Closed loop visual tracking using observer-based dynamic active contours

Patricio A. Vela* Marc Niethammer* Jimi Malcolm†

Allen Tannenbaum*

Georgia Insitute of Technology, Atlanta, GA, USA

{pvela,marcn}@ece.gatech.edu, malcolm@cc.gatech.edu, tannenba@ece.gatech.edu

This paper describes and demonstrates an algorithm to visually track flying vehicles in a leader-follower scenario. The algorithm consists of two nested control loops. One that assures a proper position estimation for the object(s) to be followed (the visual observer) and one that exploits this position estimation to control the maneuvers of the follower such that it pursues the leader. The visual observer is based on some novel geometric principles tailored to controlled active vision. In particular, we describe the use of dynamic active contours as motion priors for such geometrically motivated observers.

I. Introduction

Visual tracking aims at estimating the position (and possibly additional quantities) of one or multiple (moving) objects throughout time. A controlled active vision system uses these visually estimated quantities to achieve desired control objectives. In the case of this paper, our goal is to control the maneuvers of a flying vehicle in order to follow a leader in a prespecified manner. While visual sensing is unobtrusive, the resulting image information may in general not be straightforward to analyze. Indeed, visual tracking (due to the presence of noise, occlusions, and image clutter) is a challenging problem, and closing the loop using only the resulting visual information (usually) demands additional post-processing (for example, additional estimation steps) to relate image information back to controllable quantities.[1] This paper focuses on the visual tracking aspect of the controlled active vision problem. In particular, it concentrates on the use of dynamic active contours as motion priors for geometrically motivated observers for visual tracking.

A large body of research has been and is devoted to the formulation of observers for dynamical systems. While a comprehensive theory exists for linear dynamical systems, the theory for nonlinear and/or infinite dimensional systems is still in its infancy and often times limits itself to special system classes.[2] In the context of visual tracking, finite dimensional system models have been especially successful. These allow for the application of classical observer theory, like the Kalman filter and its derivatives (e.g., the extended Kalman filter, the unscented Kalman filter, etc.) as well as for the more recent and increasingly popular particle filtering approaches. The robust performance and stability of such an observer depends on the system model applied, the quality of measurements obtained as well as its ability to reject disturbances (e.g., process and measurement noise like image clutter). Unfortunately, only rarely is a good system model available. Also, ideally the dynamical description would need to take into account (for the single camera

*School of Electrical and Computer Engineering, Atlanta, GA, 30332-0250, USA.

†College of Computing, Atlanta, GA, 30332-0280, USA.

case) the projection of an originally three-dimensional scene on a two-dimensional image plane – in itself a strong nonlinearity. Consequentially, simplified finite dimensional motion models are usually employed: "rigid, similarity, affine, projective as well as quadratic models".[3]

While visual trackers using finite dimensional system models may lead to high levels of robustness, they suffer from descriptive inflexibilities: the chosen motion group needs to be rich enough to describe all possible object segmentations based on a single fixed object template (shape). Tracking deforming (e.g., elastic) objects requires infinite dimensional motion groups or the ability to change a shape template over time. Both approaches have been proposed previously, from an optimization point of view: the visual tracker by Jackson *et al.*[4] for example is based on the ideas of Yezzi and Soatto[5] where finite dimensional motion group action is combined with an infinite dimensional deformation required to be "as small as possible." See Paragios and Deriche[3] for a related approach.

This paper aims at an infinite dimensional filtering perspective on visual tracking. To this end an infinite dimensional motion model is needed. We employ the dynamic active contour[6] as a motion prior (uncoupled from image information) and perform error injection to account for deviations between system states and measurements[a] resulting structurally in an observer-based dynamic active contour. Measurements are performed by static segmentations and local optical-flow measurements. While only a curve is evolving dynamically, the measurement step can incorporate any kind of static segmentation, potentially non-edge based for increased robustness.

The implementation of the proposed visual tracking framework is completely implicit, using level set evolutions and transport equations to relate information from the measurements to the dynamically evolving curve. This naturally allows (if necessary) for topological changes during tracking: an object breaking into multiple independent parts can still be followed, since there is no global finite dimensional motion model. To facilitate future real-time visual tracking, the proposed algorithm is of comparatively low computational cost. Its performance is demonstrated on simulated image sequences.

