AIAA 05-7116

Spacecraft Relative Motion Estimation using Visual Sensing Techniques

Mark J. Monda and Hanspeter Schaub
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0203

AIAA Infotech@Aerospace Conference
September 26–29, 2005 / Arlington, VA

Spacecraft Relative Motion Estimation using Visual Sensing Techniques

Mark J. Monda* and Hanspeter Schaub†
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0203

Precise relative navigation between an unmanned vehicle and a target, which could be a stationary or a second moving vehicle, is an important capability. Possible applications include formation operations, as well as autonomous rendezvous and docking of either spacecraft or aircraft. A test bed setup is described where unmanned ground vehicles are used to simulate the physical motion of aerospace vehicles, and provide the attached sensor packages with realistic relative motion in both indoor and outdoor environments. Using an unmanned robotic ground vehicle, the visual servoing problem is investigated and simulated using actual hardware in real world test conditions. A simple camera and an onboard computer running a color statistical pressure snake algorithm are employed to track a visual target within the camera images in real time. The images are used to estimate the relative position and motion between the vehicle and the target. The effectiveness of the nonlinear visual servoing algorithm is demonstrated through experimental hardware tests.

I. Introduction

MANY future space vehicle systems will require precise relative navigation between different vehicles. For example, a small, unmanned, free-flying scout vehicle that is able to maintain precise formation with a larger vehicle, such as the Space Shuttle or the International Space Station, could serve many purposes. The scout could fly around the larger craft, inspecting for damage or unusual wear and tear. It could also act as a helper vehicle, assisting astronauts performing space construction projects. Or, it could conduct repair or deployment operations in proximity to the larger vehicle.

In all of these applications, precise centimeter-level accuracy relative navigation between the vehicles is required. Many possible technologies have been suggested for achieving this type of precision navigation at extremely close ranges. For example, differential GPS, optical beacon systems, or radio triangulation all show promise in certain applications, but also have drawbacks. Differential GPS would be unable to guide vehicles in proximity to spacecraft flying to the Moon or Mars. In Low Earth Orbits (LEOs) the differential GPS solution would also be susceptible to extreme multi-path errors due to the very close proximity of the scout vehicle to the spacecraft. Optical Beacon systems or radio triangulation require the addition of substantial hardware to the spacecraft, and are dependent on active communication between the vehicles.[1] However, such systems could function at very close ranges.

A novel visual sensing technique using active deformable contours[2–5] is investigated in this paper. Here the scout craft is equipped with a simple camera. Using the deformable contour technique to track a visual feature on the primary spacecraft, the relative motion of the scout craft is estimated. This method has several exciting benefits: First, it does not require any additional hardware on the primary vehicle, nor any active communication between the vehicles. In addition, visual sensing works regardless of location (orbiting about a planet or underway between planets). Finally, visual sensing tends to become more accurate at close ranges, when precise relative navigation is most important.

However, visual sensing techniques have drawbacks as well. It has proven difficult to develop methods that work well in changing lighting conditions. Therefore, it can be difficult to extract precise relative navigation data when

*Graduate Student, Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute
†Assistant Professor, Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute, Senior AIAA Member.

varying lighting conditions cause irregular reflection and high noise levels, especially near the target's edges and boundaries.[6] In addition, visual sensing is usually rather computationally intensive, and thus challenging to use for real-time control applications.

Much work in the field of computer vision has focused on developing relative motion visual sensing techniques. The use of Kalman filtering in these algorithms has proven useful for mitigating the effects of lighting variation and image noise.[7] Many existing algorithms, though, are designed to analyze the motion of an object in an image flow taken from a stationary observer.[8] Matthies, Szeliski, and Kanade have shown that range to a target can be estimated by prescribing a known camera motion.[9] However, in many possible space scout vehicle applications, the opposite is desired: Given an image flow taken from a moving drone vehicle, the relative range, orientation, and motion between the vehicles is sought.

The use of the visual snake sensing technique is explored to extract relative navigation information from an image flow. A collaborative, although non-communicative, target image fixed to the target vehicle is considered. Statistical pressure snake algorithms are used to track the target and calculate properties such as center of mass, area, and side lengths. A Kalman filter is developed to estimate and tune the target size/camera focal length calibration parameter during maneuvers. The scout spacecraft operates in inertial free space and is able to measure its inertial motion through an on-board IMU. The performance of this visual servoing technique is demonstrated in hardware simulations. In Reference 10, a related velocity-based visual guidance law is presented with a constant feedback gain. The visual servo control in this development will allow the feedback gain to be time dependent. This will allow estimator-covariance dependent gain scheduling to be implemented. Compared to common robotic applications, this scout vehicle vehicle is assumed to be free-flying about the primary craft. Any thrusting that occurs is measured using an accelerometer.

II. Visual Sensing using Color Statistical Pressure Snakes

The visual sensor hardware used in these relative navigation experiments consists of a digital camera, a frame-grabber card, and an on-board PC-104 computer. The computer processes the images using a recent real-time statistical pressure snake algorithm.[4] Tracking a target in an image is a segmentation problem. Given the $N \times M$ pixels of the image, the task is to find a contour which will outline the desired target in each frame. How to perform this action in an unstructured environment where the lighting conditions can vary is a very challenging problem which is actively being researched in the computer science imaging community.

