\
&= E\left(\left[\log(\rho)\right]^2 + \left[\log(\hat{\rho}^-)\right]^2 - 2 \cdot \log(\rho) \cdot \log(\hat{\rho}^-) + \cdot \log(\eta) \cdot \left[\log(\rho) - \log(\hat{\rho}^-)\right]\right) \\
&= E\left(\left[\log(\rho) - \log(\hat{\rho}^-)\right]^2 + \log(\rho) \cdot \log(\eta) - \log(\hat{\rho}^-) \cdot \log(\eta)\right) \\
&= \hat{P}^-
\end{aligned}$$

Similarly, substituting ζ by $\rho \cdot \eta$ and regrouping terms yield the denominator:

$$\begin{aligned}
K_{den} &= E\left(\left[\log(\rho \cdot \eta)\right]^2 + \left[\log(\hat{\rho}^-)\right]^2 - 2 \cdot \log(\hat{\rho}^-) \cdot \log(\rho \cdot \eta)\right) \\
&= E\Big(\left[\log(\rho)\right]^2 + \left[\log(\eta)\right]^2 + \left[\log(\hat{\rho}^-)\right]^2 - 2 \cdot \log(\rho) \cdot \log(\hat{\rho}^-) \\
&\quad + 2 \cdot \log(\eta) \cdot \left[\log(\rho) - \log(\hat{\rho}^-)\right]\Big) \\
&= E\left(\left[\log(\rho) - \log(\hat{\rho}^-)\right]^2 + \left[\log(\eta)\right]^2 + 2 \cdot \log(\rho) \cdot \log(\eta) - 2 \cdot \log(\hat{\rho}^-) \cdot \log(\eta)\right) \\
&= \hat{P}^- + R
\end{aligned}$$

Finally, the optimal gain is expressed as:

$$K = \frac{\hat{P}^-}{\hat{P}^- + R}$$

5.2.4 Filtering Equations

In addition to computing the gain, the error variance P needs to be estimated at the prediction step and updated at the correction step. These updates are computed using the state estimate update:

$$\begin{aligned}
\hat{P}_t^+ &= E\left(\left[\log(\rho_t) - \log(\hat{\rho}_t^+)\right]^2\right) \\
&= E\left(\left[\log(\rho_t) - (1 - K_t)\log(\hat{\rho_t}^-) - K_t \log(\zeta_t)\right]^2\right) \\
&= E\left(\left[(1 - K_t)\left[\log(\rho_t) - \log(\hat{\rho}_t^-)\right] - K_t \log(\eta_t)\right]^2\right) \\
&= (1 - K_t)^2\ \hat{P}_t^- + K_t^2\, R.
\end{aligned}$$

Assume the process noise τ to be independent from both the process and the observation noise. The predicted covariance $\hat{P}_t^-$ is then given by:

$$\begin{aligned}
\hat{P}_t^- &= E\left(\left[\log(\rho_t) - \log(\hat{\rho}_t^-)\right]^2\right) \\
&= E\left(\left[\log(\rho_{t-1}) - \log(\hat{\rho}_{t-1}^+) - \log(\tau_t)\right]^2\right) \\
&= E\left(\left[\log(\rho_{t-1}) - \log(\hat{\rho}_{t-1}^+)\right]^2\right) + E\left(\left[\log(\tau_t)\right]^2\right) \\
&= \hat{P}_{t-1}^+ + Q,
\end{aligned}$$

where $Q = E\left(\left[\log(\tau)\right]^2\right)$. These prediction and update calculations complete the derivation of the filtering equations for the system (see Table 5).

Table 5: Filtering equations for the visual tracking system.

Prediction	$\begin{cases} \hat{\rho}_t^- = \hat{\rho}_{t-1}^+ \\ \hat{P}_t^- = \hat{P}_{t-1}^+ + Q \end{cases}$
Update	$\begin{cases} K_t = \hat{P}_t^-(\hat{P}_t^- + R)^{-1} \\ \hat{\rho}_t^+ = \left(\hat{\rho}_t^-\right)^{1-K_t} \cdot (\zeta_t)^{K_t} \\ \hat{P}_t^+ = (1-K_t)^2\ \hat{P}_t^- + K_t^2\, R \end{cases}$

5.3 Extension to Vector-Valued Images

This section extends the previous optimal filtering scheme to the case of vector-valued images. Applications would include tracking on sequences of color, multi-spectral and hyper-spectral images.

Consider an image I defined over a compact domain of the plane and taking values in $\mathbb{R}^d$, $d \in \mathbb{N}$. Assume corruption of the measured pixel intensities by the additive Gaussian noise ν where ν follows a zero-mean multivariate normal distribution $\mathcal{N}(0, \Sigma_\nu)$. Similarly to the scalar case, classification performed through Bayesian segmentation using Gaussian models for the class distributions and uniform priors yield the following measurement likelihoods:

$$\zeta_F(r) = \sqrt{\Delta} \cdot e^{-\frac{1}{2}(I(r)+\nu(r)-\mu_F)^T\, \Sigma_F^{-1}\, (I(r)+\nu(r)-\mu_F)},$$

where Δ remains a positive normalizing factor. For ease of notation, the pixel location r will be dropped in the remainder of this section. When expanded, the measured likelihood

can be expressed as:

$$\zeta_F(r) = \overbrace{\sqrt{\Delta} \cdot e^{-\frac{1}{2}(I-\mu_F)^T \, \Sigma_F^{-1} \, (I-\mu_F)}}^{\rho_F} \cdot \underbrace{e^{-\frac{1}{2}(I-\mu_F)^T \, \Sigma_F^{-1} \, \nu} \cdot e^{-\frac{1}{2}\nu^T \, \Sigma_F^{-1} \, (I-\mu_F)} \cdot e^{-\frac{1}{2}\nu^T \, \Sigma_F^{-1} \, \nu}}_{\eta_F}.$$

Here again, the measured classification likelihood is given by the product of the true classification likelihood P_F and the class measurement noise η_F. It is straightforward to see that the optimal estimation retains the same predictor-corrector structure with a geometric averaging update model. While the filtering equations and the optimal gain derivation remain the same as in the scalar case, the noise statistics need to be computed. Let R^* denote the second order moment of the noise in the log-space associated to the densities:

$$\begin{aligned}
4\,R^* = {} & E\left(\left[(I-\mu_F)^T \, \Sigma_F^{-1} \, \nu\right]^2\right) \\
& + E\left(\left[\nu^T \, \Sigma_F^{-1} \, (I-\mu_F)\right]^2\right) + E\left(\left[\nu^T \, \Sigma_F^{-1} \, \nu\right]^2\right) \\
& + E\left((I-\mu_F)^T \, \Sigma_F^{-1} \nu\nu^T \, \Sigma_F^{-1} \, (I-\mu_F)\right) \\
& + E\left(\nu^T \, \Sigma_F^{-1} \, (I-\mu_F) \, (I-\mu_F)^T \, \Sigma_F^{-1} \, \nu\right) \\
& + E\left(\nu^T \, \Sigma_F^{-1} \, (I-\mu_F) \, \nu^T \, \Sigma_F^{-1} \, \nu\right) \\
& + E\left(\nu^T \, \Sigma_F^{-1} \nu\nu^T \, \Sigma_F^{-1} \, (I-\mu_F)\right) \\
& + E\left((I-\mu_F)^T \, \Sigma_F^{-1} \nu\nu^T \, \Sigma_F^{-1} \, \nu\right) \\
& + E\left(\nu^T \, \Sigma_F^{-1} \nu \, (I-\mu_F)^T \, \Sigma_F^{-1} \, \nu\right) \\
= {} & A + B + C + D + E + F + G + H + J.
\end{aligned}$$

Each term in the previous expression is denoted by an alphabetical letter corresponding to the order of appearance. In the following, these terms are evaluated using the independence of the imaging noise ν from the image I, the zero-mean normal distribution of ν, the commutativity of the linear operators *trace* and *expectation*, the symmetric property of Σ_ν, Σ_F and their inverses, and the identities $x^T\Omega x = Tr(xx^T\Omega)$ and $x^T\Sigma y = y^T\Sigma x$ (for

$x, y \in \mathbb{R}^d$, Ω a square matrix of size d and Σ a symmetric matrix of size d):

$$
\begin{aligned}
A &= E\left((I-\mu_F)^T \ \Sigma_F^{-1}\nu \overbrace{(I-\mu_F)^T \ \Sigma_F^{-1}\nu}^{\nu^T\Sigma_F^{-1}(I-\mu_F)} \right) \\
&= E\left((I-\mu_F)^T \ \Sigma_F^{-1}\nu\nu^T\Sigma_F^{-1} \, (I-\mu_F) \right) \\
&= E\left(Tr\left[(I-\mu_F)\,(I-\mu_F)^T \ \Sigma_F^{-1}\nu\nu^T\Sigma_F^{-1} \right] \right) \\
&= Tr\left[E\left((I-\mu_F)\,(I-\mu_F)^T \right) \cdot \Sigma_F^{-1} \cdot E\left(\nu\nu^T\right) \cdot \Sigma_F^{-1} \right] \\
&= Tr\left[\Sigma_F \cdot \Sigma_F^{-1} \cdot \Sigma_\nu \cdot \Sigma_F^{-1} \right] \\
&= Tr\left[\Sigma_\nu \cdot \Sigma_F^{-1} \right].
\end{aligned}
$$

$$
\begin{aligned}
B &= E\left(\nu^T \, \Sigma_F^{-1}\,(I-\mu_F) \overbrace{\nu^T \, \Sigma_F^{-1}\,(I-\mu_F)}^{(I-\mu_F)^T\Sigma_F^{-1}\nu} \right) \\
&= E\left(\nu^T \, \Sigma_F^{-1}\,(I-\mu_F)\,(I-\mu_F)^T \ \Sigma_F^{-1}\nu \right) \\
&= E\left(Tr\left[\nu\nu^T \, \Sigma_F^{-1}\,(I-\mu_F)\,(I-\mu_F)^T \ \Sigma_F^{-1} \right] \right) \\
&= Tr\left[E\left(\nu\nu^T\right) \ \Sigma_F^{-1} \, E\left((I-\mu_F)\,(I-\mu_F)^T \right) \ \Sigma_F^{-1} \right] \\
&= Tr\left[\Sigma_\nu \cdot \Sigma_F^{-1} \cdot \Sigma_F \cdot \Sigma_F^{-1} \right] \\
&= Tr\left[\Sigma_\nu \cdot \Sigma_F^{-1} \right].
\end{aligned}
$$

For ease of notation, let Ω denote Σ_F^{-1}.

$$
\begin{aligned}
C &= E\left(\left[\nu^T \, \Omega \, \nu\right]^2 \right) = E\left(\left[\sum_{ij} \nu_i \, \Omega_{ij} \, \nu_j \right]^2 \right) \\
&= E\left(\sum_{ij} \sum_{kl} \nu_i \, \nu_j \, \nu_k \, \nu_l \, \Omega_{ij} \, \Omega_{kl} \right) \\
&= \sum_{ij} \sum_{kl} E\left(\nu_i \, \nu_j \, \nu_k \, \nu_l \right) \, \Omega_{ij} \, \Omega_{kl}.
\end{aligned}
$$

Since ν is Gaussian, Wick's theorem can be applied:

$$\begin{aligned} E(\nu_i \nu_j \nu_k \nu_l) &= E(\nu_i \nu_j) \cdot E(\nu_k \nu_l) \\ &\quad + E(\nu_i \nu_k) \cdot E(\nu_j \nu_l) + E(\nu_i \nu_l) \cdot E(\nu_j \nu_k) \\ &= V_{ij} \cdot V_{kl} + V_{ik} \cdot V_{jl} + V_{il} \cdot V_{jk}, \end{aligned}$$

where V denotes the covariance matrix of ν ($V = \Sigma_\nu$ is used to alleviate notation).

$$\begin{aligned} C &= \sum_{ij} \sum_{kl} \left[V_{ij} \cdot V_{kl} + V_{ik} \cdot V_{jl} + V_{il} \cdot V_{jk} \right] \Omega_{ij} \Omega_{kl} \\ &= \sum_{ij} V_{ij} \Omega_{ji} \cdot \sum_{kl} V_{lk} \Omega_{kl} + 2 \sum_{ijkl} V_{ik} \Omega_{kl} V_{lj} \Omega_{ji} \\ &= \left(Tr \left[V\Omega \right] \right)^2 + 2\, Tr \left[\left(V\Omega \right)^2 \right]. \end{aligned}$$

Replacing V by Σ_ν and Ω by Σ_F^{-1} yields:

$$C = \left(Tr \left[\Sigma_\nu \Sigma_F^{-1} \right] \right)^2 + 2\, Tr \left[\left(\Sigma_\nu \Sigma_F^{-1} \right)^2 \right].$$

Continuing,

$$\begin{aligned} D &= E\left((I - \mu_F)^T \, \Sigma_F^{-1} \nu \nu^T \, \Sigma_F^{-1} \, (I - \mu_F) \right) \\ &= Tr \left[E\left((I - \mu_F) (I - \mu_F)^T \, \Sigma_F^{-1} \nu \nu^T \, \Sigma_F^{-1} \right) \right] \\ &= Tr \left[\Sigma_F \cdot \Sigma_F^{-1} \cdot \Sigma_\nu \cdot \Sigma_F^{-1} \right] \\ &= Tr \left[\Sigma_\nu \cdot \Sigma_F^{-1} \right], \end{aligned}$$

$$\begin{aligned} E &= E\left(\nu^T \, \Sigma_F^{-1} (I - \mu_F) (I - \mu_F)^T \, \Sigma_F^{-1} \, \nu \right) \\ &= Tr \left[E\left(\nu \nu^T \, \Sigma_F^{-1} (I - \mu_F) (I - \mu_F)^T \, \Sigma_F^{-1} \right) \right] \\ &= Tr \left[\Sigma_\nu \cdot \Sigma_F^{-1} \cdot \Sigma_F \cdot \Sigma_F^{-1} \cdot \right] \\ &= Tr \left[\Sigma_\nu \cdot \Sigma_F^{-1} \right], \end{aligned}$$

$$
\begin{aligned}
F &= E\left(\nu^T \Sigma_F^{-1} \left(I - \mu_F\right) \nu^T \Sigma_F^{-1} \nu\right) \\
&= E\left(Tr\left[\nu\nu^T \Sigma_F^{-1} \left(I - \mu_F\right) \nu^T \Sigma_F^{-1}\right]\right) \\
&= E\left(Tr\left[\Sigma_F^{-1} \left(I - \mu_F\right) \nu^T \Sigma_F^{-1} \nu\nu^T\right]\right) \\
&= Tr\left[\Sigma_F^{-1} E\left(\left(I - \mu_F\right)\right)\ E\left(\nu^T \Sigma_F^{-1} \nu\nu^T\right)\right] \\
&= 0,
\end{aligned}
$$

$$
\begin{aligned}
G &= E\left(\nu^T \Sigma_F^{-1} \nu\nu^T \Sigma_F^{-1}\ \left(I - \mu_F\right)\right) \\
&= E\left(\nu^T \Sigma_F^{-1} \nu\nu^T \Sigma_F^{-1}\right)\ E\left(\left(I - \mu_F\right)\right) \\
&= 0,
\end{aligned}
$$

$$
\begin{aligned}
H &= E\left(\left(I - \mu_F\right)^T\ \Sigma_F^{-1} \nu\nu^T \Sigma_F^{-1} \nu\right) \\
&= E\left(\left(I - \mu_F\right)^T\right)\ E\left(\Sigma_F^{-1} \nu\nu^T \Sigma_F^{-1} \nu\right) \\
&= 0,
\end{aligned}
$$

and

$$
\begin{aligned}
J &= E\left(\nu^T \Sigma_F^{-1} \nu \left(I - \mu_F\right)^T\ \Sigma_F^{-1} \nu\right) \\
&= Tr\left[E\left(\nu\nu^T \Sigma_F^{-1} \nu\right)\ E\left(\left(I - \mu_F\right)^T\right)\ \Sigma_F^{-1}\right] \\
&= 0.
\end{aligned}
$$

The final result for R^* is:

$$
R^* = \frac{1}{4}\left(Tr\left[\Sigma_\nu \Sigma_F^{-1}\right]\right)^2 + \frac{1}{2} Tr\left[\left(\Sigma_\nu \Sigma_F^{-1}\right)^2\right] + Tr\left[\Sigma_\nu \Sigma_F^{-1}\right].
$$

The mean for the foreground measurement noise is:

$$\begin{aligned} E\left(\log(\eta_F)\right) &= -\frac{1}{2}E\left(\nu^T\,\Sigma_F^{-1}\,\nu\right) = -\frac{1}{2}E\left(Tr\left[\nu\nu^T\Sigma_F^{-1}\right]\right) \\ &= -\frac{1}{2}Tr\left[E\left(\nu\nu^T\Sigma_F^{-1}\right)\right] \\ &= -\frac{1}{2}Tr\left[E\left(\nu\nu^T\right)\cdot E\left(\Sigma_F^{-1}\right)\right] \\ &= -\frac{1}{2}Tr\left[\Sigma_\nu\Sigma_F^{-1}\right]. \end{aligned}$$

Therefore, the central second-order moment R is:

$$\begin{aligned} R &= E\left(\left[\log(\eta_F) + \frac{1}{2}Tr\left[\Sigma_\nu\Sigma_F^{-1}\right]\right]^2\right) \\ &= \frac{1}{2}\,Tr\left[\left(\Sigma_\nu\Sigma_F^{-1}\right)^2\right] + Tr\left[\Sigma_\nu\cdot\Sigma_F^{-1}\right]. \end{aligned} \tag{2}$$

As alluded to earlier, the filtering equations and optimal gain are similar to the scalar case (see Table 5), where the measurement error covariance R is now given in the general multivariate case by the expression (2).

5.4 Algorithm and Implementation

Based on the description of the design, the optimal estimation algorithm can be summarized as follows:

- Estimate the additive imaging noise prior to the visual tracking process [65].
- For every pixel, run two filters to estimate the foreground and background likelihoods ($\hat{\rho}_F(r)$ and $\hat{\rho}_B(r)$):
    1. at the prediction step, run the corresponding equations in Table 5 to obtain the predictions.
    2. obtain a measurement by taking the classification likelihood given by Bayesian segmentation on the current image.
    3. at the update step, run the corresponding equations in Table 5 to obtain the updates.

- The estimated classification probability field is obtained by normalizing the likelihood estimates previously obtained: $\frac{\hat{\rho}_F}{\hat{\rho}_F+\hat{\rho}_B}$. The 50% contour of this probability field defines the bounding contour of the target.

The implementation of the algorithm follows the algorithmic steps just described. Given the typically small size of the target relative to the image dimensions, windowing can be used in order to speed up the technique. In that case, a localization procedure [5, 26, 91] should be applied prior to performing segmentation measurements. Such a localization procedure guarantees that the prediction and measurement are in consistent coordinate frames, i.e. prediction and measurement are aligned.