II. Setup

This paper is concerned with the tracking of a leading Unmanned Aerial Vehicle (UAV) (the leader) by another UAV (the follower) without communication between the two vehicles. The complete tracking closed loop system is summarized by the block diagram of Figure 1. It is assumed that each UAV has it's own inner controller which receives acceleration commands from the guidance system.[1] The follower's only leader-dependent measurements with which to infer the leader's state are given by a fixed, forward-pointing monocular camera mounted on the follower. The guidance module requires the relative range r, the line of sight (LOS) angle λ and the its rate $\dot{\lambda}$ as an input. These quantities are estimated by an extended Kalman filter as given in Betser *et al.*[1] The Kalman filter's inputs are the relative accelerations normal and tangent to the LOS, the LOS angle and the angle subtended by the leader in the image plane. The leader needs to be identified within the camera image to extract the latter two geometrical quantities. Identifying the leader amounts to image segmentation. To account for possible measurement noise this segmentation is performed using an observer-based dynamic active contour that reconciles the predicted outline of the leader with its measured outline (see Section IV).

III. Main Tracker

We first review some of the results on tracking from Betser *et al.*[1]

In the setting of this paper, the leader follows an unknown trajectory relative to which the follower must track. The desired range and the desired relative (lead) angle between the leader and the follower are denoted by $r_1^\star$ and $\epsilon_1^\star$, respectively.

The Guidance block receives inputs from the Estimation block, namely: the estimated relative range $\hat{r}_1$,

[a]For more details and extensions to this approach see Niethammer *et al.*[7]

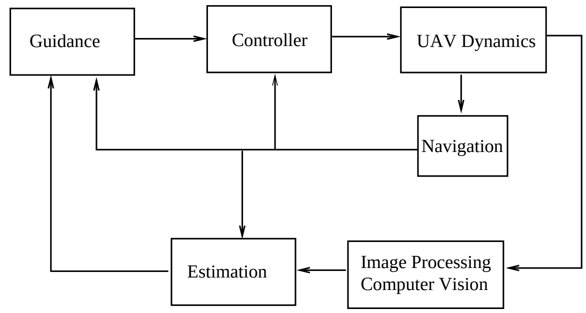


Figure 1. System Block Diagram

the estimated relative angle $\hat{\epsilon}_1$, and the estimated LOS rate $\hat{\dot{\lambda}}_1$. In addition, it gets two predefined commands: the desired relative range r_1^* and the desired relative (lead) angle ϵ_1^*. The outputs of the algorithm are the acceleration commands to the inner control block of the UAV. Control acceleration for the autonomous vehicle is decomposed into normal and tangential acceleration components, denoted as a_{c1}^N and a_{c1}^T, respectively.

The purpose of the input a_{c1}^N is to maintain the desired LOS angle between the leader and follower. The algorithm is based on standard Proportional Navigation and LOS guidance laws.[8] Since the leader dynamics are unknown, we assume for the guidance law that the leader is stationary. Using the estimated values for the LOS and the LOS rate we define the normal acceleration command as:

$$a_{c1}^N = NV_1 \left(K_N \hat{\dot{\lambda}}_1 + (1 - K_N) e_{\epsilon_1} \right). \tag{1}$$

Here N is the proportional navigation constant, K_N is a parameter in the range $[0, 1]$, V_1 is the forward velocity of the follower, and $e_{\epsilon_1} \equiv \epsilon_1^* - \hat{\epsilon}_1$, where $\hat{\epsilon}_1$ and $\hat{\dot{\lambda}}_1$ are the estimates of ϵ_1 and the LOS rate $\dot{\lambda}_1$, respectively.

We note that the role of a_{c1}^T is to track the desired relative range between the leader and follower. Consequently, a_{c1}^T is a function of the range error, $e_{r_1} = r_1^* - \hat{r}_1$, where $\hat{r}_1$ is the estimated relative range between the leader and the follower. Any velocity or acceleration control loops designed to control the error signal should take into account the model of the follower.