In 1987, Kass et al. proposed the original active deformable model to track targets within an images stream.[11] Also referred to as a visual snake, the parametric curve is of the form

$$S(u) = I(x(u), y(u))', \qquad u = [0, 1] \tag{1}$$

where I is the stored image. This curve is then placed into an image gradient derived potential field and allowed to change its shape and position in order to minimize the energy E along the length of the curve $S(u)$. The active deformable models can be divided into two groups:[12] parametric models (snakes)[11,13] and level-set models (geometric contours).[14] The original Kass snake formulation is a parametric snake solution. However, it is very difficult to tune and has several well documented limitations. For example, the target contours tend to implode in the presence of weak gradients. While level-set models show excellent segmentation and robustness capabilities, they remain challenging to implement in real-time applications. Instead, this current test bed setup uses a modified parametric snake formulations proposed by Ivins and Porrill.[2] Here a pressure function is introduced which computes the statistical similarity of pixel values around a control point to create a pressure force which drives the snake towards the target boundaries. The energy function is given by

$$E = \int_0^1 \Big[E_{\text{int}}(S(u)) + E_{\text{pres}}(S(u)) \Big] \mathrm{d}u \tag{2}$$

where the pressure energy function E_{pres} is

$$E_{\text{pres}} = \rho \left(\partial S / \partial u \right)^{\perp} (\epsilon - 1) \tag{3}$$

and ϵ is the statistical error measure of the curve $S(u)$ covering the target. Perrin and Smith suggest replacing the E_{int} expression with a single term that maintains a constant third derivative.[13] This simplified formulation includes an even snake point spacing constraint. The resulting algorithm does not contain the difficult to tune tension and curvature force terms, yielding an easier to use and more efficient parametric snake algorithm.

Numerical efficiency is critical when trying to apply visual snakes to the control of autonomous vehicles. A fast snake point cross-over check algorithm is implemented which yields significant speed improvements for large sets of snake points.[5] The resulting visual snake implementation is referred to as the color statistical pressure snake and is currently operational in the Virginia Tech Autonomous Vehicle Systems (AVS) Lab. The promising tracking performance illustrated in Figure 1 has also been independently verified in Reference 15.

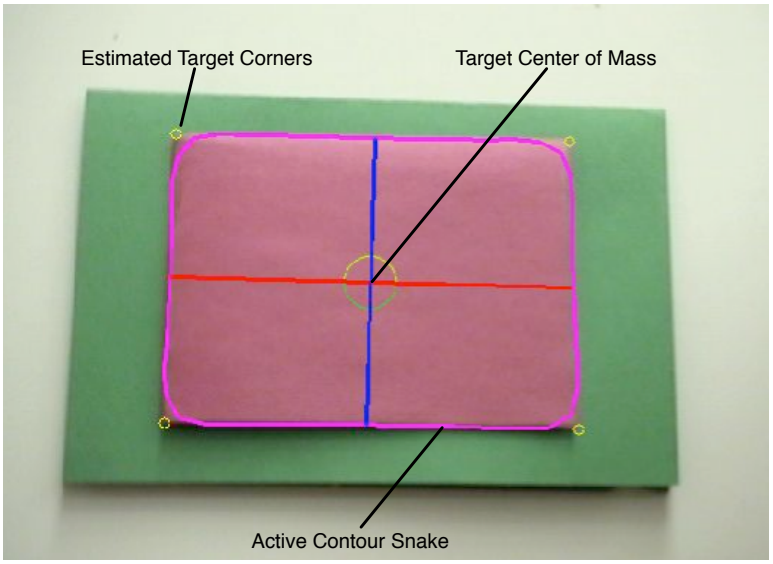


Figure 1: Statistical pressure snake tracking a visual target. The snake, principal axes, and corner locations are shown on the image.

Figure 1 shows an image capture where the visual snake algorithm is tracking a target. This tracking can be performed at the full frame rate of 30 Hz, even with a relatively slow 800 MHz processor. No specialized DSP chips are employed. Further, as will be discussed in Section III., the visual snake contour line can be used to extract additional target features such as target image principal axes direction and size, centroid location,[16] and even target corners.[17]

III. Extracting Target Image Features

A. Calculating the Moments of a Parametric Curve

To extract the desired image shape features, it is required to compute the area moments M_{ij} of the parameterized curve. Given these moments, it is possible to compute the shape area, center of mass (C.M.), and shape eigenfactors. Assume the snake is roughly outlining a shape as shown in Figure 1.

The ij-th area moments of the shape are defined as

$$M_{ij} = \iint x^i y^j \, \mathrm{d}x \, \mathrm{d}y \tag{4}$$

where the area integral should be taken over the area defined by the closed snake curve. If the area A was defined through a set of pixel coordinates (x_n, y_m), then it would be trivial to compute the various area moments through

$$M_{ij} = \sum_{n=1}^{N} \sum_{m=1}^{M} x_n^i y_m^j \tag{5}$$

However, in the current problem setup the area A is defined through a parametric curve outlining this area. Recall Green's theorem which relates an area integral to a line integral through

$$\iint \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \mathrm{d}x \, \mathrm{d}y = \oint P \mathrm{d}x + Q \mathrm{d}y \tag{6}$$

Using this theorem, it is possible to avoid the lengthy area integral and simply compute a line integral along the snake points. An optimized routine discussed in Reference 18 can be used to compute the moments of an area defined through a discrete set of edge points.

Note that the area A of the target shape is simply the 00-th area moment:

$$A = M_{00} = \iint \mathrm{d}x\,\mathrm{d}y \tag{7}$$

The center of mass (x_c, y_c) of the shape is computed using the first area moments M_{10} and M_{01}:

$$x_c = \frac{M_{10}}{A} = \frac{1}{A}\iint x\,\mathrm{d}x\,\mathrm{d}y \tag{8a}$$

$$y_c = \frac{M_{01}}{A} = \frac{1}{A}\iint y\,\mathrm{d}x\,\mathrm{d}y \tag{8b}$$

Since an area integral is used to compute the target C.M., this computation is rather insensitive to noise and minor deviations between the snake contour and the actual target contour. Minor deviations along the snake typically cancel each other and yield a steady and accurate C.M. estimate.