5.5 Complex Appearance Models and Multiple-Object Tracking

In computer vision, it is not uncommon to encounter complex appearance models that cannot be adequately represented by unimodal Gaussian distributions. The estimation technique presented here can be extended to handle such cases of complex object/background appearance models and also deal with multiple object tracking. We propose to represent such complex distributions with Gaussian mixture models. In the context of this work, such mixture models can be obtained by increasing the number of segmentation classes. The segmentation modeling then accounts for $m = m_F + m_B$ classes, where m_F and m_B denote the number of Gaussian components in the mixtures one would typically associate to the foreground and background in the standard additive model $\sum_{k=1}^{m} \mathcal{N}(\mu_k, \sigma_k^2)$. This approach to multimodal Gaussian modeling is similar to the fragment modeling presented in [16]. At each pixel, the m likelihoods are estimated using the optimal estimation scheme devised earlier. The likelihoods of classes forming the target are aggregated together to form the foreground likelihood and a similar operation is performed for the background classes. Subsequently, the classification probability map is generated and the 50% contour of the map is used to recover the bounding contour delineating the object. Furthermore, multiple object tracking can be achieved by initially defining collections of classes forming the different objects and aggregating those classes together once optimal estimation is completed.

5.6 Experiments and Results

This section describes the experiments used to test the validity of the estimator design. Manual segmentation is performed on the sequences of images to provide ground truth. To objectively assess the performance, the number of misclassified pixels (NMP) is used as a quantitative metric. A smoothness index is also used. At each frame, it consists of a dissimilarity measure between said frame and the previous frame. Hence, this metric provides a measure of the temporal consistency of the target and indicates how fast the shape variations are.

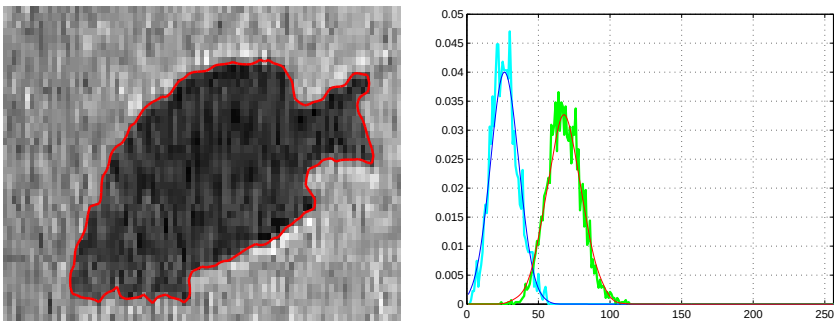
5.6.1 Modeling the Target and Background

Selection of the segmentation parameters is completed offline on the first frame of the sequence of images or using prior knowledge about the target appearance model. Figure 19(a-b) depicts a sample image and the corresponding target and background distributions in blue and red respectively. The true pdf's are given by thick lines while Gaussian-fitted pdf's are shown with fine lines. Figure 19(c-d) shows a color sample image and its target distribution modeled as a bimodal Gaussian mixture. Two ellipsoids represent the 3D unimodal Gaussian components of the mixture and the image sample points are superimposed on the distribution-fitted model. Target points are displayed in red and correspond to pixels inside the contour while background points are displayed in black and correspond to pixels outside the contour. These two examples demonstrate the ability of the Gaussian model to represent realistic distributions arising from natural image sequences. If the first frame of the sequence is available prior to visual tracking, approximate selection of target and background regions can be used in conjunction with standard parametric density estimation techniques [78] to determine the parameters of the Gaussian model.

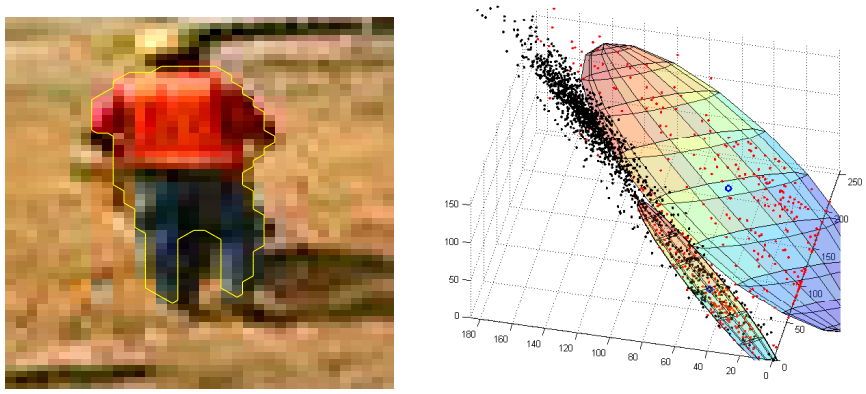
5.6.2 Optimality

5.6.2.1 Setup

This set of experiments is designed to verify the optimality of the gain. From an original high-SNR infra-red sequence of images, we generated multiple noise-corrupted grayscale and



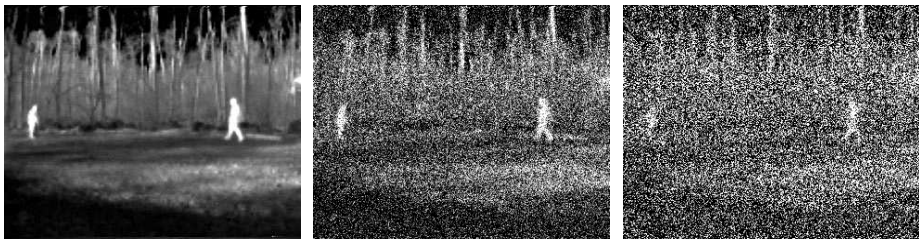
(a) Image sample (b) Target/Background pdf's



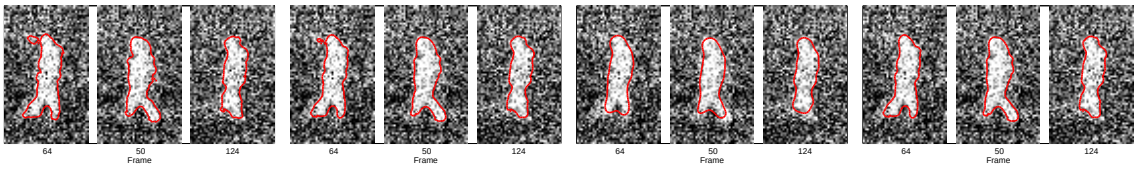
(c) Image sample (d) Target distribution

Figure 19: Modeling target/background with Gaussian pdf's.

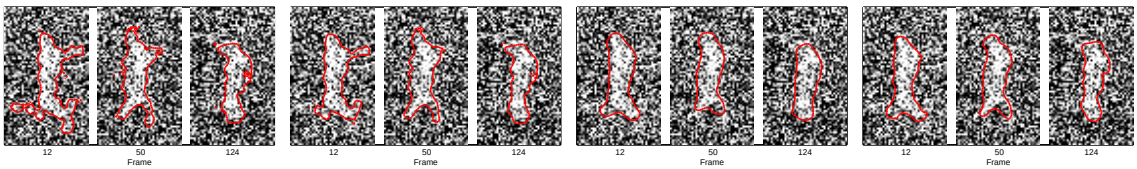
color sequences. The noise variance σ_ν of the additive imaging noise $\mathcal{N}(0, \sigma_\nu^2)$ is controlled to vary between 25 and 100 in the grayscale case. For the color sequence, the covariance matrix Σ_ν is taken diagonal ($\Sigma_\nu = \alpha \cdot \mathbb{1}$) where $\mathbb{1}$ is the identity matrix in three dimensions and α varies between 50 and 150. Figures 20(a-c) and 21(a-c) depict a sample image from the original sequence and the corrupted sample image at given noise levels. All sequences are then tracked using the filtering method presented earlier but with constant gains. Subsequently, the optimal filtering method is applied to the sequences. For these experiments and those following, the process noise covariance was chosen to be $Q = 0.3$ for the grayscale sequences and $Q = 0.9$ for the color sequences.



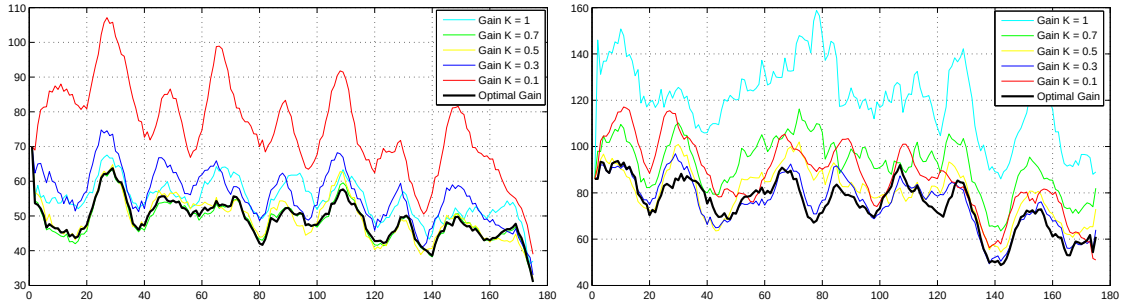
(a) Original IR sequence (b) Corrupted ($\sigma_\nu = 25$) (c) Corrupted ($\sigma_\nu = 100$)



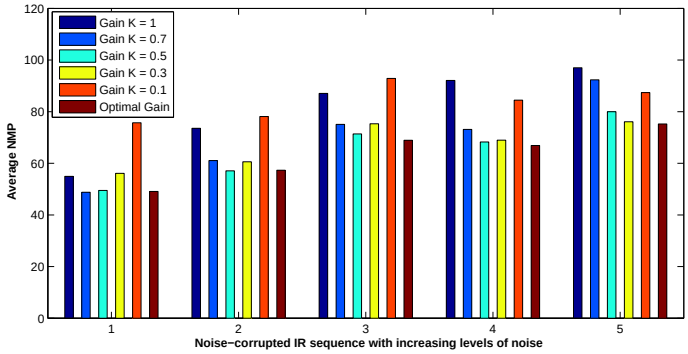
(d) $K = 1$, $\sigma_\nu = 25$ (e) $K = 0.7$, $\sigma_\nu = 25$ (f) $K = 0.1$, $\sigma_\nu = 25$ (g) K_{opt}, $\sigma_\nu = 25$



(h) $K = 1$, $\sigma_\nu = 100$ (i) $K = 0.7$, $\sigma_\nu = 100$ (j) $K = 0.1$, $\sigma_\nu = 100$ (k) K_{opt}, $\sigma_\nu = 100$

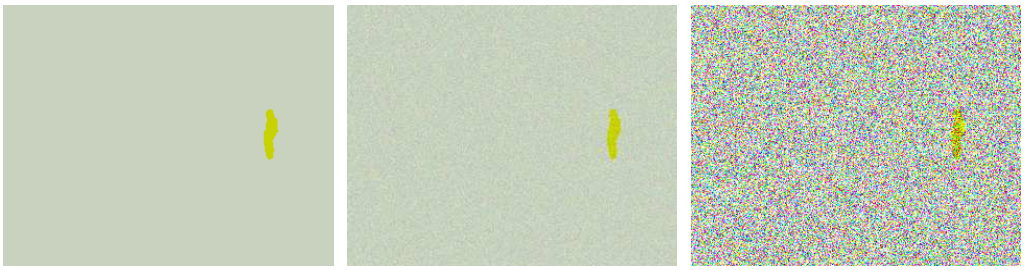


(l) NMP vs. time for a given noise level ($\sigma_\nu = 25$) (m) NMP vs. time for a given noise level ($\sigma_\nu = 100$)

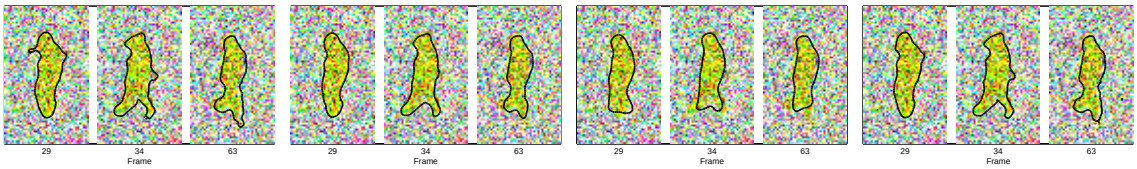


(n) Average NMP error vs. noise level

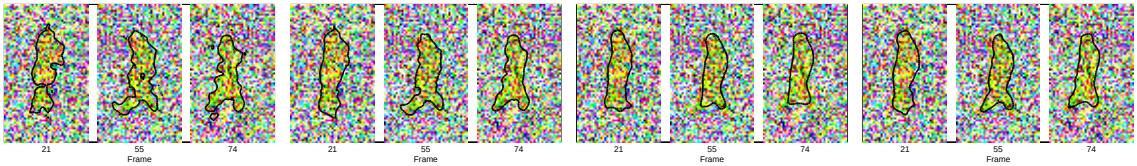
Figure 20: Quantitative assessment of optimality on grayscale sequences. From top to bottom: sample frames, sample estimates and NMP metric. The bottom figure displays the mean NMP for different noise levels.



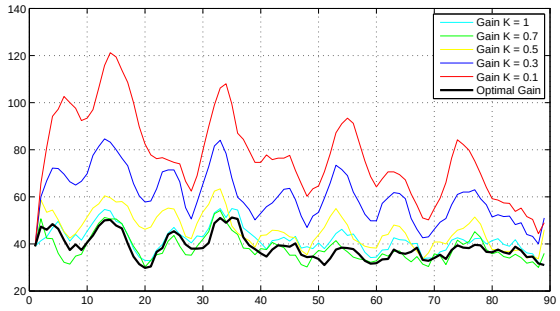
(a) Original color sequence (b) Corrupted ($\Sigma_\nu = 10 \cdot \mathbb{1}$) (c) Corrupted ($\Sigma_\nu = 100 \cdot \mathbb{1}$)



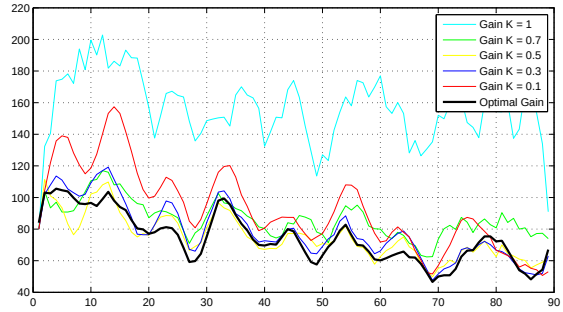
(d) $K = 1$, $\Sigma_\nu = 100 \cdot \mathbb{1}$ (e) $K = 0.7$, $\Sigma_\nu = 100 \cdot \mathbb{1}$ (f) $K = 0.1$, $\Sigma_\nu = 100 \cdot \mathbb{1}$ (g) K_{opt}, $\Sigma_\nu = 100 \cdot \mathbb{1}$



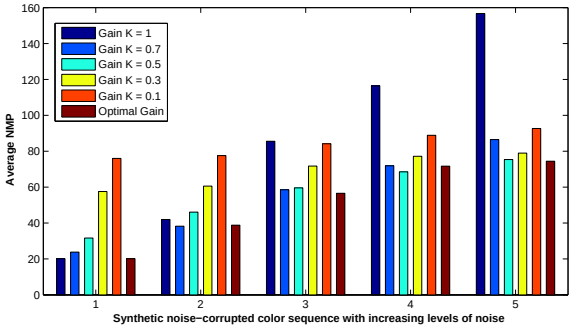
(h) $K = 1$, $\Sigma_\nu = 200 \cdot \mathbb{1}$ (i) $K = 0.7$, $\Sigma_\nu = 200 \cdot \mathbb{1}$ (j) $K = 0.1$, $\Sigma_\nu = 200 \cdot \mathbb{1}$ (k) K_{opt}, $\Sigma_\nu = 200 \cdot \mathbb{1}$



(l) NMP for a given noise level ($\Sigma_\nu = 100 \cdot \mathbb{1}$)



(m) NMP for a given noise level ($\Sigma_\nu = 200 \cdot \mathbb{1}$)



(n) Average NMP error vs. noise level

Figure 21: Quantitative assessment of optimality on color sequences. From top to bottom: sample frames, sample estimates and NMP metric. The bottom figure displays the mean NMP for different noise levels.

5.6.2.2 Results

The results are reported in Figures 20 and 21; they show that the best performance is indeed obtained when the optimal gain is used. As can be expected, in presence of severe noise, fixed low-gain filtering strategies have closer performance to the optimal estimator. In the case of small noise perturbations, the optimal estimator has performance close to high-gain filtering strategies.

5.6.3 Comparative Performance

5.6.3.1 Setup

Tracking experiments were also conducted to compare the performance of the estimator with other standard tracking techniques. We used the Bayesian segmentation [31] and the filtering method described in [36]. Additionally, for the first two grayscale sequences, an active contour tracking technique [70] was also applied. In the following, these methods are labelled respectively as Bayesian, Deformotion filter and AC. The gain parameters of the Deformotion filter were chosen to the best of our understanding, so as to yield the best results. For each test sequence, multiple runs with different values for the gain parameters were necessary; the set of parameters providing the most satisfactory results was retained. Similarly, the smoothing term of the active contour was chosen to provide the best segmentation possible. In these experiments, the Bayesian segmentations constitute the base measurements, i.e. the Deformotion filter and the optimal estimator share the same measurements provided by the Bayesian segmentation algorithm.

First, the tracking techniques were applied to one noise-corrupted IR sequence ($\sigma_\nu = 50$) and the results obtained were compared to the optimal estimator using the NMP. For this test sequence, the foreground and background were modelled using the respective distributions $\mathcal{N}(\mu_F, \sigma_F^2)$ and $\mathcal{N}(\mu_B, \sigma_B^2)$ with $(\mu_F, \sigma_F) = (202, 68)$ and $(\mu_B, \sigma_B) = (103, 85)$. The parameter σ_ν is equal to 50. A second experiment was conducted with a real-life aquarium sequence. Here again, Gaussian modelling was used but with the set of parameters $(\mu_F, \sigma_F) = (30, 14)$ and $(\mu_B, \sigma_B) = (68, 11)$. The parameter σ_ν was estimated to 25 prior to the visual tracking. A third experiment was conducted using a different real-life aquarium

sequence. The set of parameters $(\mu_F, \sigma_F) = (32, 13)$ and $(\mu_B, \sigma_B) = (80, 17)$ was used for the Gaussian modelling. The tracking techniques were applied to a noisy color sequence. The acquisition process, video compression and data transmission through a wireless channel explain the poor quality of this test sequence. In particular, the sequence of images suffers from severe noise corruption and interlacing and other compression artifacts. Here again, the target and background were modelled using Gaussian distributions with parameters

$$(\mu_F, \Sigma_F) = \left(\begin{bmatrix} 154 \\ 159 \\ 174 \end{bmatrix}, 10^4 \cdot \begin{bmatrix} 1.04 & 1.01 & 0.91 \\ 1.01 & 1.00 & 0.90 \\ 0.91 & 0.90 & 0.87 \end{bmatrix} \right)$$

$$(\mu_B, \Sigma_B) = \left(\begin{bmatrix} 113 \\ 127 \\ 133 \end{bmatrix}, 10^4 \cdot \begin{bmatrix} 0.49 & 0.47 & 0.5 \\ 0.47 & 0.54 & 0.51 \\ 0.5 & 0.51 & 0.74 \end{bmatrix} \right),$$

estimated from the first frame of the test sequence prior to visual tracking. The parameter Σ_ν was estimated to be

$$\Sigma_\nu = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}.$$

Lastly, a color sequence featuring a worker on a construction site was tested. Gaussian mixture models were used to model the target and background:

$$(\mu_{F_1}, \Sigma_{F_1}) = \left(\begin{bmatrix} 185 \\ 57 \\ 29 \end{bmatrix}, 10^3 \cdot \begin{bmatrix} 3.2 & 1.65 & 1.15 \\ 1.65 & 2.51 & 1.84 \\ 1.15 & 1.84 & 1.63 \end{bmatrix} \right),$$

$$(\mu_{F_2}, \Sigma_{F_2}) = \left(\begin{bmatrix} 40 \\ 46 \\ 41 \end{bmatrix}, 10^3 \cdot \begin{bmatrix} 0.78 & 0.68 & 0.52 \\ 0.68 & 0.7 & 0.64 \\ 0.52 & 0.64 & 0.87 \end{bmatrix} \right),$$

$$(\mu_{B_1}, \Sigma_{B_1}) = \left(\begin{bmatrix} 197 \\ 145 \\ 82 \end{bmatrix}, 10^3 \cdot \begin{bmatrix} 0.37 & 0.29 & 0.25 \\ 0.29 & 0.27 & 0.23 \\ 0.25 & 0.23 & 0.23 \end{bmatrix} \right),$$

$$(\mu_{B_2}, \Sigma_{B_2}) = \left(\begin{bmatrix} 159 \\ 127 \\ 78 \end{bmatrix}, 10^3 \cdot \begin{bmatrix} 2.83 & 2.34 & 2.27 \\ 2.34 & 2.11 & 2.07 \\ 2.27 & 2.07 & 2.23 \end{bmatrix} \right).$$

The noise parameter was also estimated:

$$\Sigma_\nu = \begin{bmatrix} 10^2 & 0 & 0 \\ 0 & 10^2 & 0 \\ 0 & 0 & 10^2 \end{bmatrix}.$$

5.6.3.2 Results

The results obtained are depicted in Figures 22, 23, 24 and 25. They clearly indicate that the optimal filtering strategy is a competitive tracking technique that is able to significantly mitigate the effects of imperfect contour measurements resulting from perturbations on the image data. Direct comparison to the Deformotion filter shows that the technique presented here has similar or better performance without the need for manual gain tuning. Moreover, Figure 23 shows that the technique is able to track elastic targets that drastically change through time without significant lag. While local shape perturbations are attenuated to maintain temporal consistency (see Figure 23-i), detrimental oversmoothing is also avoided.