IV. Observers for Visual Tracking

The second part of our proposed strategy is based on a theory of geometric observers we have been developing.[7] It is closest in spirit to previous work on joint segmentation and registration[5] and the resultant tracking methodology.[4] While these works propose a finite dimensional motion model with an overlayed deformation to account for deviations from a current shape template, we propose the use of an infinite dimensional motion model whose motion gets implicitly constrained by the measurement (which can then be shape-constrained, area-based, etc.). Since the dynamics of the proposed observer are based on measurements and geometrically motivated gains (explicit distance measurements between the observed curve or surface and the measured curve or surface), gain selection becomes more intuitive. Filtering position information is not always necessary and may be replaced by the position measurements alone. Additional quantities (like velocities) can then be propagated (and filtered) along with a moving curve whose position is assumed to be known exactly. From now on, we only allow curve evolutions with normal velocities (which is obviously a restriction for rotating objects). Preliminary results on such geometric observers have been obtained in some of our recent work.[7]

A. The Problem

Central to our observer framework are the the following four issues:

- Which observer structure is desirable; which quantities need to be estimated and which ones can simply be measured?
- What are suitable system models for the prediction part of the observer?
- Can we say anything about the robustness properties of the resulting observer, or if not, what can we do to design the observer at least qualitatively as robust as possible?
- How does one establish correspondences between two curves (surfaces)?

In the classical observer framework (e.g., as proposed by Luenberger[9]) we have a prediction and a measurement part. The prediction part incorporates all the dynamical assumptions we make about our plant, or in the context of visual tracking the movement of our object. In the simplest case this dynamical assumption is the static assumption (i.e., there is no movement). In general, the prediction part will not contain any measurement information. The measurement corrects for imperfections in the assumed dynamical model.

Observer gains for measured quantities depend on the level of trust one can put in them. If one assumes uncertain measurements, but a relatively good dynamical prediction model, the prediction part should be weighted more heavily than the measurement part. In the opposite case, or if the measurement is especially "trustworthy," the measurement should impact the dynamical evolution of the observer significantly, if the deviation from the simulated (predicted) state is significantly different from the measurement. In the extreme case we can regard (individual or all) measurements as ground-truth. This leads to reduced order observers: the observer no longer needs to estimate a particular quantity (e.g., if we know that position estimates will be precise, because of extremely good image contrast – a noise-free black blob moving on a white background – we no longer need to estimate position).

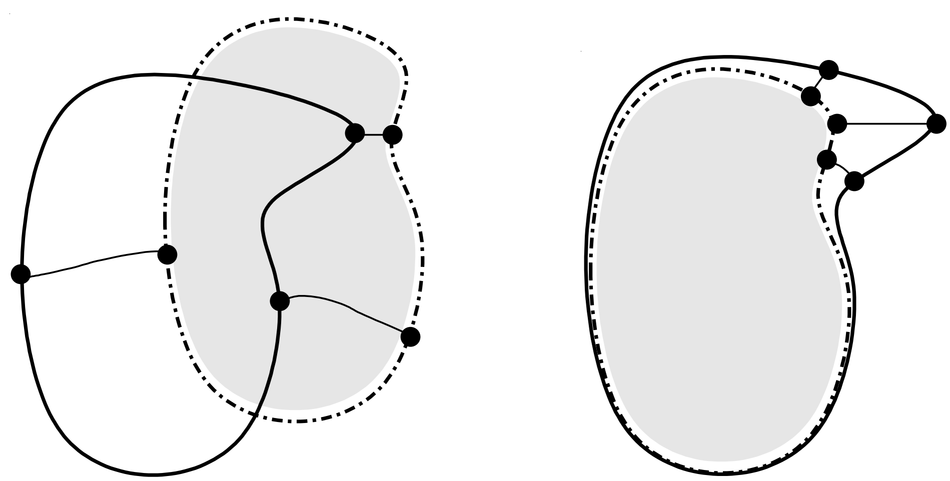
Frequently, good dynamic models will not be available or will not be necessary. The best we can hope for in many cases is a crude approximation with respect to propagation direction. This calls for measurements with large areas of convergence and is directly linked to the issue of robustness.

B. Proposed Observer Structure for Curves

In analogy with classical observer theory, our proposed observer structure will contain a prediction and a measurement part. To evolve the overall estimated curve, the prediction influence has to be combined with the measurement influence. The proposed observer for curves (see Niethammer *et al.*[7] for details) assumes continuous dynamics for the curve and discrete-time measurements. The proposed observer is geometric in the sense that it uses geometric quantities computed from the currently estimated curve and the measured curve (see Figure 2). The general structure follows the standard observer structure depicted in Figure 3.