B. Computing the Shape Eigenfactors

Given the target shape area and C.M., we would like to determine the principal shape orientation and size. These features are computed by evaluating the 2^{nd} order moments. Here we assume that the coordinate system origin has been translated to coincide with the computed target shape center of mass. Let the inertia-like matrix $[I]$ be defined as

$$[I] = \begin{bmatrix} M_{20} & M_{11} \\ M_{11} & M_{02} \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{9}$$

Note that $[I]$ is symmetric and positive definite. The principal orientation angle θ determines a coordinate system rotation which will yield a diagonal inertia matrix $[I']$. Let $[C]$ be the direction cosine matrix (rotation matrix)[19] which will map the coordinate axis orientations between the two coordinate frames.

$$[C] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{10}$$

The rotation matrix is defined such that the coordinate transformation

$$[C][I][C]^T = [I'] \tag{11}$$

or

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} = \underbrace{\begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix}}_{[I']} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{12}$$

will result in a diagonal inertia matrix $[I']$. The inertias I_1 and I_2 of the diagonal matrix are called the principal inertias. The angle θ is defined here as the orientation of the I_1 inertia axis. Note that I_1 and I_2 are the eigenvalues of the matrix $[I]$, while the corresponding eigenvectors will determine the principal inertia axes orientations. Since $[I]$ is symmetric and positive definite, the eigenvectors are guaranteed to be orthogonal.

To compute the eigenvalues I_i of $[I]$, we need to solve the quadratic equation

$$\det\left(\begin{bmatrix} I_{xx} - I_i & I_{xy} \\ I_{xy} & I_{yy} - I_i \end{bmatrix}\right) = (I_{xx} - I_i)(I_{yy} - I_i) - I_{xy}^2 = 0 \tag{13}$$

The principal inertias I_1 and I_2 are then given by:

$$I_1 = \frac{1}{2}\left(I_{xx} + I_{yy} + \sqrt{(I_{xx} - I_{yy})^2 + 4I_{xy}^2}\right) \tag{14a}$$

$$I_2 = \frac{1}{2}\left(I_{xx} + I_{yy} - \sqrt{(I_{xx} - I_{yy})^2 + 4I_{xy}^2}\right) \tag{14b}$$

Note that here we define $I_1 \geq I_2$. Thus we can assume that I_1 will always be the largest principal inertia of the target shape.

To determine the eigenvectors of $[I]$ we could solve the standard eigenvector problem. However, since we know that for a symmetric, positive definite matrix the eigenvectors will be orthogonal unit vectors, we can take advantage of this fact by simply solving for the angle θ in Eq. (12). Carrying out the matrix multiplications in Eq. (12) leads to the four equations

$$\cos\theta I_{xx} + \sin\theta I_{xy} = I_1 \cos\theta \tag{15a}$$
$$\cos\theta I_{xy} + \sin\theta I_{yy} = I_1 \sin\theta \tag{15b}$$
$$-\sin\theta I_{xx} + \cos\theta I_{xy} = -I_2 \sin\theta \tag{15c}$$
$$-\sin\theta I_{xy} + \cos\theta I_{yy} = I_2 \cos\theta \tag{15d}$$

Multiplying Eq. (15a) by $\sin\theta$, Eq. (15b) by $\cos\theta$, and subtracting one from the other, leads to

$$\sin\theta\cos\theta(I_{xx} - I_{yy}) = (\cos^2\theta - \sin^2\theta)I_{xy} \tag{16}$$

This equation can be solved for the desired angle θ in terms of the given inertias I_{xx}, I_{yy} and I_{xy}:

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2I_{xy}}{I_{xx} - I_{yy}}\right) \tag{17}$$

When numerically evaluating the $\tan^{-1}()$ function, it is important to use the `atan2()` function which takes both the numerator and denominator as arguments. Doing so will then return the angle θ in the proper quadrant and without singularities.

Given the principal axis orientation angle θ, it is now trivial to compute the desired eigenvectors (principal axis unit direction vectors) using:

$$\boldsymbol{v}_1 = \begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix} \tag{18a}$$
$$\boldsymbol{v}_2 = \begin{pmatrix}-\sin\theta \\ \cos\theta\end{pmatrix} \tag{18b}$$

Note that the eigenvector directions are only unique to within a sign. Both θ and $\theta + \pi$ would yield the proper eigenaxis orientation angle. The snake algorithm currently keeps track of the previous primary eigenaxis angle and make sure that no switching occurs between the two possible solutions.

C. Determining the Principal Axis Dimensions

Given the principal inertias I_1 and I_2 of the shape being tracked by the snake, it is possible to estimate some shape dimensions if we can assume that the target is of a particular shape. For example, let the target area be a rectangular shape with a half-height h and half-length l. Note that we are assuming here that $l \geq h$. The principal inertias I_i are then related to the box dimensions h and l through

$$I_1 = \frac{A}{3}l^2 \tag{19}$$
$$I_2 = \frac{A}{3}h^2 \tag{20}$$

where $A = 4lh$ is the box area. These two equations can be trivially solved for the desired box dimensions:

$$l = \sqrt{\frac{3I_1}{A}} \tag{21}$$
$$h = \sqrt{\frac{3I_2}{A}} \tag{22}$$

If the true target shape is not a perfect rectangle, then this routine will approximate the equivalent box dimensions by using the principal inertias. This is important when tracking non-collaborative visual targets with unknown shapes.

The computed principal axis still provide information about the apparent image size, which can be used to estimate the depth of the target.

Figure 1 illustrates the rectangular box dimensions being estimated from the snake area principal inertias. The primary inertia axis is shown in red, while the secondary inertia axis is shown in blue. The target center of mass is highlighted by a green circle. The yellow arc on the green circle illustrates the heading angle θ. Note that despite the snake rounding off the box corners, the box dimensions are estimated rather accurately to within an image pixel. This computation is robust to small shape errors because the area integral is only secondarily affected by these snake-tracking errors.