5.6.4 Analysis

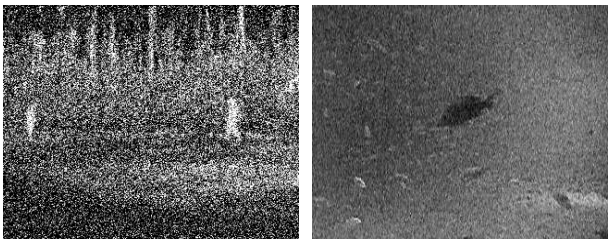
These experiments show that the visual contour tracking technique presented here constitutes a viable tracking technique when operating in an online, recursive setting. Contrary to many estimation-based tracking techniques that require manual specification of the gain parameter, we laid out a framework to determine the optimal gain parameter associated to the filtering process. In this framework, the optimal gain computation is formally tied to quantitative uncertainty levels on the image data, e.g. the noise variance, estimated prior to performing visual tracking. The first set of experiments validates the optimal gain derivation and the filtering strategy.

The last experiments use real-life sequences of images to further test the performance of the system and compare it against other tracking techniques. The first aspect to note

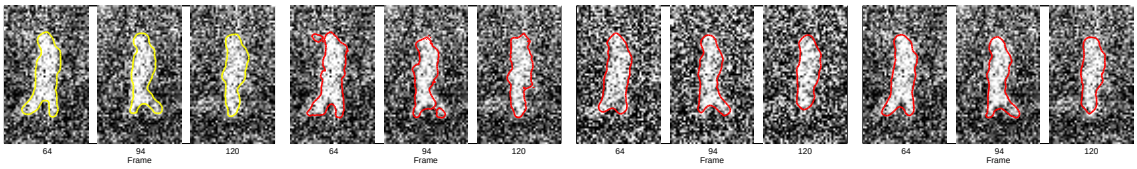
is that the optimal estimation technique provides good estimates in the presence of severe perturbations. Even when these perturbations diverge from the Gaussian noise model, the filter is able to estimate the target contour. Secondly, the estimation technique can be used to deal with imperfect measurements arising from imperfect segmentation models (see Figure 25). Thirdly, comparison to other tracking techniques shows that the optimal estimator is a competitive technique with equal or better performance.

On the computational aspect, the optimal estimation process introduced here is fast and very efficient. The operations required by the algorithm involve only scalar additions and multiplications. Our sub-optimal Matlab implementation can process about 5 frames per second. Optimization of the implementation should enable real-time tracking.

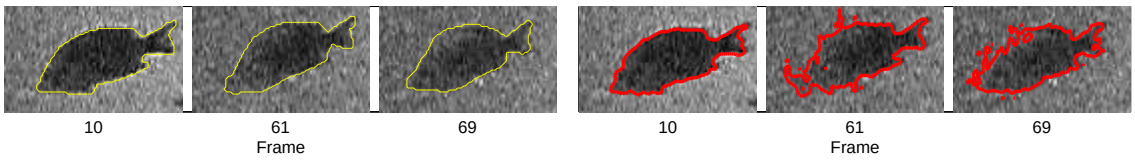
For these experiments, the class measurement error covariance R is taken constant ($R_t = R$) and is computed offline from the estimation of the noise variance on the first frame of the sequence. In the case of varying perturbations, an adaptative filter can be easily obtained by estimating online the noise variance and updating R_t at each frame.



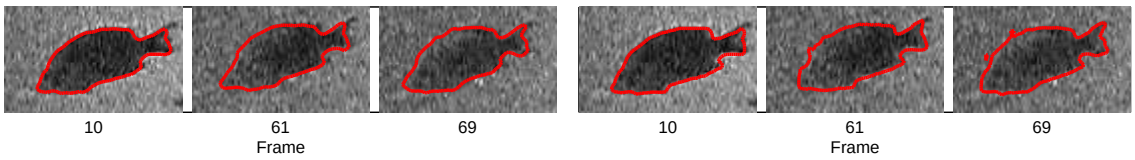
(a) IR sequence (b) Fish sequence



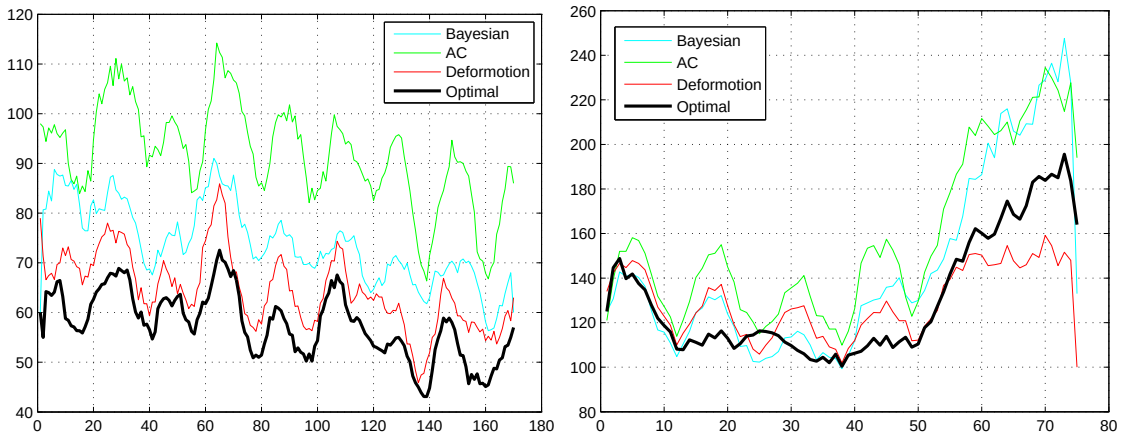
(c) Ground truth (d) Bayesian segmentation (e) Deformotion filter (f) Optimal estimator



(g) Ground truth (h) Bayesian segmentation



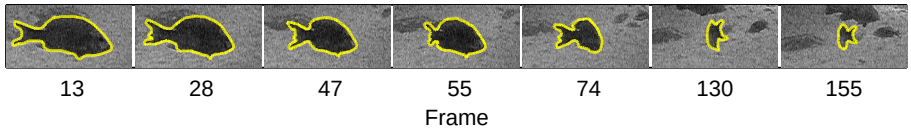
(i) Deformotion filter (with manual gain selection) (j) Optimal estimator



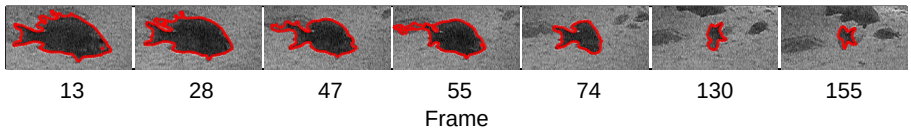
(k) NMP error vs. time for test algorithms on IR sequence (l) NMP error vs. time for test algorithms on fish sequence

Figure 22: Quantitative assessment of performance on grayscale sequences. On top, figures depict samples from the two test sequences. The next three rows show sample estimates obtained using the optimal filtering scheme and other tracking techniques (active contour estimates are similar to the Bayesian segmentation estimates). The bottom figures compare the performances of each technique using a quantitative metric (the number of misclassified pixels).

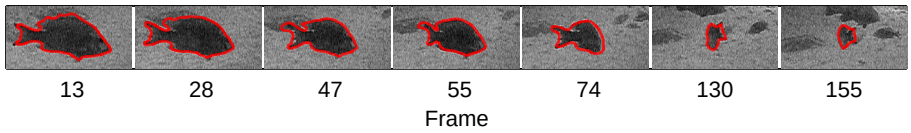
(a) Sample Frame 1 (b) Sample Frame 75 (c) Sample Frame 102



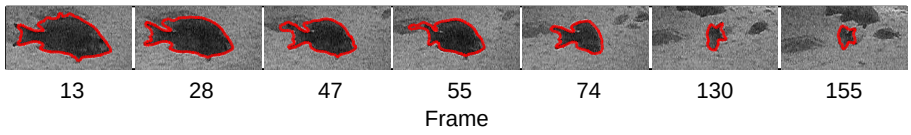
(d) Ground truth



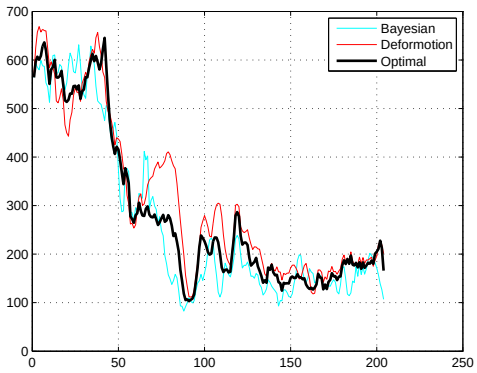
(e) Bayesian segmentation



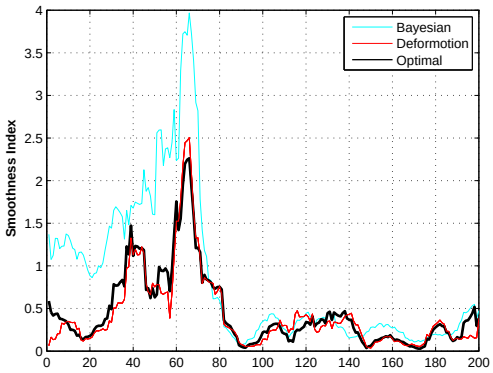
(f) Deformotion filter (with manual gain selection)



(g) Optimal estimator

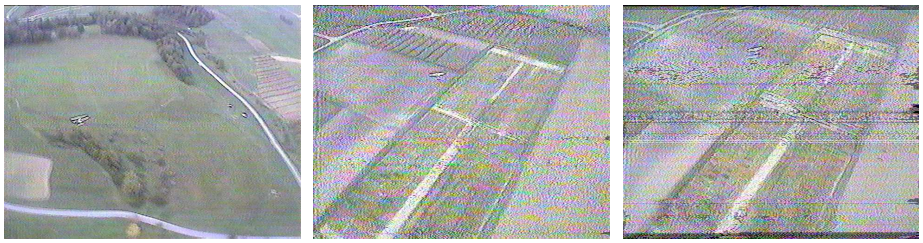


(h) NMP error vs. time

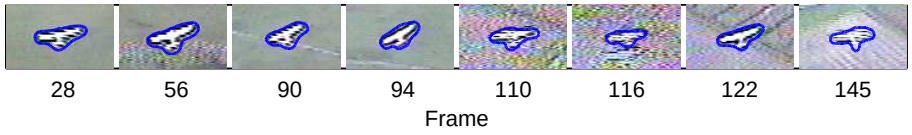


(i) Smoothness index

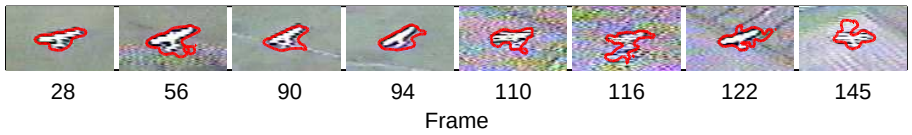
Figure 23: Quantitative assessment of performance on a highly deformable target. The top row depict sample frames from the test sequence. The next four rows show the ground truth and sample estimates obtained using the tracking techniques. The bottom left figure displays the number of misclassified pixels. The bottom right figure compares a smoothness index capturing the temporal consistency of the target.



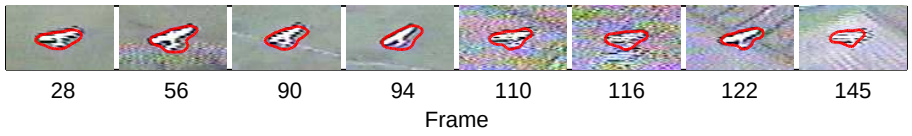
(a) Sample Frame 50 (b) Sample Frame 111 (c) Sample Frame 117



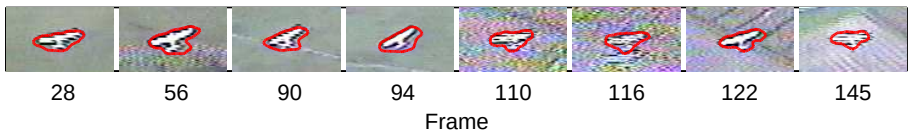
(d) Ground truth



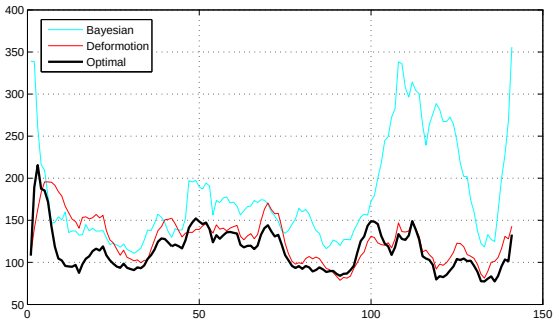
(e) Bayesian segmentation



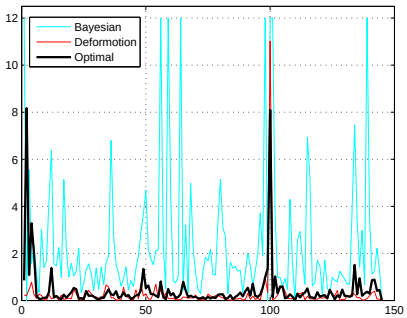
(f) Deformotion filter (with manual gain selection)



(g) Optimal estimator



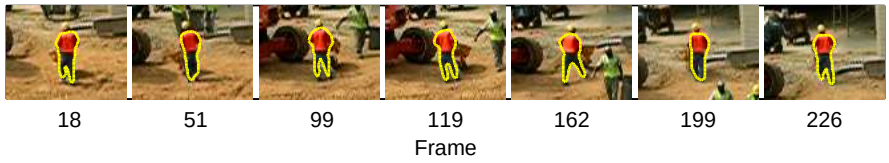
(h) NMP error vs. time for test algorithms on noisy color sequence



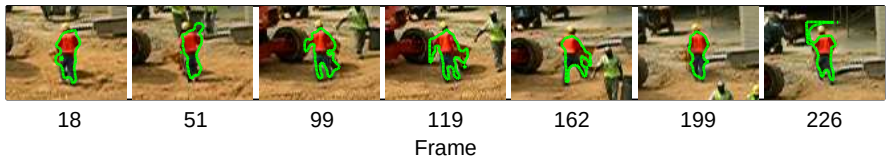
(i) Smoothness index

Figure 24: Quantitative assessment of performance on a color sequence. The top row depicts sample frames from the test sequence. The next four rows show sample estimates obtained using the different tracking techniques. The bottom figure displays the performance of each technique using the number of misclassified pixels.

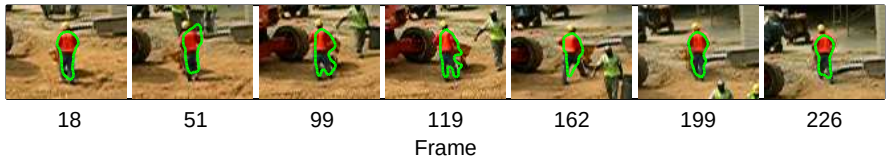
(a) Sample Frame 1 (b) Sample Frame 38 (c) Sample Frame 220



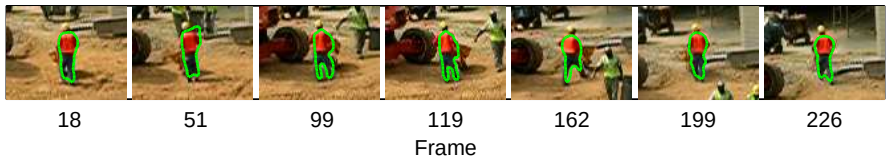
(d) Ground truth



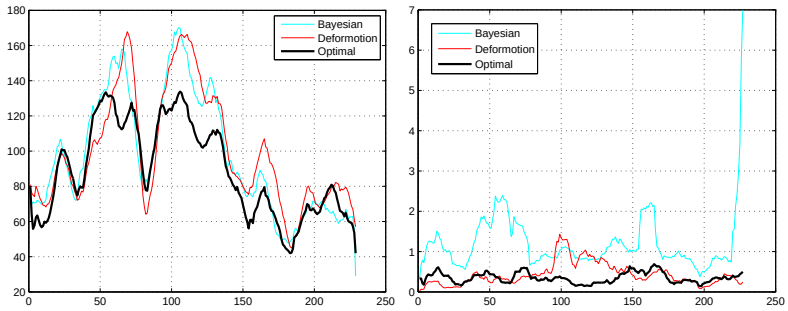
(e) Bayesian segmentation



(f) Deformotion filter (with manual gain selection)



(g) Optimal estimator



(h) NMP error vs. time (i) Smoothness index

Figure 25: Quantitative assessment of performance on a color sequence. The test sequence features non-stationary camera and multimodal target distribution. Sample estimates are displayed. The bottom figures show the number of misclassified pixels and a smoothness index capturing the temporal consistency of the target.

5.7 Conclusion

This chapter presented the derivation of an optimal estimator for online visual contour tracking. In contrast to the prevailing methods using a top-down approach, we employ a bottom-up approach starting from the measurement strategy. In this framework, filtering on the infinite-dimensional space of closed curves is reduced to a series of point-wise scalar filtering problems. The optimal gain derivation is formally tied to measurable uncertainty levels of the image data and, therefore, does not require manual gain tuning. The resulting optimal estimator is able to handle severe noise perturbations and compares favorably with other estimation-based tracking techniques.