It proceeds in the following manner:

0) Initialize the evolving curve (position and velocity).
1) Propagate the estimated curve by one time increment (when a new camera measurement becomes available) based on a chosen motion model.
2) Perform a static segmentation on the camera image resulting in the measurement curve.
3) Compute correspondence points and correspondence trajectories between the predicted curve and the measurement curve.
4) Interpolate the predicted and the measured curves to obtain the position of the estimated curve.
5) Interpolate corresponding additional curve states based on the predicted curve's states and the measured states to obtain the remaining states of the estimated curve.
6) Repeat from 1.



(a) Simulated curve (solid line) and measured curve (dash-dotted) including correspondence lines.

(b) Measured curve (dash-dotted) without shape information and measured curve (solid line) with shape information including correspondence lines.

Figure 2. Observer structure. A measurement curve is used to correct for the possibly faulty curve state generated by the prediction part of the dynamic curve evolution alone. By evolving two different measurement contours, one with and one without shape-information and computing distances between correspondence points, a measure of merit can be calculated for every point of the measurement contour assessing the goodness of fit of the shape-based measurement contour to the underlying image information.

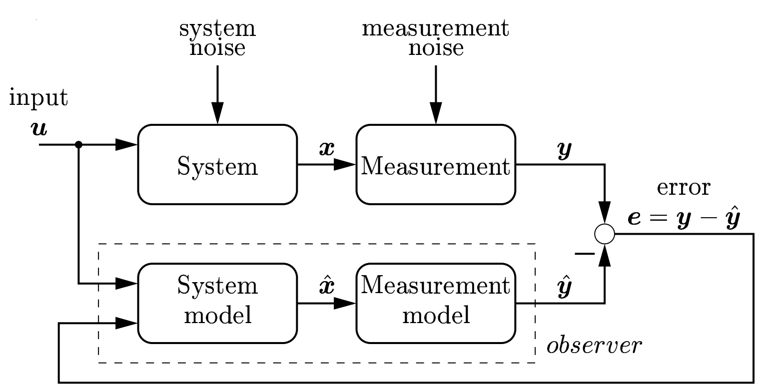


Figure 3. General Observer Structure. The error e is implicitly defined by means of the correspondence procedure. The measurement makes use of standard segmentation algorithms and the prediction part is given by a chosen prior.

Section C describes possible motion models to predict the curve state from one image frame to the next. Section D talks about obtaining system measurements from a camera image and the curve observer. Finally, Section E discusses curve interpolation, curve alignment, and error correction, which completes the proposed observer.

C. Motion Models

The motion model is a motion prior. It is problem dependent and should model as precisely as possible the dynamics of the object(s) to be tracked. Ideally, the measurement part of an observer should only need to correct for inaccuracies due to noise. However, in practice it will be difficult (or even impossible) to provide an exact motion model so that the measurement part also needs to compensate for inaccuracies of the motion prior.

Possible motion priors[7] are the static motion prior (the simplest of them all, where no inter-frame motion is assumed)

$$\hat{\mathcal{C}}_t = \mathbf{0},$$

the constant velocity prior

$$\hat{\mathcal{C}}_{tt} = \mathbf{0},$$

and the dynamic elastic prior (restricted to normal curve evolution).

$$\mu\hat{\mathcal{C}}_{tt} = \left(\frac{1}{2}\mu\|\hat{\mathcal{C}}_t\|^2 + a\right)\kappa\mathcal{N} - (\nabla a\cdot\mathcal{N})\mathcal{N} - \frac{1}{2}\mu(\|\hat{\mathcal{C}}_t\|^2)_s\mathcal{T}.$$

Here, $\hat{C}(p,t) : S^1 \times [0,\tau) \mapsto \mathbb{R}^2$ denotes the position of a closed, simple, planar curve, where $p \in [0,1]$ is the curve's parameterization on the unit circle S^1, t denotes time, $\hat{\mathcal{C}}(p,t) = [x(p,t),\ y(p,t)]^T$, and $\hat{\mathcal{C}}(0,t) = \hat{\mathcal{C}}(1,t)$; subscripts denote partial derivatives, s arclength, κ is signed curvature $\mathcal{T}$ is the curve's unit tangent and $\mathcal{N}$ its unit inward normal. The dynamic elastic prior is based on the dynamic active contour.[6] Here, a is independent of image information, but is a design parameter for curve regularization, μ denotes the mass parameter controlling the curve's inertia. In contrast to the constant velocity prior, the dynamic elastic prior exhibits regularization forces (i.e., accelerations) that try to push in curve convexities and push out curve concavities; it is dynamically regularizing.