This algorithm can be modified for any pre-defined two-dimensional shapes which are parameterized through two coordiinates, as long as an analytical solution exists to extract the shape parameter from the principal inertias. If the target shape is assumed to be nearly elliptical, then the semi-major axis a and semi-minor axis b are computed using

$$a = \sqrt{\frac{4I_1}{A}} \tag{23}$$

$$b = \sqrt{\frac{4I_2}{A}} \tag{24}$$

where $A = ab\pi$ is the ellipse area.

Thus, besides tracking the visual target centroid, the visual snakes allow us to track other features such as the principal axis or target area. This information can be used to determine further relative motion information beyond the centroid heading. The following section discusses how the principal image axis are used to estimate depth/camera parameters on the fly.

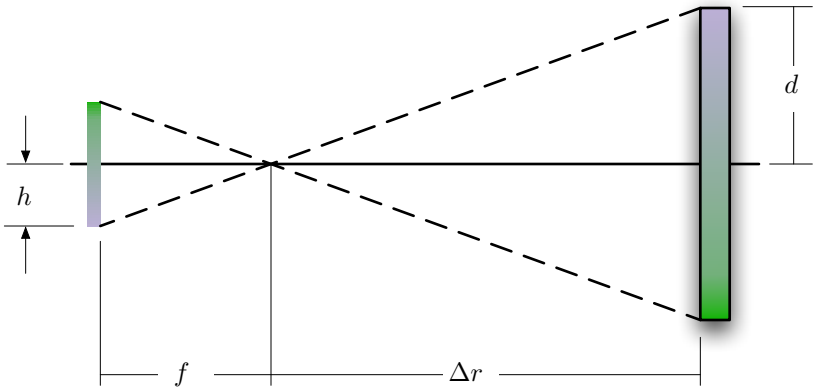


Figure 2: Pin-Hole Camera Illustration

IV. Tracking Law with Time-Dependent Gain

Assume a craft is to move to a visual target with piecewise-constant or very slowly varying visual features, and stop at a prescribed distance Δr_r. Using the visual snakes as the tracking method, the snake principal axes can be used to determine the target distance.[20] If the target has a known size d and the camera has a known focal length f, then the principal semi-major axis can be used to determine the distance to the target. Consider the simple pin-hole camera model in Figure 2. By measuring the the length of the target principal semi-major axis h, the distance is computed using

$$\Delta r = \frac{df}{h} = \frac{1}{\alpha h} \tag{25}$$

where

$$\alpha = \frac{1}{df} \tag{26}$$

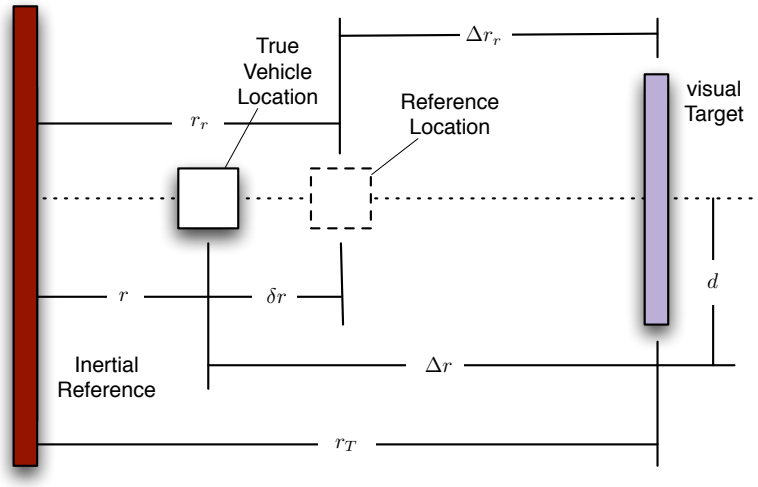


Figure 3: Range Coordinates and Tracking Errors.

is piecewise-constant. Let α be the true value for a given camera and target combination. Using a calibration routine an estimated value $\hat{\alpha}$ can be obtained. The corresponding estimated range to target is

$$\Delta\hat{r} = \frac{1}{\hat{\alpha}h} \tag{27}$$

We assume for now that α is a known constant. Let Δr_r be a desired distance between the camera and the target as shown in Figure 3. The target position relative to an inertial reference frame is r_T, while the inertial camera position is r. The current separation distance and rate between camera and target is

$$\Delta r = r_T - r > 0 \tag{28}$$
$$\Delta\dot{r} = \dot{r}_T - \dot{r} \tag{29}$$

The range tracking error is then defined as

$$\delta r = \Delta r_r - \Delta r \tag{30}$$
$$\delta\dot{r} = \Delta\dot{r}_r - \Delta\dot{r} \tag{31}$$

Thus δr is the visual target tracking error in the separation distance control. The commanded vehicle velocity $\dot{r}$ in Reference 10 is

$$\dot{r} = -\gamma(t) f(\delta r) \tag{32}$$

where $f()$ is an odd, smooth bounding function and $\gamma(t)$ is a time varying feedback gain. Let $c > 0$ be a finite constant value, then $f(\delta r)$ must satisfy

$$\lim_{\delta r \to \pm\infty} f(\delta r) = \pm \mathrm{c} \tag{33a}$$
$$f(0) = 0 \tag{33b}$$
$$f(\delta r)\delta r > 0 \tag{33c}$$

For example, $f(x) = \arctan(x)$ could be used. For small tracking errors, this nonlinear steering law approximates a classical proportional feedback control law, while for large tracking errors a smoothy saturated control signal is produced. Further, the two kinematic control law gains γ and c have simple interpretations. The gain c should be set less than the maximum vehicle velocity, while γ should be set to provide the desired regulator stiffness. Note that having a time-dependent gain $\gamma(t)$ makes the closed-loop dynamics time dependent, and thus the system has become non-autonomous.