CHAPTER VI

A DISTRIBUTED FILTERING TECHNIQUE FOR SPATIAL CONSISTENCY

6.1 Introduction

This chapter extends the optimal estimation technique derived earlier for the segmentation and tracking of moving objects in a temporal sequence of images. In the previous chapter, rather than using a top-down approach with an overall description of an estimator, we began with the segmentation measurement strategy, considered the effects of image perturbations on the measurements and further derived the estimator components. By considering the problem in this manner, the constitutive elements of the filter were customized. However, only temporal consistency of the target was achieved with the recursive filtering: an external post-processing smoothing procedure was necessary to obtain spatial consistency. Here, we consider a distributed filtering method to incorporate neighboring information when estimating the state of a given pixel. This results in a more natural way to obtain spatial consistency that fits elegantly into the estimation framework. The new estimation technique is then applied to recorded imagery and quantitatively compared against other visual tracking techniques in order to validate the filter design.

The chapter is organized as follows. In Section 6.2, we discuss the distributed filtering strategy that is considered in order to obtain spatial consistency. Experimental validation follows in Section 6.3. Section 6.4 concludes the chapter.

6.2 Distributed Filtering for Spatial Consistency

This section discusses the use of distributed filtering methods for maintaining spatial consistency when tracking a target in an image sequence using the probabilistic optimal estimator presented in chapter 5. While temporal consistency may be obtained by application of the estimation scheme with Bayesian segmentation as a measurement strategy, so far each pixel makes an individual decision as to whether it belongs to the foreground or background.

Note that in the literature [32], smoothing techniques such as lowpass Gaussian filtering (or similar isotropic diffusion) and anisotropic diffusion over the image domain are recommended in order to guarantee spatial continuity of the target. Following that rationale, the state estimate update obtained with the estimation scheme was post-processed in a similar fashion to provide a smooth target boundary in [55]. One alternative to such post-processing is to consider distributed filtering methods that would incorporate neighboring information when estimating the state of a given pixel.

6.2.1 A Standard Distributed Kalman Filtering Method

In this paragraph, we briefly describe the additional steps involved in a distributed algorithm for multisensor Kalman filtering introduced by Rao and Durrant-Whyte [66]. It consists first of performing the prediction and update equations used in the information form of the Kalman filter to obtain at each node i the so-called *local partial estimates* $\tilde{\mathbf{x}}_i(t|t)$ and $\tilde{P}_i(t|t)$. The nodes then communicate with each other and assimilate received information according to:

$$\hat{P}_i^{-1}(t|t) = \hat{P}_i^{-1}(t|t-1) + \sum_{j=1}^{m} \left(\tilde{P}_j^{-1}(t|t) - \hat{P}_j^{-1}(t|t-1) \right) \tag{3}$$

$$\hat{\mathbf{x}}_i(t|t) = \hat{P}_i(t|t) \cdot [\hat{P}_i^{-1}(t|t-1)\, \hat{\mathbf{x}}_i(t|t-1) + \sum_{j=1}^{m} \left(\tilde{P}_j^{-1}(t|t)\, \tilde{\mathbf{x}}_j(t|t) - \hat{P}_j^{-1}(t|t-1)\, \hat{\mathbf{x}}_j(t|t-1) \right)] \tag{4}$$

The terms under summation are the state error information and the variance error information. They are the two terms that need to be transmitted by each node to its m neighbors. The decentralised Kalman filter is mathematically equivalent to the centralised Kalman filter. For more details on this scheme, we refer to [66].

6.2.2 Application to Visual Tracking

The distributed Kalman filter detailed previously can be applied to the visual tracking problem. In the following, we consider that each pixel is spatially connected to its m closest neighbors, where m is a value to be specified later. In particular, we make the fundamental assumption that a given pixel and its m closest neighbors capture the same visual

phenemonon, only from different but close viewpoints. Therefore, the distributed Kalman filtering approach can be applied to deal with the estimation problem at hand. Similarly to the general approach presented previously, each node performs the filtering, then broadcasts the local filtered state to neighboring nodes. After it has received information from its neighbors, each node incorporates the received local state estimates in order to refine its own previous estimate. The communication scheme may forbid pixels to communicate accross the target contour, i.e. pixels that belong to the foreground may be forbidden to communicate with background pixels and vice-versa. Figure 26 gives a depiction of the network topology when the aforementioned communication is enforced.

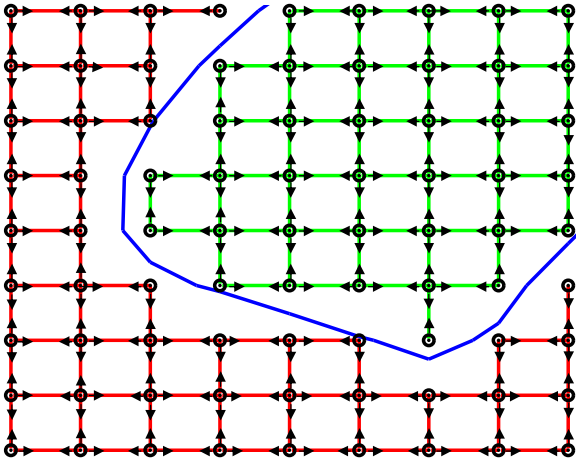


Figure 26: Network topology for distributed filtering (4-connectivity).

6.2.3 Implementation

First, the optimal filter is declined in its information form. Table 6 summarizes the filtering equations for the system when using the information form of the filter.

At each node i, predictions $\hat{\rho}_{t,i}^{-}$ and $\hat{P}_{t,i}^{-}$ are obtained by running the corresponding equations in Table 6. Once a measurement ζ_t becomes available, the update step described in Table 6 is carried out to arrive at local partial estimates $\tilde{\rho}_{t,i}$ and $\tilde{P}_{t,i}$. The node then broadcasts these partial estimates to its m closest neighbors. Similarly, it receives partial estimates from its neighbors and then assimilates such information by running the set of

Table 6: Filtering equations using the information form.

Prediction	$\begin{cases} \hat{\rho}_t^- = \hat{\rho}_{t-1}^+ \\ \hat{P}_t^- = \hat{P}_{t-1}^+ + Q \end{cases}$
Update	$\begin{cases} \left(\hat{P}_t^+\right)^{-1} = R^{-1} + \left(\hat{P}_t^-\right)^{-1} \\ K_t = \hat{P}_t^+ \cdot R^{-1} \\ \hat{\rho}_t^+ = \left(\hat{\rho}_t^-\right)^{1-K_t} \cdot (\zeta_t)^{K_t} \end{cases}$

equations:

$$\begin{cases} \left(\hat{P}_{t,i}^+\right)^{-1} = \left(\hat{P}_{t,i}^-\right)^{-1} + \sum_{j=1}^{m} \left[\left(\tilde{P}_{t,j}^-\right)^{-1} - \left(\hat{P}_{t,j}^-\right)^{-1}\right] \\ \hat{\rho}_{t,i}^+ = \left(\hat{\rho}_{t,i}^-\right)^{\hat{P}_{t,i}^+ \cdot \hat{P}_{t,i}^-} \cdot \prod_{j=1}^{m} \frac{\left(\tilde{\rho}_{t,j}\right)^{\hat{P}_{t,i}^+ \cdot \tilde{P}_{t,j}}}{\left(\hat{\rho}_{t,j}^-\right)^{\hat{P}_{t,i}^+ \cdot \hat{P}_{t,j}^-}} \end{cases} \tag{5}$$

The set of equations (4) is obtained directly from the original assimilation equations $(2-3)$ by taking into account that the linear filtering occurs in the log-space associated to the densities ($\mathbf{x} = log\,(\rho)$).

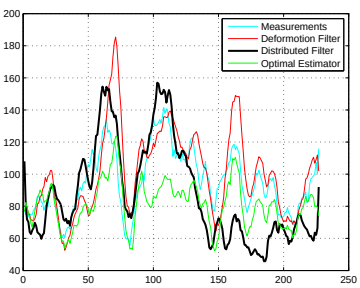
6.3 Experiments and Results

In order to validate the filtering design, we conducted experiments applying the resulting filter to a color and grayscale sequences.

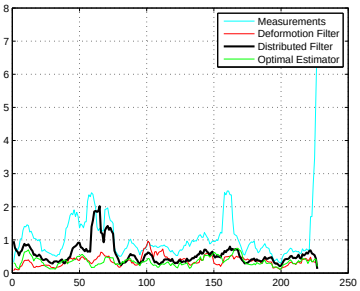
Base measurements are obtained by applying Bayesian segmentation [32] to the test sequences. Subsequently, the deformotion filtering technique [36], the optimal estimator derived in chapter 5 and the distributed filter are applied to the measurements. Ground truth was obtained via manual segmentation. To provide objective comparison, the number of misclassified pixels (wrt ground truth) is used as a quantitative error metric. It represents a measure of fidelity to the true curve being estimated. Additionally, the smoothness index defined in §5.6 is employed.

The results of the experiments are displayed in Figures 27-28. They show that the distributed filter is capable of attenuating contour perturbations arising from imaging noise and other visual disturbances. Spatial consistency manifests in the form of smooth target boundaries without the need of external ad hoc post-processing procedures.

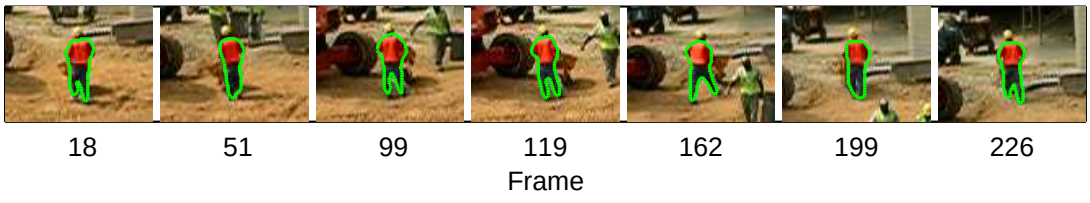
(a) Sample Frame (sequence 1)



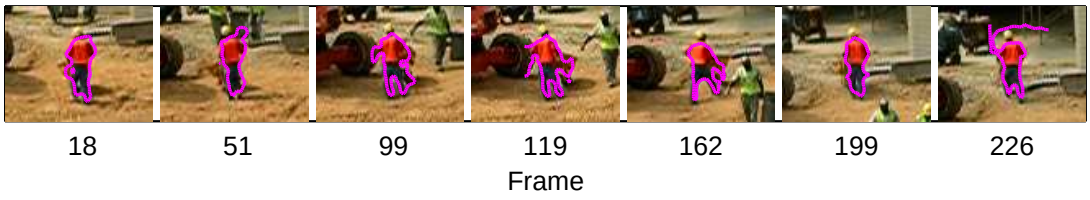
(b) NMP error vs. time



(c) Smoothness index



(d) Ground truth



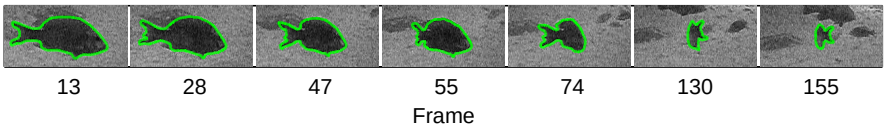
(e) Bayesian segmentation measurements

(f) Deformotion filter

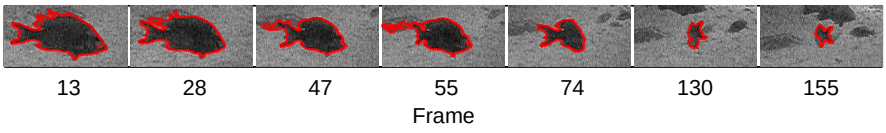
(g) Distributed filter

Figure 27: Quantitative comparison of performance for sequence 1. The top row depicts a sample frame, the number of misclassified pixels and a smoothness index capturing the temporal consistency of the target. Sample estimates are also displayed for each tracking technique.

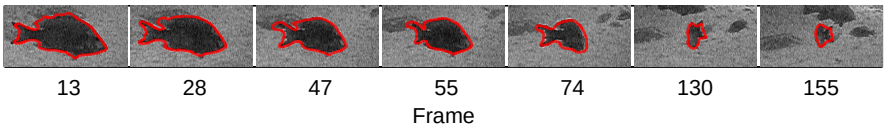
(a) Sample Frame (sequence 3)



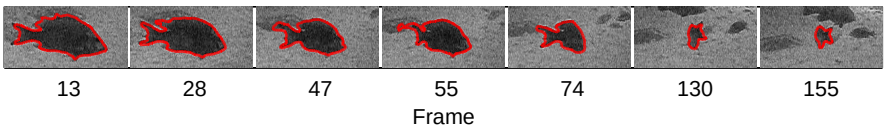
(b) Ground truth



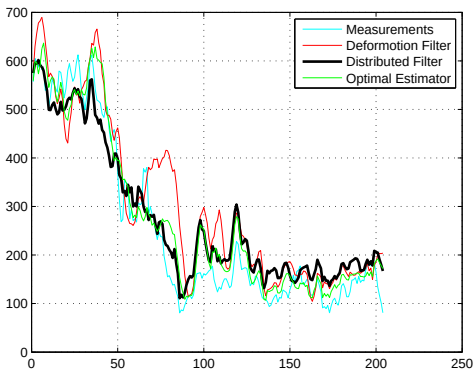
(c) Bayesian segmentation measurements



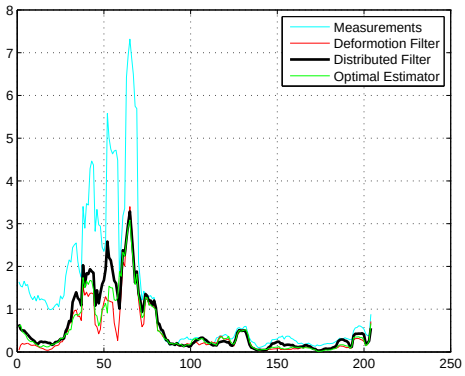
(d) Deformotion filter



(e) Distributed filter



(f) NMP error vs. time



(g) Smoothness index

Figure 28: Quantitative comparison of performance for sequence 3. Sample estimates are displayed for each tracking technique, along with the number of misclassified pixels and a smoothness index capturing the temporal consistency of the target.

6.4 Conclusion

This chapter presented an extension of the optimal estimator described in chapter 5. It discusses a distributed filter for online visual contour tracking. The consideration of a distributed filtering scheme provides spatial consistency of the target, which manifests in the form of smooth target boundaries without requiring post-processing or other regularizing procedures. Experiments conducted on recorded imagery validate the design and show that the filter's performance is comparable to the results obtained in chapter 5.

CHAPTER VII

LOCAL OPTIMAL FILTERS FOR DYNAMIC CURVE ESTIMATION

This chapter considers the task of closed curve filtering. Estimation theory is applied to solve the problem of tracking deformable moving objects in an image sequence. Segmentation-based visual tracking strategies provide the closed curve measurements to filter. We discuss the derivation of a local, linear description for planar curve variation and curve uncertainty. It consists of a family of non-intersecting trajectories transverse to a given curve. Along one of the single-dimensional transverse trajectories, linear curve operations are feasible. Using the linear operation, simple locally optimal filtering procedures are derived. In particular, it is shown that an optimal first-order filtering strategy can be rigorously obtained. Extending further the work, we derive sub-optimal second-order curve filtering strategies. The second-order models account naturally for the curve velocities, which results in better curve estimates when dealing with highly elastic objects. In contrast to the first-order model, the second-order curve dynamics are nonlinear, which motivates a linear discrete approximation prior to deriving an extended Kalman filtering approach. Once the curve filtering equations are derived, they are placed within the greater context of observer design for the estimation of a curve's position and deformations as it evolves in the plane. Application to online visual tracking is emphasized through experimentation with recorded imagery and objective comparison to other tracking methods.

7.1 Introduction

Segmentation-based tracking algorithms in computer vision involve the tracking of a target boundary. Due to overlapping clutter on the image data, poor resolution, imaging noise, compression artifacts or imperfect segmentation models, the bounding contour may be incorrect and corrupted. Over time, poor measurements lead to loss of track. Consequently, information gathered from past detections has been introduced in a variety of manners to increase robustness. Related literature includes methods exploiting temporal consistency of

the target [58], methods processing volumetrically the image sequence with batch techniques [60], and methods introducing strong shape priors into the detection [19].

7.1.1 Prior Related Work

Filtering of the segmenting contour provides another means to address the problem [6]. The references [9, 23, 36, 50, 55, 59, 77, 83, 84] provide such contributions for tracking in the controlled active vision literature. This chapter reports progress towards strategies for optimally filtering planar, smooth, closed curves, given a temporal sequence of such curves as generated by a segmentation algorithm. Due to the infinite-dimensional nature of the space of smooth, closed curves, this is but one potential approach [48]. The contents advance recent work in this area [59] and integrates related work [36].

Currently, most of the recent filtering designs proposed for tracking consider a fixed gain, manually specified. High gains are chosen when the image sequences are free of visual disturbances or the segmentation measurement model easily distinguishes target from background. When the image sequence is presumed to feature severe perturbations, fixed low gains are then adopted to moderate the injection of the measurements into the state estimate. While the gains do relate to qualitatively observable aspects of the image sequence, they have not been formally tied to quantifiable aspects of the image sequence, such as uncertainty of the image data, of the segmentation, or of the segmentation models. Particle filters [69, 83] do not require gain selection, however the need to maintain a density estimate in the shape space introduces a substantial computational burden. Low dimensional parametrized approximations [67] are then often used to minimize the number of particles used, exposing corresponding techniques [68] to the well-known out-of-sample problem.

7.1.2 Contributions

The principal contribution of the chapter is the derivation of a mechanism for estimating the optimal gain associated to the curve filtering process for planar curves, given quantitative uncertainty levels. To achieve this, we define a family of transverse curves to the curve in question. The family of transverse curves forms a collection of coordinate frames within which curve operations are linear. Linearity is only valid within a neighborhood of the curve

in question and is exploited to derive a locally optimal first-order curve filtering. Following, we present a second-order extended Kalman curve filtering strategy. The second-order model accounts naturally for the curve velocities and couples the update of the curve's position and velocity, which results in better curve estimates. The nonlinearity of the second-order curve dynamics prevents the derivation of a Kalman curve filtering strategy. Instead, we propose sensible linear discrete approximations that enable an appropriate extended Kalman filtering approach. The filtering is subsequently utilized to estimate the position and deformations of a curve as it evolves in the plane. The observer states include the rigid pose and the deforming curve states. The correction of the two states is decoupled. Specifically, the curve filtering is utilized in conjunction with standard filtering strategies on the (object) pose to estimate the position, shape and local shape deformations of a deformable moving object in an image sequence. The work can be extended to the 3D case involving surfaces.

7.1.3 Organization

The chapter is organized as follows. Section 7.2 presents the state-space representation used. In Section 7.3, we describe the transverse coordinate system for closed curves. Following, the formulation of an optimal first-order filtering strategy is reported in Section 7.4. Section 7.5 discusses the derivation of the sub-optimal second-order filtering methods. Experimental validation is discussed in Section 7.6. Section 7.7 concludes the chapter.