D. Measurements

The predicted measurement is based on the current state of the predicted curve, i.e., its current position, velocities, etc. Any of the standard segmentation algorithms may be used to come up with the "real" measurement (obtained from a camera image).

This observer setup has two crucial advantages:

- While the dynamic model is a model of a dynamically evolving curve, the measurement can be based for example on area-based, or region-based segmentation algorithms.
- Static and dynamic approaches incorporating shape information already exist.[5,10] If these approaches are used for the measurement curve, shape information can be introduced into the infinite dimensional model without the need for explicit incorporation of the shape information in the dynamical model.

Including shape information in the measurement part will introduce such information into the dynamically evolving curve. This is in contrast to previous approaches that aimed at including shape information into the dynamics of an evolving curve itself, e.g., the CONDENSATION filter based curve trackers of Blake and Isard[11] where the motion gets restricted to affine motion.

1. *Measurement of Additional Quantities*

Whereas the position measurement is arguably the most important measurement, additional quantities may need to get measured. When using a dynamic prior (in conjunction with a dynamic observer) velocity measurements may be required. These can be estimated, for example, by measuring the optical flow at the measured position of a curve. Assuming that there is strong edge information at the boundaries of the measured curve, normal optical flow can be computed as

$$\begin{pmatrix} u \\ v \end{pmatrix} = -\frac{\partial I}{\partial t}\frac{\nabla I}{\|\nabla I\|^2},$$

where I denotes image intensity and u and v are the computed velocities in x and y direction respectively. If there is no strong boundary information regularization has to be performed to obtain the optical flow field.

E. Interpolation, Alignment, and Error Correction

Given the predicted and the measured curves, the crucial questions are (i) how to define an error between them and (ii) how to feed this error information back to the observer.

To define this error it is necessary to establish point correspondences between the measured and the predicted curves; see Figure 2 for an illustration.

1. *Correspondences*

Previous approaches to compute correspondence points (see for example[6, 11–13]) searched for point correspondences along the normals of the predicted curve, which is problematic in case of large deviations between the two curves to be set into correspondence. For large deviations it is thus desirable to employ a correspondence strategy that is non-local (e.g., area based) to capture large deviations and thus to increase robustness.

Computing curve correspondence points is an elastic registration problem. Given a flow field $\boldsymbol{v}$ that transforms the predicted curve into the measured curve in unit time, correspondence points may be defined based on the induced particle trajectories. One way to obtain this flow field is by solving a Laplace equation between the two curves with constant Dirichlet boundary conditions on the two curves respectively. A scaled version of the gradient of the Laplace solution then determines the flow field.[7] The Laplace approach is fast and gives reasonable correspondences. It may be combined with a curve pre-alignment (e.g., affine transformation based on moment computations or Fourier based methods to account for translation, rotation and scale).

2. *Transporting Information*

To interpolate curve position and additional state quantities (positions, velocities, accelerations, etc.) need to be transported along the correspondence trajectories, from the measured curve towards the predicted curve and vice versa. Given the correspondence flow field $\boldsymbol{v}$ this may simply be achieved by solving the equation

$$\begin{cases} \Xi(\cdot, 0) = \Xi_0, \\ \frac{\partial \Xi}{\partial t} + D\Xi\boldsymbol{v} = 0. \end{cases}$$

to transport quantities Ξ from the predicted to the measured curve (or vice versa with an inverted flow field). D denotes the Jacobian operator and Ξ_0 the initial conditions whose values should coincide with the values to be transported on the predicted curve (or the measured curve for the reversed flow).