To analyze the stability of this kinematic steering law, let us define the Lyapunov function

$$V(\delta r) = \frac{\delta r^2}{2} \tag{34}$$

Because this Lyapunov function does not explicitly depend on time, it is trivial to find time-independent functions $W_1(\delta r)$ and $W_2(\delta r)$ such that $W_1 \leq V \leq W_2$. Thus this is a valid Lyapunov function V for a non-autonomous system.

Let us assume that the target is inertially stationary and $\dot{r}_T = 0$. Taking the time derivative, the Lyapunov rate is

$$\dot{V} = \delta\dot{r}\delta r = -\gamma(t)\,\delta r\, f(\delta r) \tag{35}$$

To show that the steering law $\dot{r}$ in Eq. (32) is asymptotically stabilizing with a non-autonomous system, a positive definite function $W_3(\delta r)$ must be found such that[21]

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial r}\dot{r} \leq -W_3(\delta r) \tag{36}$$

The term $\frac{\partial V}{\partial t}$ is zero because a $V(\delta r)$ is chosen which does not depend on time. If the time-dependent feedback gain satisfies the lower-bound condition

$$\gamma(t) \geq \gamma_0 > 0 \tag{37}$$

then setting

$$W_3(\delta r) = \gamma_0\,\delta r\, f(\delta r) \tag{38}$$

will satisfy the condition in Eq. (36), and the proposed range steering law $\dot{r}$ is asymptotically stabilizing. The gains can vary with time, and convergence is guaranteed as long as the gain never becomes zero.

To implement this control, the true α parameter is not available. Using the estimated $\hat{\alpha}$ value, the steering law then becomes

$$\dot{r} = -\gamma(t) f\left(\frac{1}{\hat{\alpha}h} - \Delta r_r\right) \tag{39}$$

If $\hat{\alpha} \neq \alpha$, then the steering law will not yield asymptotic convergence. The control is still stabilizing, but only the *estimated* tracking errors $\delta\hat{r} = \Delta r_r - \Delta\hat{r}$ will go to zero, not the actual tracking error δr.

V. Parameter Estimation

If the target has an unknown or changing physical size, or the camera focal length is not precisely known, we would like to be able to estimate the parameter α on the fly. Note that this process assumes that the physical target size is piecewise-constant throughout the vehicle motion. If the target is a flat rectangle, for example, then the target image area and principal axes will depend on the viewing angle. It is assumed here that the craft will perform a straight-on approach where the heading angle changes relative to the target will be small. In this case, image foreshortening effects are negligible since they depend on second order terms of the heading miss-alignments.[20]

To estimate the target size/focal length relationship α, the Kalman update equation is used.[22] This is essentially a Kalman filter where no parameter dynamics are present (i.e. $\dot{\alpha} = 0$).

Let $\boldsymbol{y}$ be the vector of M measured states, while $\boldsymbol{x}$ is the vector of N estimated parameters. The following notation will be used. The $\hat{\boldsymbol{x}}$ symbol is used for estimated states, while hatless symbols $\boldsymbol{x}$ are the true states. The right subscript denotes the current discrete time value of this estimate, while the right superscript $\pm$ symbol indicates if a measurement correction has been applied (+) or not (-). Thus, $\boldsymbol{x}_k^-$ is the estimate of the state vector $\boldsymbol{x}$ at time step k before any measurement corrections were applied.

To estimate the parameter α, with a slight abuse of notation we set

$$\boldsymbol{x} = \alpha \tag{40}$$

To estimate α using the Kalman update equation, we need α to appear linearly in our measurement vector. The target principal semi-major axis d is measured through the image length h. This measurement and α are related through

$$\frac{1}{h} = \alpha\Delta r \tag{41}$$

where the desired parameter α appears linearly. If the distance Δr between the vehicle and the target were known, then this equation could be used as the measurement equation. However, the actual distance to the target is an unknown state that we are attempting to determine using the visual snake information. Instead of using Eq. (41) directly, let us write this equation at two different time steps k and $k-1$:

$$\frac{1}{h_k} = \alpha \Delta r_k \tag{42a}$$

$$\frac{1}{h_{k-1}} = \alpha \Delta r_{k-1} \tag{42b}$$

If we write the M-dimensional measurement vector $\boldsymbol{y}_k$ as

$$\boldsymbol{y}_k = \frac{1}{h_k} - \frac{1}{h_{k-1}} = \alpha \left(\Delta r_k - \Delta r_{k-1} \right) \tag{43}$$

then we no longer need to know the actual distance to the target. Instead, the *distance traveled toward the target* $\Delta r_k - \Delta r_{k-1}$ must be known. This information can be obtained from an IMU, or using the vehicle wheel encoders along with dead-reckoning. A similar technique of extracting the camera/target parameters by performing known motions is also discussed in Reference 23 where it is applied to the visual 6 degree-of-freedom control of a manipulator end effector. For the visual control of the unmanned ground vehicle, it is of interest if the wheel encoder information will be sufficient to effectively estimate the α parameter. The encoder derived vehicle motion will not be precise, but should provide a reasonable estimate of the distance traveled. The goal of this control is not to achieve very high positioning accuracy, but rather, to achieve a robust autonomous strategy which will reposition the vehicle in an unstructured environment. Once the vehicle arrives in the proximity of the target, a human operator can provide further guidance.