7.2 State-Space Representation

Under *deformotion* [90], a deformable moving object in a video sequence can be described by a group motion and a shape deformation. Local shape deformations serve to describe the non-rigid curve motion. Naturally then, the state description will include a rigid group component and a shape component. The shape description consists of the planar curve, denoted $\mathcal{C} : S^1 \to \mathbb{R}^2$. For the second-order system, the shape description will be augmented with the curve's normal velocity $\beta : S^1 \to \mathbb{R}$. In what follows, the curve $\mathcal{C}$ is implicitly

described as the zero level set of a higher-dimensional signed distance function:

$$\mathcal{C} = \{x \in D \mid \Psi(x) = 0\},$$

where $D \subset \mathbb{R}^2$ is a compact domain of the plane. When this is the case, the normal velocity will be extended over the level set domain to define $\bar{\beta} : \mathbb{R}^2 \to \mathbb{R}$ such that $\bar{\beta} \circ \mathcal{C} = \beta$. For more details on this setup and other velocities extension schemes, we refer to [57].

The group motion represents the pose and the pose velocities of the object in a reference coordinate frame; here it is denoted g and ξ, respectively, and will be the special Euclidean group $SE2$, or its subgroup $E2$, and its Lie algebra. Since the group variable and the Lie algebra live in a finite-dimensional space, standard filtering strategies [38, 39, 72] on finite-dimensional spaces can be applied to the pose and its velocity. This work focuses on the curve filtering process.

7.3 Transverse Curve Coordinates

In [59], a curve filtering strategy was proposed relying on curve correspondences derived from a combined Laplace-Poisson equation approach. Given two smooth curves, a scalar function defined on the curve domain was derived. The scalar function generated an associated error vector field. In this section, we describe how the setup is modified to obtain a characteristic vector field (which agrees with the error vector field modulo sign). The characteristic vector field will define the family of 1D transverse coordinate systems for a given smooth curve local to the two smooth curves.

7.3.1 The Characteristic Vector Field

The error vector field associated to two smooth curves $\mathcal{C}_0$ and $\mathcal{C}_1$ and their local region is obtained from the solution to a Laplace-Poisson partial differential equation (PDE), for which the induced flow is a diffeomorphism. The variational formulation of the Laplace-Poisson problem is

$$\min_u \int \|\nabla u\|^2 \, d\Omega, \tag{6}$$

such that $trace(\mathcal{C}_0) = u^{-1}(0)$ and $trace(\mathcal{C}_1) = u^{-1}(1)$. Its solution requires careful construction of the interior and boundary conditions. The source curve and the target curve define

the following solution domain decomposition of the total space Ω, $R := \overline{int(\mathcal{C}_0) \ominus int(\mathcal{C}_1)}$, $R_{pi} := int(\mathcal{C}_0) \cap int(\mathcal{C}_1)$, and $R_{lo} := \Omega \setminus \left(\overline{R \cup R_{pi}}\right)$, where $int(\mathcal{C})$ denotes the interior of the curve $\mathcal{C}$ and $\ominus$ is the set-symmetric difference; see Figure 29(a) for a depiction of the domains.

Instead of the defined boundary conditions above, set the boundary conditions to 0 for the interior curve portions $(\partial\overline{R_{pi}} \setminus (\mathcal{C}_0 \cap \mathcal{C}_1))$ and to 1 for the exterior curve portions $(\partial\,(R \cup R_{pi}))$. The exterior and the interior curve parts may comprise of subsets of $\mathcal{C}_0$ *and* $\mathcal{C}_1$ if $\mathcal{C}_0$ and $\mathcal{C}_1$ intersect. Via the calculus of variations, a solution to (6) in the domain enclosed by the source and target curves *with modified boundary conditions* satisfies

$$\Delta u_s(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in R, \tag{7}$$

where $\Delta = \nabla^2$, with the boundary conditions

$$\begin{aligned} u_s(\boldsymbol{x}) &= 0,\ \boldsymbol{x} \in \partial\overline{R_{pi}} \setminus (\mathcal{C}_0 \cap \mathcal{C}_1)\,, \\ u_s(\boldsymbol{x}) &= 1,\ \boldsymbol{x} \in \partial\,(R \cup R_{pi})\,, \end{aligned} \tag{8}$$

which is a simple reformulation of the minimization problem (6) based on the domain decomposition depicted in Figure 29. On R_{pi}:

$$\Delta u_{pi}(\boldsymbol{x}) = c \quad \boldsymbol{x} \in \overline{R_{pi}},\ c > 0 \tag{9}$$

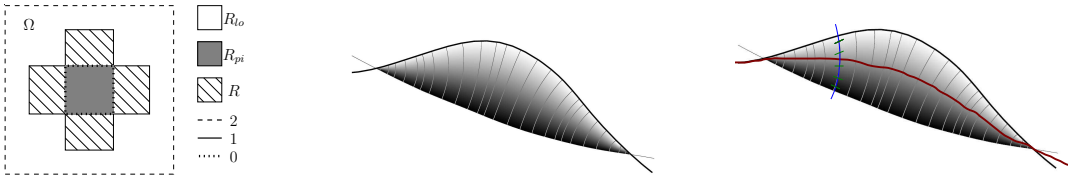
$$\Delta u_{lo}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \overline{R_{lo}}, \tag{10}$$

with boundary conditions

$$\begin{aligned} &u_{pi}(\boldsymbol{x}) = 0,\ \boldsymbol{x} \in \partial\overline{R_{pi}}, \\ &u_{lo}(\boldsymbol{x}) = 1,\ \boldsymbol{x} \in \partial\,(R \cup R_{pi})\,, \\ &u_{lo}(\boldsymbol{x}) = 2,\ \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{11}$$

The combined solution

$$u(\boldsymbol{x}) = \begin{cases} u_{lo}(\boldsymbol{x}), & \boldsymbol{x} \in R_{lo}, \\ u_{pi}(\boldsymbol{x}), & \boldsymbol{x} \in R_{pi}, \\ u_s(\boldsymbol{x}), & \boldsymbol{x} \in R \end{cases} \tag{12}$$



(a) Solution domain decomposition.

(b) Distance map using X_{char}.

(c) Transverse coordinates.

Figure 29: The topology and geometry of curve comparison and transverse curve coordinates.

defines the characteristic vector field X_{char} on Ω,

$$X_{char}(\boldsymbol{x}) := \begin{cases} \nabla u/\|\nabla u\|, & \boldsymbol{x} \in R, \\ \nabla u_o/\|\nabla u_o\|, & \boldsymbol{x} \in R_{lo}, \\ \nabla u_i/\|\nabla u_i\|, & \boldsymbol{x} \in R_{pi} \end{cases} \tag{13}$$

via the normalized gradient. Notice that the error vector field computed this way exhibits some interesting properties. In particular, its characteristic curves are non-intersecting, they approach the curves $\mathcal{C}_0$ and $\mathcal{C}_1$ from normal directions and their speed never vanishes. Figure 29(b) depicts the distance map generated by the characteristic vector field arising from two curves. Following the distance characteristics forwards and backwards starting at a curve point defines the local transverse coordinate system. Figure 29(c) depicts the coordinate system along one of the transverse curves (where $\mathcal{C}$ is in red).

The described scheme is fast and parametrization-free. It is not invariant to translations, rotations or scale. Therefore, we will assume that the two curves are or have been registered with respect to rotation and translation.

7.3.2 The Family of 1D Transverse Coordinate Frames.

A family of transverse 1D curves will be defined for a closed curve $\mathcal{C}$ that lies within a local region of the curve $\mathcal{C}_0$ and $\mathcal{C}_1$ (it may well be equal to $\mathcal{C}_0$ or $\mathcal{C}_1$). Given the field X_{char} and a particle $\boldsymbol{x}_0 \in \mathcal{C}$, its traveling distance, d, at position $\boldsymbol{x}$ along the characteristic of X_{char} through $\boldsymbol{x_0}$ is defined as the arc-length of the characteristic curve connecting $\boldsymbol{x}_0$ and $\boldsymbol{x}$. To measure these traveling distances from a complete set of initial locations, as specified by

$d^{-1}(\cdot, 0)$, solve

$$\begin{cases} d(\cdot, 0) = 0, \\ d_\tau + X_{char}^T \nabla_x d = 1, \end{cases} \tag{14}$$

where τ is an artificial time parameter for the PDE equation and $d : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$. From (13), X_{char} has unit norm.

For a given curve $\mathcal{C}$, the travelling distance map and its characteristics define the 1D family of transverse coordinate frames. Let s be the arc-length parameter of the curves in question, and assume that $\mathcal{C}_0(0)$, $\mathcal{C}_1(0)$ and $\mathcal{C}(0)$ lie on the same characteristic. Then at $\mathcal{C}(s)$ the associated transverse curve intersects, for example, the curves $\mathcal{C}_0$ and $\mathcal{C}_1$, at the same arc-length parameter s. The coordinate location on the transverse curve of the curve point $\mathcal{C}_0(s)$ is given by

$$sd(\mathcal{C}_0(s); \mathcal{C}(s)) = d(\mathcal{C}_0(s), 0), \quad \text{where } \mathcal{C} = d^{-1}(\cdot, 0),$$

where $sd(p; \mathcal{C})$ describes the signed travelling distance from a point $\boldsymbol{x}$ to its corresponding point on $\mathcal{C}$. The distance is negative when $\boldsymbol{x}$ lies interior to $\mathcal{C}$ and positive exterior to $\mathcal{C}$.

To be more concrete about the fact that these transverse curves have their own coordinates, we will specify the coordinates on the transverse line of a particular curve point $\hat{\mathcal{C}}(s)$ by $\hat{x}(s)$, whose value is given by $\hat{x}(s) = sd(\hat{\mathcal{C}}(s); \mathcal{C}(s))$. We will call this the *point notation* for a curve. The point notation will be used to define linear operations on curves that are locally close, and to define curve variances.

7.4 First-Order Curve Filtering for Visual Tracking

This section describes the derivation of a first-order curve filtering scheme on the space of smooth, closed curves and its application to the visual tracking problem.

7.4.1 Optimal First-Order Curve Filtering

The goal of the filtering process is to arrive at estimates of the curve state $\hat{\mathcal{C}}$. The filtering scheme itself relies on three fundamental assumptions:

Assumption 1 *The curves have been approximately registered.*

Assumption 2 *Given the level of uncertainty about the closed curve, the error is local. In particular, local curve metrics suffice to quantify the error.*

Assumption 3 *Curve variance and uncertainty is given by a smooth scalar function along the curve.*

These assumptions exist to make the curve filter feasible in practice. The rest of this section details the consequences of the assumptions.

Because the contents of this section will ultimately serve to temporally filter a curve, we will utilize conventional estimation notation (involving hats). Consider two curves, $\hat{\mathcal{C}}$ and $\hat{\mathcal{C}}^-$, which are both estimates for the true curve $\mathcal{C}$. Further, consider an additional curve $\mathcal{C}^m$ representing a noisy measurement of the true curve $\mathcal{C}$.

The curve error between $\hat{\mathcal{C}}$ and $\mathcal{C}$, and $\hat{\mathcal{C}}^-$ and $\mathcal{C}$ can be described using the distance function derived from X_{char}. We will use the curve $\mathcal{C}$ as the zero curve, i.e. $\mathcal{C} = d^{-1}(\cdot; 0)$. In point notation, the curve errors of $\hat{\mathcal{C}}^-$ and $\hat{\mathcal{C}}$ with respect to the curve $\mathcal{C}(s)$ are given by

$$\hat{e}^-(s) = \hat{x}^-(s) - x(s), \text{ and}$$
$$\hat{e}(s) = \hat{x}(s) - x(s)).$$

The variance associated with the error is

$$P^-(s) = E\left(\left(\hat{x}^-(s) - x(s)\right)^2\right) > 0, \text{ and}$$
$$P(s) = E\left(\left(\hat{x}(s) - x(s)\right)^2\right) > 0.$$

Furthermore, presume that the measurement error variance, computed in a similar fashion is $R(s) > 0$ and varies smoothly with s, and that the measurements are independent of $\hat{C}^-$, $\mathrm{Cov}(\mathcal{C}^{\mathrm{m}}, \hat{\mathcal{C}}^-) = 0$.

What we seek is an optimal selection of K so that the covariance $P(s)$ is minimized under the update

$$\hat{x}(s) = \hat{x}^-(s) + K\left(\hat{x}^m(s) - \hat{x}^-(s)\right), \tag{15}$$

given in point notation,

Note that this setup essentially reduces the problem of finding the optimal selection of K to a one-dimensional problem. In particular, it reduces to the Kalman gain for a 1D

system [10]. Thus the optimal choice of K is

$$K = P^- \left(P^- + R\right)^{-1}, \tag{16}$$

and the associated error variance is

$$P^+ = P^- \left(1 - K\right). \tag{17}$$

Since all quantities are smooth and bounded away from zero as needed for denominators, the above operations are smooth. Therefore, the resulting curve will be smooth.

7.4.2 Extension to Visual Tracking

The derived filtering strategy is applied to visual tracking in this subsection. Contour-based visual tracking will require the curve to be locally registered from frame to frame, therefore we will consider the problem to be one of identifying both the rigid motion and the non-rigid motion associated to the track target [90]. As described earlier, rigid motion can be given by the Euclidean linear group, $E(2)$, or the special Euclidean group, $SE(2)$, depending on one's needs. Non-rigid motion will be determined by a curve and its normal motion.

Combining the curve filtering strategy discussed previously, with a group filtering strategy, plus a dynamical model for the system will result in an observer for visual tracking. The observer components are described below in the ordering: prediction model, sensor measurement model, system measurement model, and correction. The observer is continuous-discrete, i.e. with continuous time dynamics and discrete time measurements.

7.4.2.1 Prediction Model

The prediction model describes the target's evolution. For the rigid group states, we presume a constant velocity model,

$$\dot{g} = g \cdot \xi, \qquad \dot{\xi} = 0.$$

For the contour, a first-order model is assumed here. Therefore, a static prediction model will be used ($\mathcal{C}_t = 0$). This model is appropriate for rigid body objects, almost rigid objects, and slowly varying objects (relative to the measurement rate). The curve covariance

increases by $Q(s)$, a smooth scalar function. Further the noise process leading to $Q(s)$ is independent of the curve $\mathcal{C}(s)$ and the measurement $\mathcal{C}^m$.

The prediction will generate $(\hat{g}^-, \hat{\xi}^-, \hat{\mathcal{C}}^-)$, in addition to propagating forward covariances associated to the group variable.

7.4.2.2 Sensor Measurement

Measurement of the target can be achieved through any segmentation algorithm applied to the current image. The segmentation can then be converted into a curve description if it is not already of that form. Candidate algorithms include active contours [6, 70], Bayesian segmentation [32], graph cuts [43], etc.

In a classical observer the measurements would be completely independent of the observer states, however image analysis of video has the nature of not explicitly providing the necessary signal. Instead it must be extracted from the image using an image processing or computer vision algorithm. The measurement procedure may not completely determine the necessary target state measurements (due to non-uniqueness of the group + shape decomposition). Consequently, a registration step is required to describe the predicted and measured shape with respect to the same coordinate frame.

Once segmentation and localization are performed on the current image, a registration procedure is applied to match the measured contour with the predicted contour, yielding a measurement g^m for the group motion and the measurement $\mathcal{C}^m$ for the shape. In practice, the group velocity ξ is not directly measurable.

Curve measurements $\mathcal{C}^m$ (or equivalently Ψ^m) need to be converted into the transverse coordinate system. The first step is to generate the characteristic vector field X_{char} arising from the predicted and measured curves by following the procedure delineated in Section 7.3. While the actual curve state was used to define the origin of the transverse curve coordinates in the previous sections, any curve within the local neighborhood can be used as the reference curve for the transverse coordinate frames. Invariance to the reference curve is due to the fact that the filtering equations rely on differences and that the characteristic vector field is independent of the chosen curve. Here we use the predicted curve and thus

the zero coordinate refers to the predicted curve $\hat{\mathcal{C}}^-$, thereby yielding the local distance to the measurements x^m.

7.4.2.3 Model Measurement

The model measurement is obtained by extracting the components of the internal state model that are equivalent to those obtained from the sensor measurement.

7.4.2.4 Correction

From a prediction on the state and a measurement on the observable part of the state, the correction step updates the estimate of the observer internal state.

Group correction Since the group motion is either a translation or a composition of a rotation and translation, it can be represented by a three-state vector. Depending on the prediction motion model chosen (linear vs. nonlinear), finite-dimensional filtering such as Kalman, extended or unscented Kalman are appropriate to use in order to correct the measured group. As a case in point, EKF can be applied to filter the nonlinear constant group velocity model provided earlier.

Shape correction Correction on the shape requires prior construction of the transverse coordinates. The origin of the coordinate system is placed at the predicted curve. The correction applied to the curve, in point notation, is then

$$\hat{x}^+(s) = \hat{x}^-(s) + K(x^m(s) - \hat{x}^-) = Kx^m(s),$$

since the predicted curve is at the origin. The curve state $\hat{\mathcal{C}}^+(s)$ is uniquely defined by $\hat{x}^+(s)$. The gain chosen is the optimal gain selection from Equation (16), and the covariance update is as per Equation (17).

7.5 Second-Order Curve Filtering

The major limitation of the first-order filtering technique is the absence of a dynamical prediction model: the shape dynamics are not accounted for. This is appropriate when dealing with rigid-body objects, quasi-rigid objects or when the rate of change relative to the

measurement rate is low. However, there is an inability of the model to predict accurately in situations where the target object undergoes fast or dramatic changes in shape due to the object's intrinsic shape elasticity or because of low frame rates at video acquisition. This section extends the previous work to derive sub-optimal second-order curve filtering strategies for applications to visual tracking. The curve position x and normal velocity[1] v are aggregated together to form the second-order curve state $\mathbf{x}(s) = [x(s), v(s)]$. In addition to the curve state, the filter state is now composed of the curve covariance matrix $\mathbb{P} : S^1 \to \mathbb{R}^{2x2}$, which is a measure of the curve uncertainty transverse to the curve and the normal curve velocity uncertainty. Due to the infinite-dimensional nature of curve geometry, some technical care is necessary. We retain the filtering assumptions made in §7.4. In particular, it is expected that all curves are aligned or have been registered prior to filtering. The curve estimate error is such that local curve metrics sufficiently quantify the error. Further, the initial measurement covariance $\mathbb{R}$ and initial curve uncertainty $\mathbb{P}$ are presumed to vary smoothly along the curve. Following, the correction gain resulting from the filtering strategy will also be a slowly varying function along the curve. Since the curve filtering process is reduced to filterings of curve particles along the single-dimensional transverse curves, these smoothness, locality, and constancy assumptions guarantee that the estimated state can be smoothly reconstructed.