3. Curve Interpolation

Given any implicit correspondence flowing the estimated into the measured curve, we can define a distance field d_1 given the velocity field $\boldsymbol{v}$ by computing

$$\begin{cases} \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \cdot \nabla d_1 = 1, \\ d_1(\hat{\mathcal{C}}) = 0. \end{cases}$$

This is a static Hamilton-Jacobi equation that can be solved as in.[14] By solving the equation backwards (i.e., from the measured curve $\mathcal{C}$ to the estimated curve $\hat{\mathcal{C}}$ with correspondence velocity field $\overline{\boldsymbol{v}} = -\boldsymbol{v}$)

$$\begin{cases} \frac{\overline{\boldsymbol{v}}}{\|\overline{\boldsymbol{v}}\|} \cdot \nabla d_2 = 1, \\ d_2(\mathcal{C}) = 0, \end{cases}$$

the overall distance field can be computed as

$$d := d_1 + d_2.$$

Then scaling the original distance field d_1 by d results in

$$d_s := \frac{d_1}{d_1 + d_2} = \frac{d_1}{d},$$

a distance field that can be used to geometrically blend (see Figure 4) the estimated curve into the measured curve such that

$$\begin{cases} \{d_s : d_s = 0\} = \hat{\mathcal{C}}, \\ \{d_s : d_s = 1\} = \mathcal{C}, \end{cases}$$

and values of d_s such that $0 < d_s < 1$ result in interpolations between the estimated curve and the measured curve.

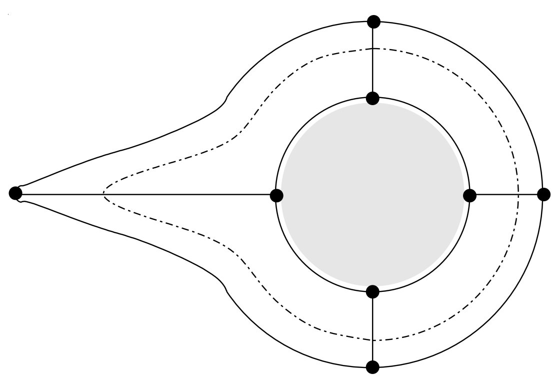


Figure 4. Geometric blending of two curves into each other. The dash-dotted curve denotes the blended one. Since geometric blending is used, the blended curve will approximate the measured one (the circle) increasingly well as d_s approaches 1.

4. Error Correction

The observer models a continuous dynamic system with discrete-time measurements. Error correction is performed at these discrete times in the following way, given a predicted curve, a measured curve, and a flow field $\boldsymbol{v}$ that flows the predicted curve into the measured curve in unit time:

1) Compute a desired interpolated curve as described in Section 3. The result is the estimated curve position.

2) Transport additional state quantities (e.g., velocities) between the measured curve and the predicted curve as described in Section 2. Then interpolate these quantities on the estimated curve of 1) to obtain the estimated additional state information.

5. Shape Prior

A shape prior may be used in the proposed framework in different ways

1) to perform a shape based measurement in addition to an unconstrained measurement, where the resulting curve differences may in future work be used to come up with locally adaptive observer gains,

2) to bias the measurement to past measurements.

The principle is the following: Assume the shape prior $\mathcal{C}_k^{prior}$ (initially this shape prior can be derived from a segmentation without shape, or it can be learned) to be the measurement of the previous frame $k-1$. One way to incorporate shape information into the measurement segmentation at time step k is to follow directly along the lines of Cremers *et al.*[10] where a joint energy gets minimized, penalizing deviations from a template shape (modulo translations and rotations) and deviations from a static segmentation, i.e.,

$$E = (1-\alpha)E_{shape} + \alpha E_{segmentation}.$$

While this is a linear combination of energies it does not guarantee a geometric compromise between shape and image influence. One possible way to perform such a geometric shape based segmentation is to

1) Perform an unconstrained (not shape based) segmentation in frame k.

2) Align the shape template (obtained at frame $k-1$) to the unconstrained segmentation of step 1).

3) Geometrically interpolate the two resulting curves from step 1) and 2) as described in Section 3.

Step 2) may be any kind of registration method (i.e., affine registration).

V. Results, Future Work, and Conclusions

Figure 5 shows a frame from a simulated leader-follower scenario taken by the on-board camera of the follower. The contour is estimated by the visual observer and used to estimate inputs for the extended Kalman filter for range and line of sight estimation. The output of the latter is in turn used to create acceleration commands for the follower to track the leader with a prespecified range and at a prespecified angle.