To use the Kalman update equations to estimate α during the vehicle motion, the following equations are used. Let use define the $M \times N$ matrix $[H]$ as

$$[H_k] = [\Delta r_k - \Delta r_{k-1}] \tag{44}$$

To current observation vector is then

$$\boldsymbol{y}_k = [H_k] \boldsymbol{x}_k \tag{45}$$

Given a current state estimate $\hat{\boldsymbol{x}}_k^-$, applying the measurement update yields the corrected estimate:

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + [K_k] \left(\boldsymbol{y}_k - [H_k] \hat{\boldsymbol{x}}_k^- \right) \tag{46}$$

where $[K_k]$ is the current Kalman gain matrix

$$[K_k] = [P_k^-][H_{k]}^T \left([H_k][P_k^-][H_k]^T + [R_k] \right)^{-1} \tag{47}$$

The $N \times N$ matrix $[P_k^-]$ is the current, un-updated covariance matrix of the estimated states $\hat{\boldsymbol{x}}_k$, while the $M \times M$ $[R_k]$ is the covariance matrix of the measurement $\boldsymbol{y}_k$. To update the covariance matrix $[P_k]$ we use[22]

$$[P_k^+] = \Big([I_{N \times N}] - [K_k][H_k] \Big) [P_k^-] \tag{48}$$

With a Kalman filter, the states would be updated between measurements using their differential equations. However, in this piecewise-constant parameter estimation problem, there are no differential equations. The only quantity that must be updated between measurements is the covariance matrix. Let the $N \times N$ matrix $[Q_k]$ be the current process noise matrix. The covariance matrix $[P_k]$ is then updated using

$$[P_{k+1}^-] = [P_k^+] + [Q_k] \Delta t \tag{49}$$

where Δt is the time between measurement updates. With a non-zero $[Q_k]$, the uncertainty of the current estimate $[P_k]$ will increase over time without measurement updates.

VI. Hardware Simulation

The following hardware implementation is considered. The feedback gain $\gamma(t)$ is set to be dependent on the covariance matrix value. The higher the covariance P (state estimation uncertainty), the lower the feedback gain should be. Thus, if the vehicle is very uncertain about its α estimation, then it proceeds with caution. Once the confidence level of the $\hat{\alpha}$ value increases, so does the feedback gain and the vehicle response will be more aggressive. The time dependent gain value is

$$\gamma(t) = \tilde{\gamma}\left(1 - \kappa\frac{P}{P_{\text{max}}}\right) \tag{50}$$

where P_{max} is a maximum allowable covariance value, $\tilde{\gamma}$ is the nominal feedback gain value if the $\hat{\alpha}$ estimation has a high confidence value, and $0 < \kappa < 1$. Note that this $\gamma(t)$ definition satisfies the gain stability condition in Eq. (37). The larger κ is, the more conservative the steering law will perform if the target distance uncertainty is large.

To initialize the visual servo control, the $\hat{\alpha}$ covariance P is set to P_{max} because the target size/focal length parameter is not known initially. $\hat{\alpha}$ is initially set to

$$\hat{\alpha} = \frac{5}{6}\frac{1}{\Delta r_r h} \tag{51}$$

so that the control's initial range estimation given by Eq. (27) is

$$\Delta\hat{r} = \frac{6}{5}\Delta r_r \tag{52}$$

This visual servoing procedure is initialized when the human operator double-clicks on the target in the screen. Assuming the target size is unknown, the vehicle will not know if this is a large target far away, or a small target up close. The outlined automatic initial parameter $\hat{\alpha}$ algorithm makes the craft assume that the range to target distance is only 20% too large. This ensures that the control begins to move toward the target cautiously. As the vehicle moves and the $\hat{\alpha}$ parameter is estimated more accurately, the vehicle quickly learns how far away the target is.

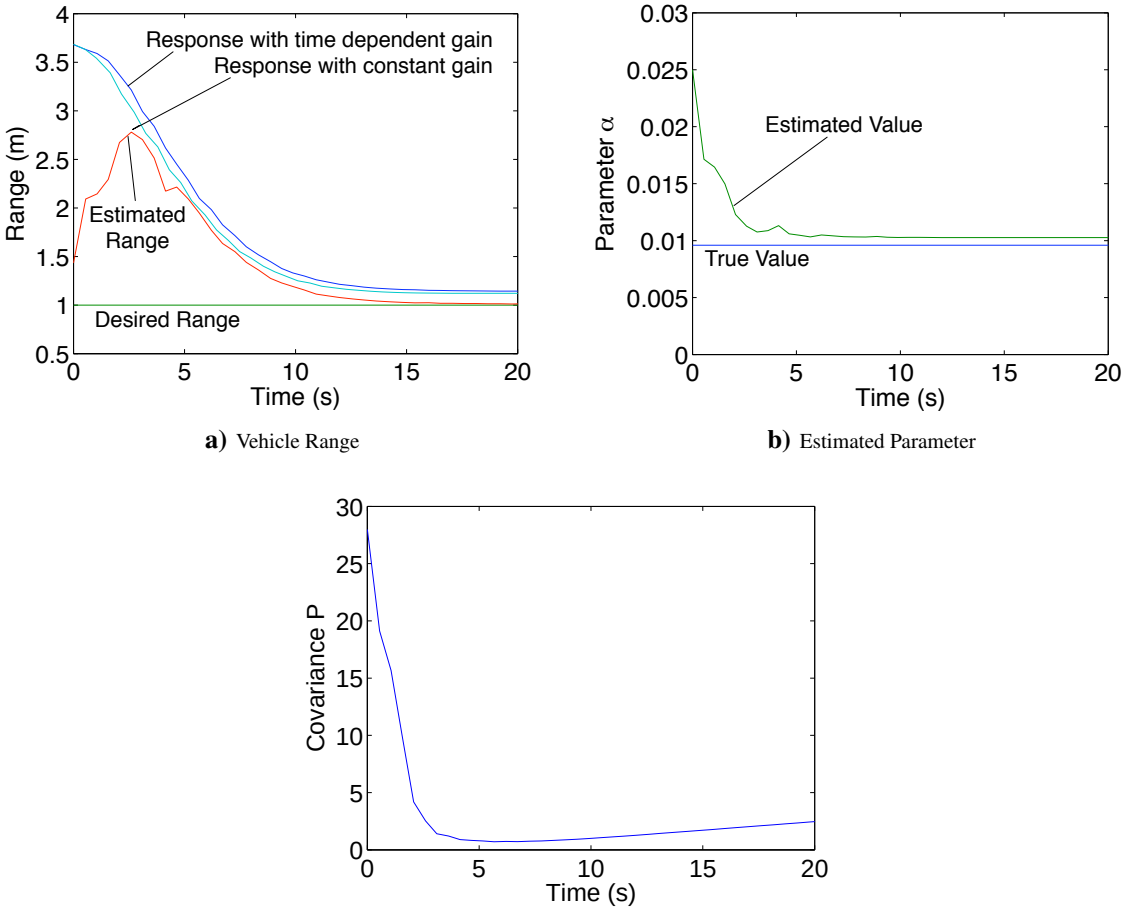
Several cases are considered. These involve approaching a target that may change in size from various initial ranges. However, in all cases, the same parameters are used, and are shown in Table 1. The ability to achieve reasonable performance while using the same parameters in widely varying test cases shows the performance robustness of the demonstrated sensing and control techniques.