7.5.1 Dynamical Prediction Models

The dynamical prediction models produce an estimate of the filter state at a future time from past estimates. Usually, they rely on prior knowledge about the state evolution resulting in phenomenological motion models. Here, the prediction models do not assume such priors. First, we propose a constant velocity model:

$$\begin{aligned} \hat{\mathcal{C}}_t &= \beta \mathcal{N}, \\ \hat{\beta}_t &= 0, \end{aligned} \tag{18}$$

[1]When using level set representations for the curve evolution, we use the more common notation β for the normal velocity.

where $\mathcal{N}$ is the curve normal. Rewritten in the level set framework, the curve evolution is described by the set of equations:

$$\begin{aligned} \hat{\Psi}_t &= \hat{\bar{\beta}} \cdot \left|\left|\nabla \hat{\Psi}\right|\right|, \\ \hat{\bar{\beta}}_t &= 0, \end{aligned} \tag{19}$$

Alternatively, a general purpose second-order model can also be used [59]. It is

$$\begin{aligned} \hat{\mathcal{C}}_t &= \beta \mathcal{N}, \\ \hat{\beta}_t &= \left(\frac{1}{2}\hat{\beta}^2 + \frac{a}{\mu}\right)\kappa, \end{aligned} \tag{20}$$

where a is a regularization constant, κ is the curvature and μ is the mass density constant for the curve. In the implicit signed-distance level set representation, the equivalent curve dynamics are then given by

$$\begin{aligned} \hat{\Psi}_t &= \hat{\bar{\beta}} \cdot \left|\left|\nabla \hat{\Psi}\right|\right|, \\ \hat{\bar{\beta}}_t &= \left(\frac{1}{2}\hat{\bar{\beta}}^2 + \frac{a}{\mu}\right)\nabla \cdot \left(\frac{\nabla \hat{\Psi}}{\left|\left|\nabla \hat{\Psi}\right|\right|}\right), \end{aligned} \tag{21}$$

which relies on the facts that Ψ is a signed-distance function and that $\nabla\Psi \perp \nabla\hat{\bar{\beta}}$ (since $\bar{\beta}$ is β extended along the normal characteristics of the curve).

In addition to the curve states, the curve covariances must be propagated. While the constant velocity motion model is linear, the second-order dynamical model is nonlinear. Given that the evolution equations are nonlinear in nature, a linear discrete approximation to the covariance update is needed. Here, the first-order approximation is

$$\mathbb{P}(s, t + \Delta t) = \mathbb{F} \cdot \mathbb{P}(s, t) \cdot \mathbb{F}^T + \mathbb{Q} \cdot \Delta t,$$

where

$$\mathbb{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 + \beta\kappa\, \Delta t \end{bmatrix},$$

and $\mathbb{Q}$ is the process noise covariance. Notice that the state transition model in the case of the constant velocity model is given by:

$$\mathbb{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}.$$

When the curve is updated discretely, the curve covariance is likewise updated. Furthermore, the update occurs for the extended covariance matrix $\bar{\mathbb{P}} : D \rightarrow \mathbb{R}^{2x2}$, which is the covariance matrix $\mathbb{P}$ extended to the entire level set domain. After each time-step, the covariance is extended, much like the normal velocity (see [57] for details on evolving extended quantities).

The final predicted state will be denoted by $\hat{\mathcal{C}}^-$, $\hat{\beta}^-$, and $\mathbb{P}^-$. In the implicit representation with extended fiber elements, the prediction consists of $\hat{\Psi}^-$, $\hat{\bar{\beta}}^-$, and $\bar{\mathbb{P}}^-$.

7.5.2 Measurement Model

As in the first-order case, the curve position measurement $\mathcal{C}^m$ for the current frame is obtained from any foreground/background segmentation algorithm applied to the current image, so long as the final result is a level set function Ψ^m. To ensure proper alignment of the curves, either a localization procedure [5, 26] should be applied before segmentation or a registration method after segmentation.

Measurements for the curve velocities are obtained by computing the optical flow $(u, w)^T$ between two consecutive images and projecting it onto the unit normals of Ψ^m:

$$\bar{\beta}^m = \begin{pmatrix} u \\ w \end{pmatrix} \cdot \frac{\nabla \Psi^m}{||\nabla \Psi^m||}.$$

Note that this measurement is defined over the entire image. The measured velocities, β^m, will be the normal velocities obtained by evaluation of $\bar{\beta}^m$ on the zero level set, $\beta^m = \bar{\beta}^m \circ (\Psi^m)^{-1}(0)$.

Here again, curve measurements need to be converted into the transverse coordinate system obtained from the prediction and the measurement curves.

7.5.3 Update Model

The update model will refine the prediction once state measurements become available. Here, for each particle along its corresponding single-dimensional transverse curve, an update is performed on the state prediction. Given the setup, along one of these transverse curves, the estimation problem faced is one of linear filtering for which an optimal solution

is described by the Kalman filter [39, 72]. Following the work described in the previous section, define the curve error, the predicted curve error, and the measurement curve error by

$$\hat{\mathbf{e}}(s) = \hat{\mathbf{x}}(s) - \mathbf{x}(s),$$

$$\hat{\mathbf{e}}^{-}(s) = \hat{\mathbf{x}}^{-}(s) - \mathbf{x}(s), \text{ and}$$

$$\mathbf{e}^{m}(s) = \mathbf{x}^{m}(s) - \mathbf{x}(s),$$

and curve covariances

$$\mathbb{P}(s) = E\left(\hat{\mathbf{e}}(s)\hat{\mathbf{e}}^{T}(s)\right) > 0,$$

$$\mathbb{P}^{-}(s) = E\left(\mathbf{e}^{m}(s)\left(\mathbf{e}^{m}(s)\right)^{T}\right) > 0,$$

and $\mathbb{P}^{m} = \mathbb{R} > 0$ where $\mathbb{R}$ is the measurement error covariance (smoothly varying along the curve). It is assumed that the measurements and predictions are independent, $\text{Cov}(\mathbf{x}^{-}, \mathbf{x}^{m}) = 0$. This setup reproduces the formalism introduced in the previous section but is second-order.

The optimal correction gain associated to these errors and covariances under the update law, given in point representation,

$$\mathbf{x}^{+}(s) = \mathbf{x}^{-} + \mathbb{K}(\mathbf{x}^{m} - \mathbf{x}^{-}), \tag{22}$$

is $\mathbb{K} = \mathbb{P}^{-}\left(\mathbb{P}^{-} + \mathbb{R}\right)^{-1}$. In the remainder of this section, the gain matrix $\mathbb{K}$ is decomposed as:

$$\mathbb{K} = \begin{bmatrix} K_{xx} & K_{xv} \\ K_{vx} & K_{vv} \end{bmatrix}.$$

The following discussion sketches the implementation of the update equations for the implicit representation of the curve and its fiber (velocity plus covariance).

7.5.3.1 Curve Position Correction

In point notation, the update law of the curve position is obtained from Equation (22),

$$\hat{x}^{+} = \hat{x}^{-} + K_{xx} \cdot \left(x^{m} - \hat{x}^{-}\right) + K_{xv} \cdot \left(v^{m} - \hat{v}^{-}\right). \tag{23}$$

The curve defined by $\hat{x}^{+}(s)$ becomes the posterior curve estimate.

7.5.3.2 Curve Velocity Correction

The velocity component is updated according to

$$\hat{v}^+ = \hat{v}^- + K_{vx} \cdot \left(x^m - \hat{x}^-\right) + K_{vv} \cdot \left(v^m - \hat{v}^-\right). \tag{24}$$

7.5.3.3 Covariance Update

The covariance update is

$$\mathbb{P}^+ = (\mathbb{1} - \mathbb{K})\ \mathbb{P}^-. \tag{25}$$

The position and velocity updates are separated above because each must be dealt with separately. Their update equations follow naturally from Equation (22), however the covariance update needs a special procedure. To perform the update, predicted and measured covariances need to be transported to the updated curve location where they can be compared. Considering only the first element $\mathbb{P}_{11}$ of the covariance matrix, transport is done by solving the advection equation:

$$\phi_\tau + X_{char}^T \cdot \nabla\phi = 0, \tag{26}$$

with $\phi(\cdot; \tau = 0) = \mathbb{P}_{11}$, and τ an artificial time parameter. A similar procedure is applied to the remaining elements of the covariance matrix. By extending the covariance along the characteristics and performing the update over the domain, the update is performed for the implicit representation. Such an approach facilitates reconstruction of the updated curve, its normal velocity, and its covariance after which the curve is re-initialized to be a signed-distance function while the normal velocity and covariance are extended along the normal characteristics.

7.6 Experiments and Results

7.6.1 Error Measurement

One of the fundamental assumptions underlying the method is that curve error is localized. Being so, a locally valid linear description is feasible. To verify when such locality occurs, we generated noisy grayscale images to segment. Additionally, we hand-segmented the images to obtain ground truth. Then we computed the Laplace error of the system as a

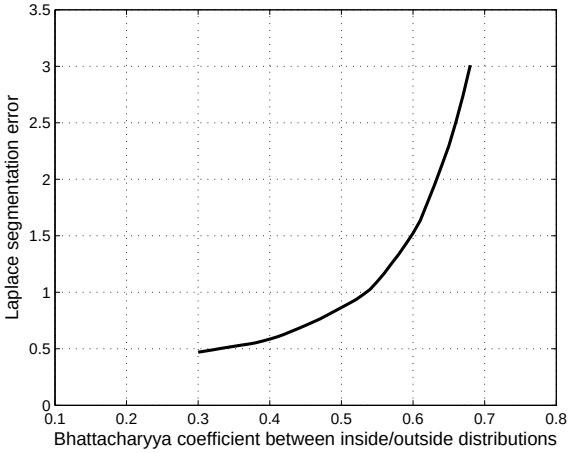
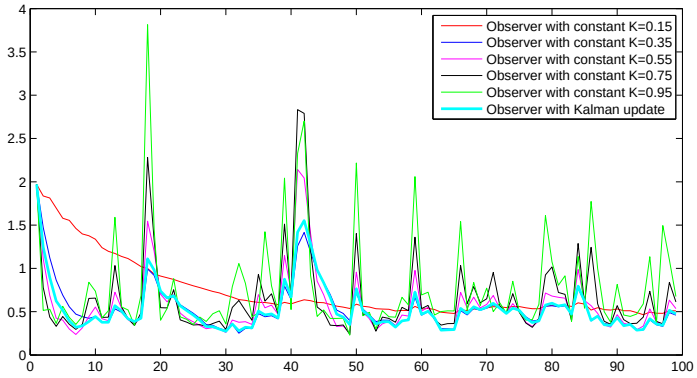


Figure 30: Expected Laplace error versus Bhattacharya coefficient.

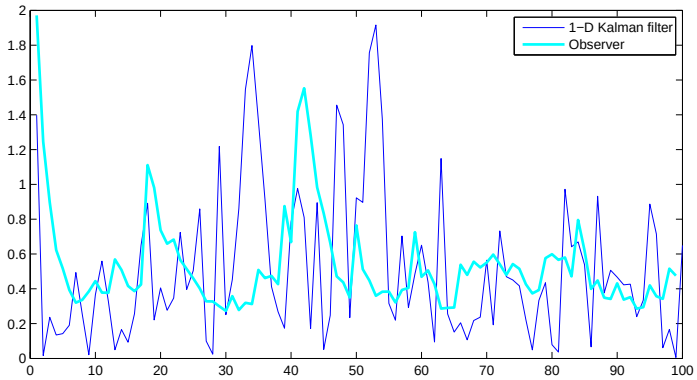
function of the level of noise. We recall that the Laplace error [62] is defined by computing the transverse curves between a given segmenting curve and the true curve and estimating locally on the segmenting curve the travelling distance to the true curve. The mean value along the curve provides an average curve error while the maximum value represents the worst (local) deviation from the truth. The latter error metric is qualitatively similar to the Hausdorff distance. The level of noise is measurable through the Bhattacharyya coefficient between intensity distributions of the foreground and background. Figure 30 depicts the expected error as a function of the noise level. At the level 0.6 and above, the image starts to get sufficiently corrupted that the segmentations become non-sensical (the error variance spikes; not depicted). Thus, for reasonable imagery, we can expect the error to be sufficiently local.

7.6.2 Static Filtering

In order to verify that the behavior as derived is occuring, we performed tests on a static image corrupted with noise. A series of segmentations were generated and then locally filtered using the first-order estimator. Additionally, the local filtering strategy was applied but using a fixed constant gain. A gain sweep from 0.05 to 0.95 in 0.05 gain increments was performed to verify if the gain of the optimal filter converged to its indeed optimal value. Several sweeps were run, one of which is depicted in Figure 31(a) for a limited set of gains (to avoid figure clutter). The red curve corresponds to a low gain, and little update



(a) Gain sweep test.



(b) Visual comparison against a true 1D system.

Figure 31: Static filtering experiments.

of the observer state. The cyan curve represents the optimal gain based on the derived mathematics.

Additionally, several 1D Kalman filter simulations were run to compare the curve filter simulations against a true 1D system. Figure 31(b) depicts the evolution of the error for a 1D system and the mean Laplace error for a simulated static tracking scenario. The converged error variance of the two systems is of the same order.

For this experiment, the gain was selected to be $K(s,0) = K_0$, and the variances $P(s) = P_0$, $Q(s) = Q_0$, and $R(s) = R_0$. Thus, as a function $K(s,k) = K_0(k)$.

7.6.3 Tracking Using the First-Order Filter

Tracking experiments were performed on a color sequence from the construction database. The optimal first-order filtering algorithm and three other contour-based tracking algorithms were tested. The three algorithms were: an active contour technique [70], the deformotion filter [36], and a shape-based filter [19]. Implementation of the contour equations was done in the level-set formulation [71]. For all these tracking techniques, we applied Kalman filtering on the rigid group state with the same measurement and noise covariance matrices.

As mentioned earlier, the shape database consisted of 67 sample shapes obtained from the construction imagery with the top 10 eigenvectors kept. For the deformotion tracking technique [36], a gain was selected to the best of our understanding. For the proposed method, we defined the observation and process noises, Q and R, for both the group and the shape spaces.

Manual segmentations of the video sequences provide the ground truth. For quantitative comparisons of the results obtained with the different techniques, local and global shape metrics are used to quantify deviations of the shapes from the truth. The number of misclassified pixels and the mean and maximum Laplace errors [62] are used for this purpose. Lastly, the number of frames tracked throughout the sequence is indicated.

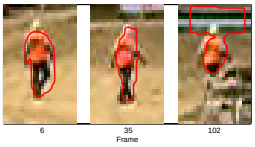
Figure 32 depicts samples of the segmentations obtained at given frames for each sequence and technique. Table 7 summarizes the performances of the different techniques under the comparison metrics. Strikeouts in the text indicate loss of track. Of note, for the test sequence considered, the measurement model fails due to shared statistics of the target and background segmentation models. The filter is able to gracefully handle the poor model.

Table 7: Quantitative comparison metrics on sequence 1.

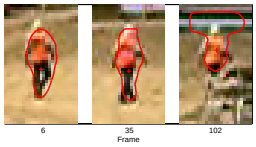
Metric / Algorithm	AC	Deformotion	Shape	Filter
Trackpt error (L_2/L_∞)	~~2.2 / 6.6~~	~~2.2 / 9.6~~	~~7.6 / 18.5~~	1.8 / 6.2
NMP (avg/max)	~~78 / 202~~	~~72 / 172~~	~~87 / 160~~	63 / 111
Mean Laplace (avg/max)	~~1.0 / 3.7~~	~~0.9 / 3.1~~	~~1.2 / 2.6~~	0.7 / 1.3
Max Laplace (avg/max)	~~2.9 / 8.9~~	~~2.3 / 7.9~~	~~3.4 / 8.4~~	2.0 / 3.5
# Frames tracked	~~109~~	~~109~~	~~115~~	350

(a) Sequence 1

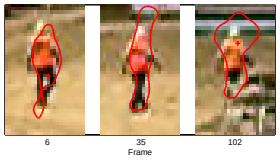
(b) Ground Truth



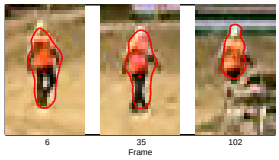
(c) ~~Active Contour~~



(d) ~~Deformotion Filter~~



(e) ~~Shape-based~~



(f) Local Optimal Filter

Figure 32: Snapshots of Sequence 1.

7.6.4 Tracking Using the Second-Order Filter

Since the second-order model is primarily motivated by the need to better estimate curve's deformations, we chose four test sequences presenting large and fast changes in shape. The first sequence of images comes from a construction database, the next two come from an aquarium database and the last sequence is a low SNR grayscale video featuring the deformation of a biomembrane. The first three experiments use the second-order constant velocity model while the last experiment uses the general purpose dynamical model. For comparison purposes, we manually segmented the construction and biomembrane sequences and 100 frames from each aquarium sequence to obtain ground truth. In addition to the number of misclassified pixels, the smoothness index described in §5.6 is employed.

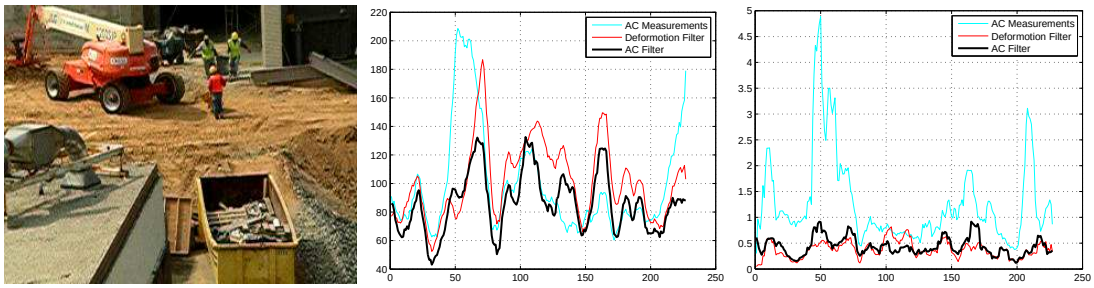
In the first experiment, the aim is to track a construction worker. Active contour measurements spike when other workers presenting similar color distributions approach the target. Our second-order filter is able to attenuate these perturbations when they occur but also capture the correct shape when facing moderate to low segmentation noise. Comparatively, the deformotion filter eliminates the perturbations but tends to oversmooth. Figure 33 (a-g) depicts sample estimates obtained with the different tracking techniques as well as the estimation error and the smoothness indices.

The second experiment targets a fish in an aquarium. The movement and shape deformations of the target are dramatically fast in this test sequence. Moreover, the color distributions of the target and elements of the background overlap significantly. This results in deteriorating performance for the active contour measurements. Both the deformotion filter and the optimal second-order filter drastically attenuate the perturbations and maintain temporal consistency of the target's shape (see Figure 33 (h-n)).

Figure 34 depicts sample estimates and the results corresponding to the third experiment. Here, a fish is quickly deforming with fast movements of the caudal fin. A small overlap between target and background color distributions results in acceptable active contour measurements with small local perturbations and some jittering on a frame-to-frame basis. Due to the shape variations, application of the deformotion filter smoothes out the shape perturbations but introduces a lag between estimates and the true shapes. The optimal second-order filter is able to eliminate those perturbations while adapting much faster to the shape variations thanks to the second-order model involving the curve velocities.

Lastly, the deforming biomembrane was tracked using the dynamical prediction model. Due to the low image quality and the nature of the target, the active contour technique is unable to provide accurate object boundaries. Filtering through the deformotion procedure and the local optimal filter provide means to correct the imperfect measurements.

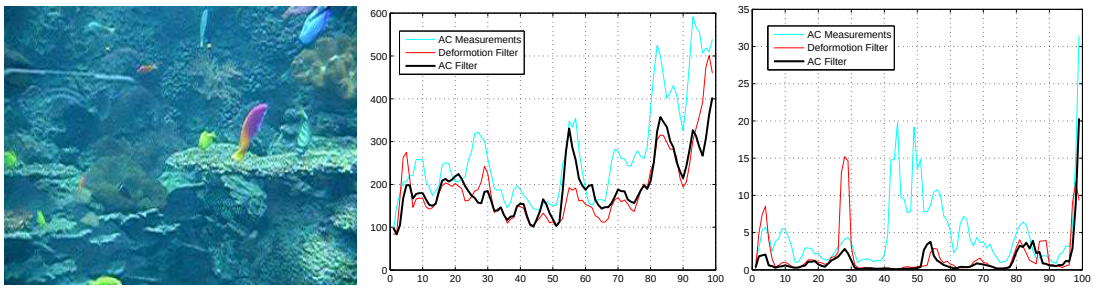
To sum up, these experiments indicate that the optimal second-order filter is capable of attenuating curve perturbations and maintaining a good fidelity to the true curve being estimated. While temporal consistency of the shape is obtained, it is not achieved with detrimental oversmoothing at the expense of curve fidelity.