The prediction uses a static motion prior and affinely aligns the predicted curve to the measured curve before computing correspondences (by means of a Laplace approach) and performing curve interpolation (the error correction).

The proposed visual observer framework has increased robustness in comparison to previous approaches like dynamic active contours, dynamic snakes, and static active contour segmentations. Establishing correspondences through a well behaved flow field is favorable over local measurements, such as searching for

Figure 5. A frame of a simulated leader-follower scenario, as seen from the follower camera. The yellow contour is estimated by the visual observer.

feature points along curve normals. Error correction to obtain the estimated curve position is performed geometrically (by computing actual distances). Thus error injection gains become meaningful quantities.

Future research will explore more complicated scenarios (e.g., cluttered environments), will evaluate differences in tracking quality based on the selected motion priors and will investigate using shape information in greater detail.

Of interest will also be to devise strategies to adaptively tune the gains of the visual observer. One possibility would be to simultaneously run a segmentation with and without shape information (see Figure 2) for the measurement. Based on the difference of the two results we can define a local measurement quality Q, which can, for example directly depend on a distance (similarity) d between the two curves as

$$Q(\mathcal{C}^m, \mathcal{C}^{sh}) = \frac{1}{\epsilon + d(\mathcal{C}^m, \mathcal{C}^{sh})},$$

where $\mathcal{C}^m$ is the measured curve without shape constraint, $\mathcal{C}^{sh}$ is the measured curve with shape constraint and $\epsilon << 1$. Based on this quality of measurement, error injection can be steered selectively, the gains can be made adaptive with high gains for trusted measurements and low gains for dubious ones.

Acknowledgments

We would like to thank Dr. Amir Betser for many helpful insights into visual tracking and about life and the meaning of things in general. This work was supported by grants from the AFOSR, MURI, MRI-HEL, ARO, and NSF.

References

[1]Betser, A., Vela, P. A., and Tannenbaum, A., "Automatic Tracking of Flying Vehicles Using Geodesic Snakes and Kalman Filtering," *Proceedings of the Conference on Decision and Control*, IEEE, 2004.

[2]Wouwer, A. V. and Zeitz, M., *Control Systems, Robotics and Automation, Theme in Encyclopedia of Life Support Systems*, chap. State estimation in distributed parameter systems, EOLSS Publishers, 2001.

[3]Paragios, N. and Deriche, R., "Geodesic active regions and level set methods for motion estimation and tracking," *Computer Vision and Image Understanding*, Vol. 97, 2005, pp. 259–282.

[4]Jackson, J. D., Yezzi, A. J., and Soatto, S., "Tracking Deformable Moving Objects Under Severe Occlusions," *Proceedings of the Conference on Decision and Control*, IEEE, 2004.

[5]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, 2003, pp. 153–167.

[6]Niethammer, M., Tannenbaum, A., and Angenent, S., "Dynamic Geodesic Snakes for Visual Tracking," submitted to the *IEEE Transactions on Automatic Control*, 2004.

[7]Niethammer, M., Vela, P. A., and Tannenbaum, A., "Geometric Observers for Dynamically Evolving Curves," *Proceedings of the Conference on Decision and Control*, IEEE, 2005.

[8]Shneydor, N., *Missile Guidance and Pursuit - Kinematics, Dynamics, and Control*, Horwood Pub., Chichester, 1998.

[9]Luenberger, D. G., "An Introduction to Observers," *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, 1971, pp. 596–602.

[10]Cremers, D. and Soatto, S., "A Pseudo-distance for Shape Priors in Level Set Segmentation," *Proceedings of the International Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, IEEE, 2003, pp. 169–176.

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

[12]Terzopoulos, D. and Szeliski, R., *Active Vision*, chap. Tracking with Kalman Snakes, MIT Press, 1992, pp. 3–20.

[13]Peterfreund, N., "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, 1999, pp. 564–569.

[14]Kao, C. Y., Osher, S., and Tsai, Y.-H., "Fast Sweeping Methods for Static Hamilton-Jacobi Equations," Tech. Rep. 03-75, UCLA, 2003.