Table 1: Simulation Parameters

Description	Value
Δr_r	1.00 meters
P_{max}	30.00
Q_k	0.15
R_k	0.15
κ	0.75
$\tilde{\gamma}$	500
c	1000

A. Long Range Test

In this case, the vehicle starts from a range of 3.7 meters and approaches a target that remains a constant size throughout the test. Figure 4 illustrates the results of this test case. If a constant feedback gain with a known α is used, then the range will asymptotically approach the desired range throughout the maneuver. However, if the α parameter is not well known, then such aggressive behavior will not be practical. Instead, the initial feedback errors are small because of the initialization of α, and initial feedback gains are small because of the covariance-dependent feedback gain. This results in the vehicle approaching the target slowly and cautiously at first. As the vehicle begins to move, the estimate of the range $\Delta\hat{r}$ improves, and the covariance of the estimated $\hat{\alpha}$ value decreases, resulting in a larger feedback gain. The resulting error dynamics after a few seconds of motion again resemble an exponential decay. Figure 4(a) also shows the performance of the covariance dependent gain compared to a constant gain. The



a) Vehicle Range

b) Estimated Parameter

c) State Covariance Estimation

Figure 4: Range Regulator Maneuver Illustration

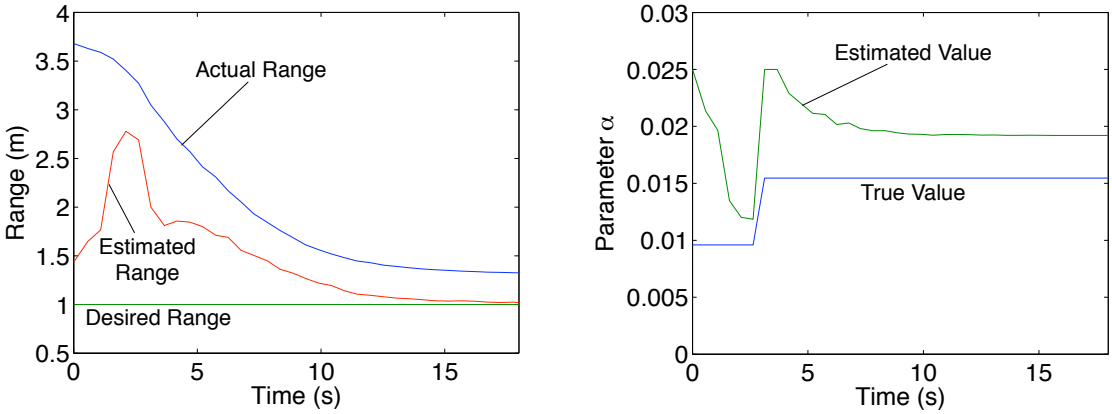
time dependent gain slows the initial vehicle response, resulting in a more cautious behavior as the vehicle begins to move.

Note that $\hat{\alpha} \not\to \alpha$ as $t \to \infty$. This estimation technique does not provide any parameter convergence guarantees. The final range tracking errors don't converge to zero. But, because the $\hat{\alpha}$ estimate is close to the actual α, this tracking error is small. However, the automatic $\hat{\alpha}$ initialization routine does provide the desired performance. Without requiring stereo vision systems, the vehicle is able to move up to the target of unknown size and stop near the desired separation distance.

The performance of these sensing and control techniques in the above case is quite good when the desired target range is relatively small ($<$ 3 m.). However, at large target ranges ($>$ 5 m.), limitations arise, and the control is sensitive to noise in the estimated parameter $\hat{\alpha}$ and the target principal axes measurements. The exact limitations are determined primarily by the target size and camera focal length, *f*. In these experiments, the target semi-major and semi-minor axes were 15 and 8.5 cm., respectively, and the camera focal length f = 4.2 mm. Future work will attempt to make this control strategy more robust at long ranges.

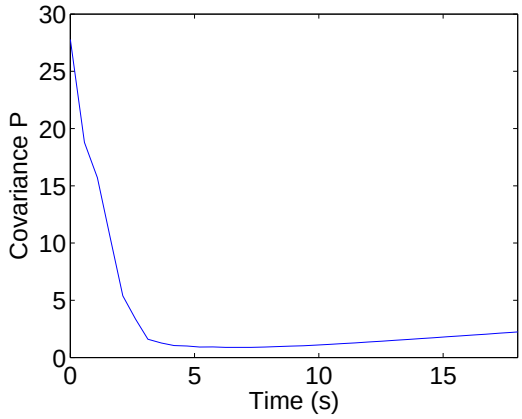
B. Changing Target Size

In this case, the vehicle again starts at a range of 3.7 meters, but this time there is a step-change in target size during the test. As seen in Figure 5, the general behavior in both of the fixed-target-size segments matches that seen in the simpler fixed target size case. At 2.6 seconds, the target area decreases by 45%. This very drastic scenario represents part of the target being obscured during the test, or having the target rotated quickly. Because the target size is instantly reduced, without any estimation of time varying parameters, the vehicle would assume that the target range error drastically increased. As a result, it would rush forward and possibly collide with the target. However, with the $\hat{\alpha}$ estimation active, the vehicle is able to automatically tune its depth gain calculation in real time during the maneuver.



a) Vehicle Range

b) Estimated Parameter



c) State Covariance Estimation

Figure 5: Range Regulator Maneuver With Changing Target Size.