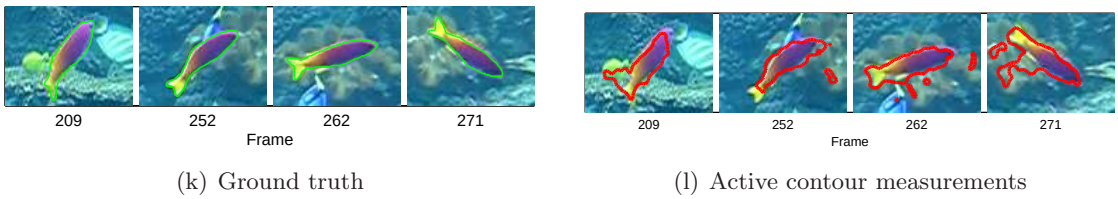
(a) Sample Frame (sequence 2) (b) NMP error vs. time (c) Smoothness index

(d) Ground truth (e) Active contour measurements

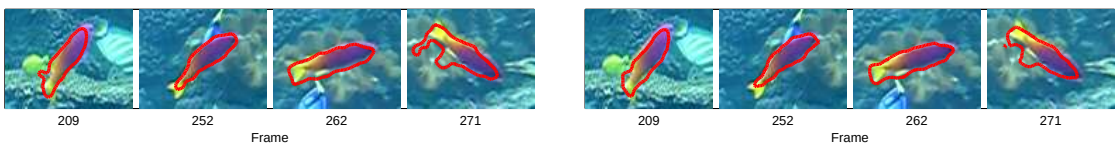
(f) Deformotion filter (g) Local Optimal Filter



(h) Sample Frame (sequence 3) (i) NMP error vs. time (j) Smoothness index



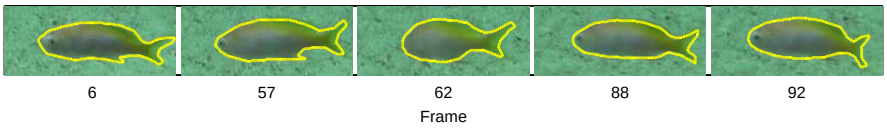
(k) Ground truth (l) Active contour measurements



(m) Deformotion filter (n) Local Optimal Filter

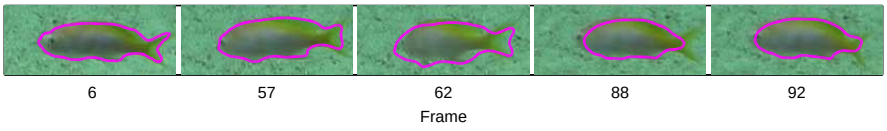
Figure 33: Quantitative comparison of performance for two sequences. The NMP, a smoothness index and sample estimates are depicted.

(a) Sample Frame (sequence 4)

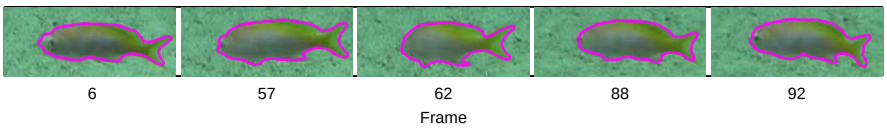


(b) Ground truth

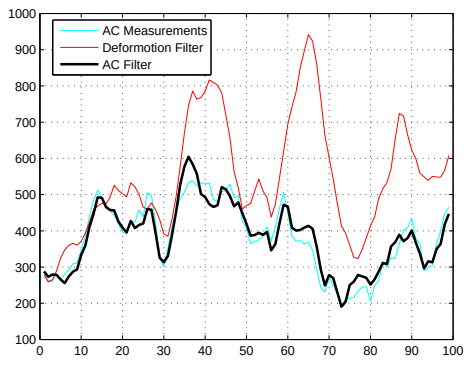
(c) Active contour measurements



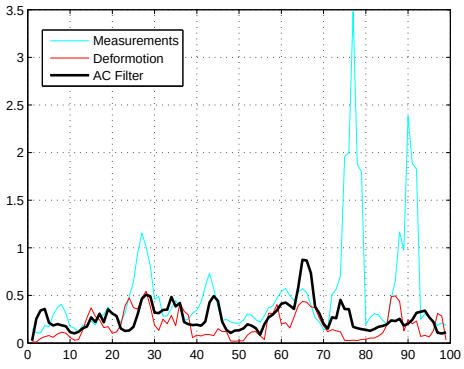
(d) Deformotion filter



(e) Local Optimal Filter

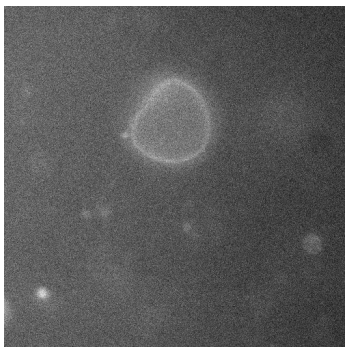


(f) NMP error vs. time

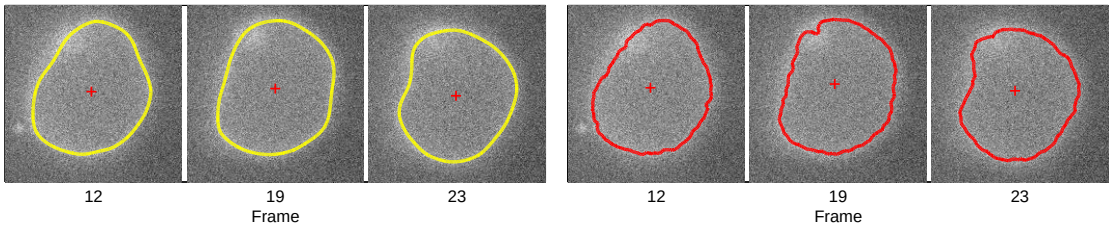


(g) Smoothness index

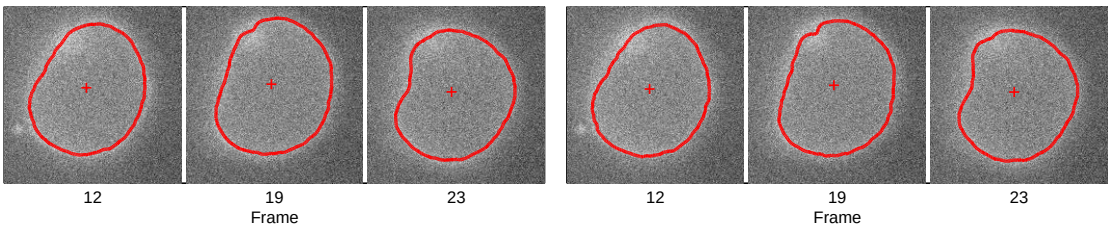
Figure 34: Comparison of performance for sequence 4. Sample estimates are displayed for each tracking technique, along with the number of misclassified pixels and a smoothness index capturing the temporal consistency of the target.



(a) Sequence 5



(b) Ground Truth (c) Active Contour



(d) Deformotion Filter (e) Local Optimal Filter

Figure 35: Snapshots of Sequence 5.

Table 8: Quantitative comparison metrics on sequence 5.

Metric / Algorithm	AC	Deformotion	Filter
Trackpt error (L_2/L_∞)	5.4 / 7.5	5.5 / 7.7	5.4 / 7.5
NMP (avg/max)	363 / 616	429 / 692	348 / 627
Mean Laplace (avg/max)	1.0 / 1.6	1.2 / 1.8	0.9 / 1.6
Max Laplace (avg/max)	4.0 / 7.9	4.0 / 8.9	3.6 / 6.8
# Frames tracked	40	40	40

7.7 Conclusion

This chapter proposed methods for estimating the optimal gain of a curve filtering strategy. The curve error is based upon a Laplace-Poisson PDE solution and provides a set of linear coordinate frames from which to perform curve operations and also compute curve variances. Following up on these results, the filter's state is extended to include both the curve's position and velocity. The resulting second-order model accounts more accurately for curve velocities, which results in better curve estimates when dealing with highly deformable objects.

The curve filter was tested on static images and compared against a 1D Kalman filtered system to verify that convergence behavior is as expected. Additionally, the filter design was validated by applying the technique to recorded imagery and comparing objectively its performance to other tracking techniques using quantitative error metrics. In particular, both the first-order and second-order filters were applied to different sets of image sequences. These experiments show that the techniques are well suited to estimate the position and deformations of deformable curves, especially in the face of noisy measurements as induced by image disturbances and imperfect segmentation models.

CHAPTER VIII

GEOMETRIC AVERAGING FOR STATISTICAL METHODS IN COMPUTER VISION

8.1 Introduction

This chapter seeks to extend the application range of the geometric averaging update model to other statistical methods in computer vision. Probability fields are often generated by machine learning classifiers [2], segmentation algorithms [32] and other statistical signal processing methods. While [22, 32, 50] rely upon a probability field to describe the target shape, [2] uses the term *confidence map* to describe a similar scalar field. In many cases, even when such a probabilistic field or confidence map is not naturally defined by the statistical method, it is relatively straightforward to generate one from other existing similarity/distance maps. Note that in general the performance of the tracking methods is intimately linked to the signal-to-noise ratio of the confidence map since the target state is usuallly obtained through simple post-processing procedures, e.g. thresholding. Therefore, the application of a filtering procedure to estimate the map should improve the overall performance of the system.

8.1.1 Contributions

The aim of this work is to apply the geometric averaging filtering strategy to a certain class of statiscal methods using confidence maps. Instead of considering the confidence map measurements to be noise-free (as implicitly implied by the current techniques), we assume the existence of perturbations in the measurement process. On the space of densities, the uncertainty is modelled as multiplicative. This fundamental assumption is used to design a recursive filtering technique to estimate the true confidence map from the noisy measurements. For tracking applications, the estimated confidence map should then result in more successful and accurate target definition.

8.1.2 Organization

The chapter is organized as follows. Section 8.2 describes the filtering strategy. Sections 8.3 discusses the experimentations using a machine learning tracking method known as ensemble tracking. Section 8.4 concludes the chapter.

8.2 The Geometric Averaging Filter

This section briefly summarizes the recursive filtering strategy that will be used. The key assumption for the filter design consists in modelling the uncertainty (on the space of densities) to be multiplicative, rather than additive. For simplicity, the filtering steps are detailed at a given pixel. Implementation for the entire confidence map follows.

8.2.1 The Update Model

Correction on the state will be performed using the geometric averaging update model. Chapter 3 showed how the traditional assumption of additive noise on the image data could lead to multiplicative uncertainty for the segmentation probabilities when Bayesian segmentation is used. Here, at the outset, we assume multiplicative uncertainty for the measured probabilities. In the log space, where multiplicative error becomes additive error, linear correction can be performed. After correction in the log-space, transformation back to the probability space results in the geometric averaging update model.

8.2.2 Observations

Combining the geometric averaging update model with a measurement strategy and a dynamic prediction model results in a sub-optimal recursive Bayesian filter. The sub-optimality arises because of the use of a non-optimal (arbitrary) gain. Deriving the optimal gain associated to the filtering goes beyond the scope of this chapter. This involves estimating the error covariance associated to the measurements, which is problem-dependent and may be an intractable problem in some cases due to the high nonlinearity of certain machine learning classifiers.

8.2.3 Implementation

Statistical methods that make use of confidence maps can employ the geometric averaging filter to obtain better estimates of the maps. In such cases, an instance of the filter will be run at each pixel. Overall, the pixels are arranged on a grid defined over a compact domain Ω of the plane. Let P denote the confidence map to be filtered. At time t, the filtering can be completed by performing the following operations:

- obtain prediction: $\hat{P}_t^- = f(\hat{P}_{t-1}^+)$, where f is the prediction function.
- obtain measurement P_t^m from the computer vision algorithm.
- obtain update: $\hat{P}_t^+ = \left(\hat{P}_t^-\right)^{1-K_t} \cdot (P_t^m)^{K_t}$

Measurement and prediction need to be aligned before applying the update law. For segmentation algorithms, localization can be applied prior to performing segmentation in order to guarantee the alignment of the measurement and the prediction. When localization is not applicable, e.g. kernel-based methods, a registration procedure can be applied to align measurement and prediction prior to performing the update.

The gain parameter K_t is defined on the grid Ω. Given the specific problem to solve and available information, K_t can be taken constant or variable. For example, occlusion detection or noise estimation routines can be used to adjust the gain parameter, both temporally and spatially. The filtering has a very low computational cost as it involves only simple scalar operations. Moreover, given the grid structure, it is easily implementable on dedicated architectures such as GPU's in order to accelerate performances.

8.3 Filtering the Ensemble Tracker

This section applies the geometric averaging filter to the *ensemble tracking* procedure [2].

8.3.1 Ensemble Tracking

Ensemble tracking approaches the visual tracking problem from a machine learning perspective to achieve binary classification at each frame. The technique considers a colletion

of weak classifiers that are trained on-line to differentiate target from background. Subsequently, AdaBoost is applied to the colllection of weak classifiers to form a strong classifier used to evaluate a confidence map on the next frame. Such a confidence map describes at each pixel of the next frame, the probability that it belongs to the foreground. Application of mean-shift to the confidence map allows the retrieval of the peak, which is the location of the target object in the next frame. The collection of weak classifiers is subsequently updated with a new weak classifier trained on the incoming image data. Thus, the accuracy of the trackpoint obtained with the ensemble tracking method is highly dependent upon the confidence map.

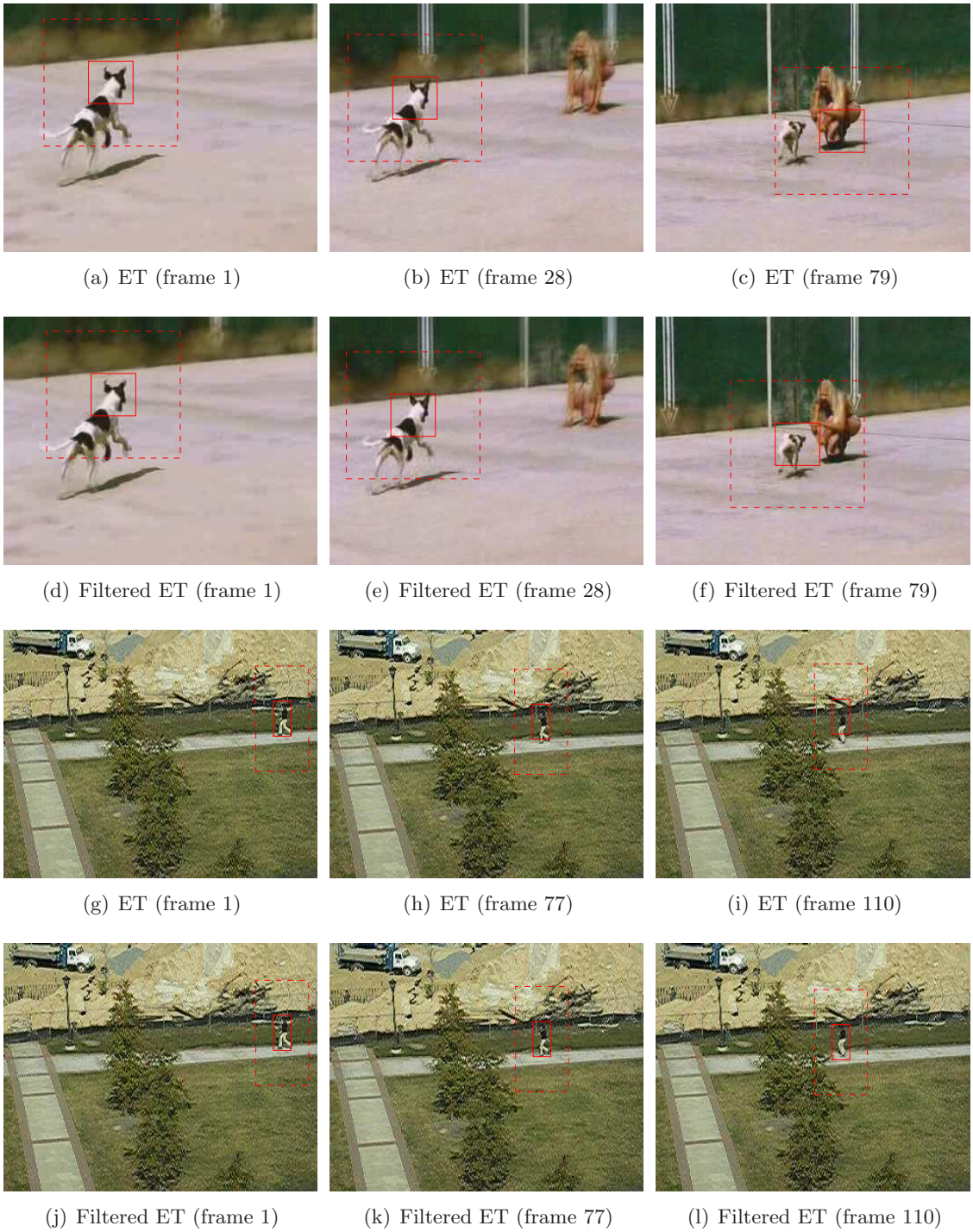
8.3.2 Experimentation

8.3.2.1 Setup

We propose to apply the recursive filter described in the previous section to estimate the confidence map generated by the ensemble tracker. We used two color sequences to perform the experimentation. The first sequence features a running dog while the second sequence shows a walking person on a campus yard. The sequences are not trivial to track due to clutter and target/background overlapping distributions. For comparison purposes, we manually defined the trackpoint on the sequences in order to obtain ground truth. Following, the ensemble tracker was applied to the two test sequences. The recursive filter was then used in conjunction with the ensemble tracker, the confidence map being filtered prior to performing mean shift on it.

8.3.2.2 Analysis

Sample frames are depicted in Figure 36. They show that the recursive filter is able to improve upon the ensemble tracking procedure. In the first sequence, while the unfiltered ensemble tracker loses track near the end of the sequence, application of the filtering enabled successful tracking during the entire sequence. Further, the computation of the $L2$ error on the trackpoint shows better performance (see Figure 37). This is clearly visible on the second sequence where perceptible fluctuations of the trackpoint can be eliminated by the filter.



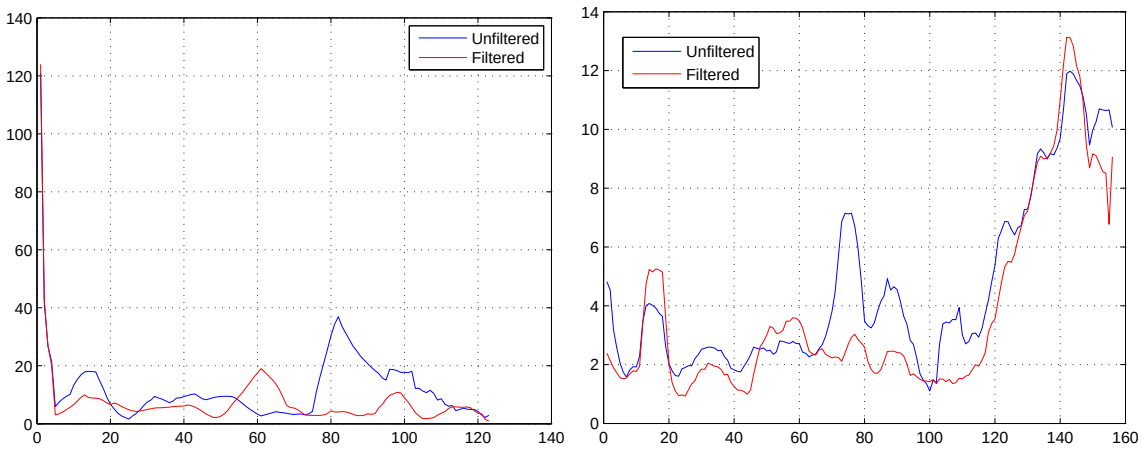
(a) ET (frame 1) (b) ET (frame 28) (c) ET (frame 79)

(d) Filtered ET (frame 1) (e) Filtered ET (frame 28) (f) Filtered ET (frame 79)

(g) ET (frame 1) (h) ET (frame 77) (i) ET (frame 110)

(j) Filtered ET (frame 1) (k) Filtered ET (frame 77) (l) Filtered ET (frame 110)

Figure 36: Geometric averaging filter applied to the ensemble tracker: sample frames.



(a) L_2 error (dog sequence) (b) L_2 error (yard sequence)

Figure 37: Geometric averaging filter applied to the ensemble tracker: L_2 error metric.

8.4 Conclusion

This chapter expands the range of applications for the geometric averaging filter. Chapter 3 showed that the hypothesis of multiplicative uncertainty for the measured segmentation probabilities could be exploited to derive a successful recursive filtering strategy. Going beyond the Bayesian segmentation case, an additional example drawn from the machine learning field was successfully improved using the filtering scheme. For a large variety of statistical methods used in computer vision, it may be reasonable to make an assumption of multiplicative uncertainty on the state measurements. This fundamental assumption justifies the use of the geometric averaging filter to obtain better state estimates, which in turn contributes to improving the performance of the related statistical methods.

CHAPTER IX

CONCLUSION

This thesis explored filtering schemes for the purpose of dynamic curve estimation. The overall hypothesis assumed that the construction of estimators for temporally evolving curves would lead to improved performance of visual tracking systems. The state-space representations chosen decompose the state of deformable moving objects into a finite-dimensional group action and infinite-dimensional local shape deformations that describe the rigid motion and the non-rigid motion respectively. Filtering strategies on each component were then decoupled. Filtering on the finite-dimensional group component being trivial, this work focused essentially into designing shape filtering strategies.

The first part of the thesis utilized an implicit probabilistic shape description. The consideration of the probabilistic shape representation simplifies the (challenging) filtering design on the infinite-dimensional space of curves to a series of point-wise filtering tasks. Appropriate dynamical prediction models and a novel geometric averaging update model were developed while designing the components of a probabilistic contour observer. In some related work, a methodology to characterize segmentation algorithms was devised, leading to the gain adaptation of the contour observer. Subsequently, an optimal contour observer was built using a bottom-up approach starting from the measurement process. In particular, the derivation of the optimal correction gain results from the fundamental assumption that foreground and background can be described with Gaussian distributions. While this seems to restrict potential tracking scenarios to piecewise smooth images, the introduction of simple Gaussian mixture models provides a way to deal with complex appearance models arising from natural scenes, without the need to re-design the optimal estimator. The determination of the optimal correction gain and its formal relation to measurable uncertainty on the image data constitutes an essential result. In practice, it eliminates the need to manually tune the gain parameter of the visual tracking algorithm.

In the second part of the thesis, locally optimal closed curve filtering strategies were explored. Level set representations for smooth, closed curves described the shape space. We introduced a local, linear description for planar curve variation and curve uncertainty. It consists of a family of non-intersecting trajectories transverse to a given curve. Along one of the single-dimensional transverse trajectories, linear curve operations are feasible. Using the linear operation, simple locally optimal filtering procedures were derived. In particular, an optimal first-order curve filtering strategy was developed. To deal with highly elastic objects, curve velocities were incorporated in the model, yielding sub-optimal second-order filtering strategies.

Once the curve filtering schemes are derived, they are placed within the greater context of observer design for the estimation of a curve's position and deformations as it evolves in the plane. Application to online visual tracking is emphasized through experimentation with recorded imagery and objective comparison to other tracking methods. The quantitative validation of the estimator designs was verified using both synthetic and real sequences of images, manual segmentations that served as ground truth and a variety of relevant error metrics. These experiments demonstrated that the estimators presented here constitute an effective class of solutions to the visual tracking problem. They compare favorably to other tracking techniques operating in an online, recursive setting. Furthermore, the computational complexity associated to these estimators is low.

Potential directions for continuing the research presented in this thesis would include robustifying the estimators, investigating methods to accurately model the noise and robustly estimate the noise statistics. A further avenue of inquiry arising in the context of this study is the role of shape. Shape is necessary when there is a massive amount of false positive image content extending over long times. Current methods use hard shape constraints to restrict the measurement to lie within a low-dimensional parametrized space. We suspect that a more robust approach consists in adopting looser constraints on the measurement model while enforcing harder constraints on the correction model, in contrast to the prevailing methods which impose hard constraints during measurement. The fundamental problem is to find a satisfactory balance between enforcing shape constraints at

measurement and filtering out (observation) noise at correction. On a different note, the difficulty associated with the nonlinear curve dynamics indicate that optimal second-order filtering for dynamic curves may be best handled within the context of curve geodesics by improving upon [41, 77].

Finally, the ideas presented in this thesis extend beyond contour-based algorithms. Chapter 8 expanded the utility of the geometric averaging model to a larger group of confidence-based statistical methods. In particular, it demonstrated how the application of the filtering scheme improved the overall performance of a template-based tracking method. Future work should explore other areas in computer vision to see how the fundamental analysis of estimation theory can lead to algorithmic improvements.

REFERENCES

[1] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[2] Avidan, S., "Ensemble tracking," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 494–501, 2005.

[3] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R., "A database and evaluation methodology for optical flow," in *proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[4] Bertalmio, M., Sapiro, G., and Randall, G., "Morphing active contours: a geometric approach to topology-independent image segmentation and tracking," in *proceedings of the IEEE International Conference on Image Processing*, pp. 318–322, 1998.

[5] Bibby, C. and Reid, I., "Robust real-time visual tracking using pixel-wise posteriors," in *proceedings of the European Conference on Computer Vision*, pp. 831–844, 2008.

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

[7] Boykov, Y. and Funka-Lea, G., "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[8] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1124–1137, 2004.

[9] Brockett, R. and Blake, A., "Estimating the shape of a moving contour," in *proceedings of the IEEE Conference on Decision and Control*, pp. 3247–3251, 1994.

[10] Brown, R. and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering*. New York: John Wiley and Sons, Inc., 1992.

[11] Caselles, V., "Geometric models for active contours," in *proceedings of the IEEE International Conference on Image Processing*, pp. 9–12, 1995.

[12] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," in *proceedings of the IEEE International Conference on Computer Vision*, pp. 694–699, 1995.

[13] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," *International Journal of Computer Vision*, vol. 13, pp. 5–22, 1997.

[14] Chan, T. and Vese, L., "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, pp. 266–277, 2001.

[15] Charpiat, G., Faugeras, O., and Keriven, R., "Approximations of shape metrics and application to shape warping and empirical shape statistics," *Foundations of Computational Mathematics*, vol. 5, no. 2/2, pp. 1–58, 2005.

[16] Chockalingam, P., Pradeep, N., and Birchfield, S., "Adaptive fragments-based tracking of non-rigid objects using level sets," in *proceedings of the IEEE International Conference on Computer Vision*, 2009.

[17] Cohen, L. D. and Kimmel, R., "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78.

[18] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[19] Cremers, D., "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1262–1273, 2006.

[20] Cremers, D., "Nonlinear dynamical shape priors for level set segmentations," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2007.

[21] Cremers, D., Osher, S., and Soatto, S., "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[22] Cremers, D., Schmidt, F., and Barthel, F., "Shape priors in variational image segmentation: convexity, Lipschitz continuity and globally optimal solutions," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, 2008.

[23] Dambreville, S., Rathi, Y., and Tannenbaum, A., "Tracking deformable objects with unscented Kalman filtering and geometric active contours," in *proceedings of the IEEE American Control Conference*, pp. 2856–2861, 2006.

[24] Dryden, I. and Mardia, K., *Statistical Shape Analysis*. John Wiley & Sons, 1998.

[25] Duta, N., Jain, A., and Dubuisson-Jolly, M., "Automatic construction of 2D shape models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 433–446, 2001.

[26] Elgammal, A., Duraiswami, R., and Davis, L. S., "Probabilistic tracking in joint feature-spatial spaces," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 781–788, 2003.

[27] Freedman, D. and Zhang, T., "Active contours for tracking distributions," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 518–526, 2004.

[28] Friedland, B., *Control System Design: An Introduction to State-Space Methods*, pp. 259–289. McGraw-Hill Higher Education, 1985.

[29] Garcin, L. and Younes, L., "Geodesic matching with free extremities," *Journal of Mathematical Imaging and Vision*, vol. 25, pp. 329–340, 2006.

[30] Haker, S., Sapiro, G., and Tannenbaum, A., "Knowledge-based segmentation of SAR images," in *proceedings of the IEEE International Conference on Image Processing*, pp. 597–601, 1998.

[31] Haker, S., Sapiro, G., and Tannenbaum, A., "Knowledge-based segmentation of SAR data with learned priors," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 299–301, 2000.

[32] Haker, S., Sapiro, G., Tannenbaum, A., and Washburn, D., "Missile tracking using knowledge-based adaptive thresholding: Tracking of high speed projectiles," in *proceedings of the IEEE International Conference on Image Processing*, pp. 786–789, 2001.

[33] Healey, G. and Kondepudy, R., "Radiometric CCD camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, 1994.

[34] Holm, D., *Geometric Mechanics Part 2: Rotating, Translating and Rolling*, pp. 109–123. Imperial College Press, 2008.

[35] Horn, B. and Schunck, B., "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[36] Jackson, J., Yezzi, A., and Soatto, S., "Tracking deformable moving objects under severe occlusions," in *proceedings of the IEEE Conference on Decision and Control*, pp. 2990–2995, 2004.

[37] Julier, S. and Uhlmann, J., "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[38] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A new approach for filtering nonlinear systems," in *proceedings of the IEEE American Control Conference*, pp. 1628–1632, 1997.

[39] Kalman, R., "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[40] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: active contour models," in *proceedings of the IEEE International Conference on Computer Vision*, pp. 259–268, 2001.

[41] Klassen, E., Srivastava, A., Mio, W., and Joshi, S., "Analysis of planar shapes using geodesic paths on shape spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 372–383, 2004.

[42] Malcolm, J., Rathi, Y., and Tannenbaum, A., "Graph cut segmentation with nonlinear shape priors," in *proceedings of the IEEE International Conference on Image Processing*, pp. 365–368, 2007.

[43] Malcolm, J., Rathi, Y., and Tannenbaum, A., "Tracking through clutter using graph cuts," in *proceedings of the British Machine Vision Conference*, p. 116, 2007.

[44] Martin, P., Refregier, P., Goudail, F., and Guerault, F., "Influence of the noise model on level set active contour segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 799–803, 2004.

[45] Maurel, P., Keriven, R., and Faugeras, O., "Reconciling landmarks and level sets," in *proceedings of the IEEE International Conference on Pattern Recognition*, pp. 69–72, 2006.

[46] Maybeck, P. S., *Stochastic models, estimation and control. Volume I.* Press, Academic, 1979.

[47] Michor, P. and Mumford, D., "Riemannian geometries on spaces of plane curves," *Journal of the European Mathematical Society*, vol. 8, pp. 1–48, 2004.

[48] Michor, P. and Mumford, D., "An overview of the Riemannian metrics on spaces of curves using the Hamiltonian approach," *Applied and Computational Harmonic Analysis*, vol. 23, no. 1, pp. 74–113, 2007.

[49] Ndiour, I. J., Arif, O., Teizer, J., and Vela, P. A., "A probabilistic shape filter for online contour tracking," in *proceedings of the IEEE International Conference on Image Processing*, pp. 433–436, 2009.

[50] Ndiour, I. J., Arif, O., Teizer, J., and Vela, P. A., "A probabilistic observer for visual tracking," in *proceedings of the IEEE American Control Conference*, 2010.

[51] Ndiour, I. J., Teizer, J., and Vela, P. A., "A probabilistic contour observer for online visual tracking," *to appear in SIAM Journal on Imaging Sciences*, 2010.

[52] Ndiour, I. J. and Vela, P. A., "A local extended Kalman filter for visual tracking." *to appear in proceedings of the IEEE Conference on Decision and Control*, 2010.

[53] Ndiour, I. J. and Vela, P. A., "Noise estimation and adaptive filtering during visual tracking," in *proceedings of the IEEE International Conference on Image Processing*, pp. 4365–4368, 2009.

[54] Ndiour, I. J. and Vela, P. A., "Towards a local Kalman filter for visual tracking," in *proceedings of the IEEE Conference on Decision and Control*, pp. 2420–2426, 2009.

[55] Ndiour, I. J. and Vela, P. A., "Optimal estimation applied to visual contour tracking," in *proceedings of the IEEE American Control Conference*, 2010.

[56] Neumaier, A. and Schneider, T., "Estimation of parameters and eigenmodes of multivariate autoregressive models," *Transactions on Mathematical Software*, vol. 27, no. 1, pp. 27–57, 2001.

[57] Niethammer, M., *Dynamic Level Sets for Visual Tracking.* PhD thesis, Georgia Institute of Technology, 2004. `http://hdl.handle.net/1853/7606/`.

[58] Niethammer, M., Tannenbaum, A., and Angenent, S., "Dynamic active contours for visual tracking," *IEEE Transactions on Automatic Control*, vol. 53, no. 2/2, pp. 153–167, 2003.

[59] Niethammer, M., Vela, P. A., and Tannenbaum, A., "Geometric observers for dynamically evolving curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1093–1108, 2008.

[60] Papadakis, N. and Mémin, E., "Variational optimal control technique for the tracking of deformable objects," in *proceedings of the IEEE International Conference on Computer Vision*, pp. 1–7, 2007.

[61] Papadakis, N. and Mémin, E., "Variational assimilation of fluid motion from image sequence," *SIAM Journal on Imaging Sciences*, vol. 1, no. 4, pp. 343–363, 2008.

[62] Pichon, E., Nain, D., and Niethammer, M., "A Laplace equation approach for shape comparison," *SPIE Medical Imaging*, vol. 2, pp. 24–132, 2006.

[63] Porikli, F., Tuzel, O., and Meer, P., "Covariance tracking using model update based on Lie algebra," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 728–735, 2006.

[64] Pryor, G., ur Rehman, T., Lankton, S., Vela, P. A., and Tannenbaum, A., "Fast optimal mass transport for dynamic active contour tracking on the GPU," in *proceedings of the IEEE Conference on Decision and Control*, pp. 2681–2688, 2007.

[65] Rank, K., Lendl, M., and Unbehauen, R., "Estimation of image noise variance," in *proceedings of the IEE Vision, Image and Signal Processing*, pp. 80–84, 1999.

[66] Rao, B. and Durrant-Whyte, H., "Fully decentralised algorithm for multisensor Kalman filtering," in *proceedings of the IEE Control Theory and Applications*, pp. 413–420, 1991.

[67] Rathi, Y., Dambreville, S., and Tannenbaum, A., "Comparative analysis of kernel methods for statistical shape learning," in *proceedings of the 2nd International Workshop on Computer Vision Approaches to Medical Image Analysis*, pp. 96–107, 2006.

[68] Rathi, Y., Vaswani, N., and Tannenbaum, A., "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1370–1382, 2007.

[69] Rathi, Y., Vaswani, N., Tannenbaum, A., and Yezzi, A., "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.

[70] Rousson, M. and Deriche, R., "A variational framework for active and adaptative segmentation of vector valued images," in *proceedings of the IEEE Workshop on Motion and Video Computing*, pp. 56–61, 2002.

[71] Sethian, J., *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[72] Simon, D., *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. Wiley and Sons, 2006.

[73] Singh, M., Arora, H., and Ahuja, N., "Robust registration and tracking using kernel density correlation," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 174–182, 2004.

[74] Sorenson, H. W., *Kalman filtering: theory and application.* IEEE Press, 1985.

[75] Stam, J., "Real-time fluid dynamics for games," in *Proceedings of the Game Developer Conference*, 2003.

[76] Sundaramoorthi, G., Jackson, J., Yezzi, A., and Mennucci, A., "Tracking with Sobolev active contours," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[77] Sundaramoorthi, G., Mennucci, A., Soatto, S., and Yezzi, A., "Tracking deforming objects by filtering and prediction in the space of curves," in *proceedings of the IEEE Conference on Decision and Control*, pp. 2395–2401, 2009.

[78] Tipping, M. and Bishop, C., "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[79] Torr, P., Szeliski, R., and Anandan, P., "An integrated Bayesian approach to layer extraction from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 297–303, 2001.

[80] Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W., and Willsky, A., "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, 2003.

[81] Unal, G., Krim, H., and Yezzi, A., "A vertex-based representation of objects in an image," in *proceedings of the IEEE International Conference on Image Processing*, pp. 896–899, 2002.

[82] Unal, G., Krim, H., and Yezzi, A., "Fast incorporation of optical flow into active polygons," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 745–759, 2005.

[83] Vaswani, N., Rathi, Y., Yezzi, A., and Tannenbaum, A., "Deform PF-MT: Particle filter with mode tracker for tracking non-affine contour deformations," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 841–857, 2010.

[84] Vaswani, N., Yezzi, A., Rathi, Y., and Tannenbaum, A., "Time-varying finite dimensional basis for tracking contour deformations," in *proceedings of the IEEE Conference on Decision and Control*, pp. 1665–1672, 2006.

[85] Vela, P. A. and Ndiour, I. J., "Estimation theory and tracking of deformable objects," in *to appear in proceedings of the IEEE Multi-Conference on Systems and Control*, 2010.

[86] Vela, P. A., Niethammer, M., Pryor, G., Tannenbaum, A., Butts, R., and Washburn, D., "Knowledge-based segmentation for tracking through deep turbulence," *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 469–474, 2008.

[87] Wang, J. and Adelson, E., "Layered representation for motion analysis," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 361–366, 1993.

[88] Xiao, J. and Shah, M., "Motion layer extraction. in the presence of occlusion using graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1644–1659, 2005.

[89] Yezzi, A. and Mennucci, A., "Metrics in the space of curves," *Preprint, arXiv:math.DG/0412454*, 2005.

[90] Yezzi, A. and Soatto, S., "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *International Journal of Computer Vision*, vol. 53, no. 2/2, pp. 153–167, 2003.

[91] Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[92] Younes, L., "Optimal matching between shapes via elastic deformations," *Image and Vision Computing Journal*, vol. 17, pp. 381–389, 1999.