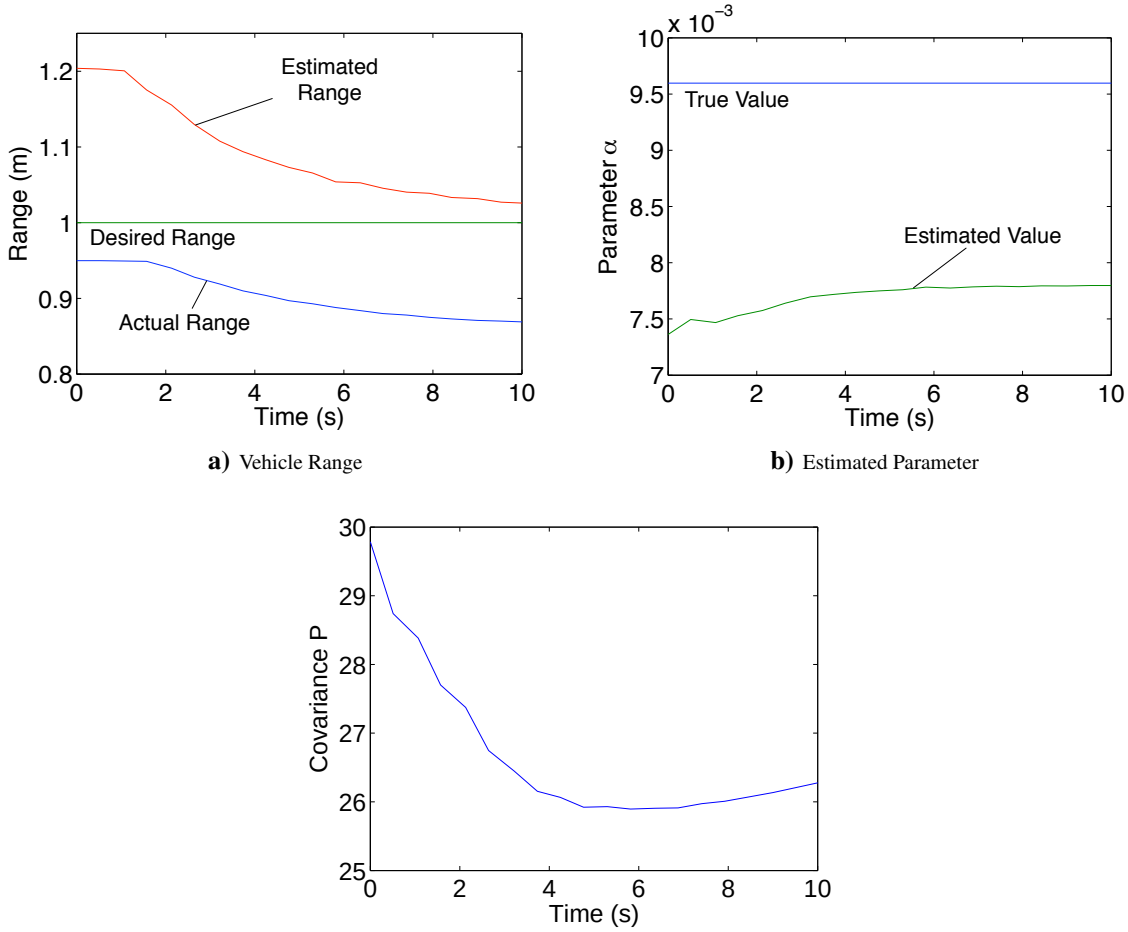
The craft does speed up slightly as the target size is changed. However, the craft also recomputes a new $\hat{\alpha}$ estimate which ensures that it does not run into the target. Due to the short range left to maneuver to the desired target range, the new $\hat{\alpha}$ estimation converges to a value with a larger error compared to the true value. However, the final range error is still acceptable, and it is conservatively further away from the target, rather than closer. Again, this control does not guarantee convergence of $\hat{\alpha}$. However, it does illustrate sophisticated and robust visual servoing capabilities in real-world applications.

C. Close Range Test

In this case, the vehicle starts at a range of 0.95 meters, which is closer to the target than the desired 1.0 meter range. Because $\hat{\alpha}$ is initially selected so that the estimated range is slightly greater than the desired range, the vehicle will initially approach the target even though it is already inside the desired range. It is critical that the vehicle not approach too close to the target. Figure 6 shows the results from this test. While the vehicle does move closer to the target, it does not exhibit dangerous behavior near the target or approach too close. $\hat{\alpha}$ is approaching α, but because the system is weakly observable for such small camera motions, the system is unable to obtain a very good estimate. However, even in this challenging case, the control results in reasonable vehicle motions with the desired conservative behavior.

VII. Conclusion

A visual target tracking control is presented which incorporates a Kalman update equation to obtain an estimate of the target range during the maneuver. This self-tuning depth parameter makes the control more robust to target size changes, as well as simpler to implement. No depth gain calibration is used. This parameter is tuned automatically during the maneuver without requiring stereo vision cameras. The time varying feedback gain depends on the estimator



a) Vehicle Range

b) Estimated Parameter

c) State Covariance Estimation

Figure 6: Range Regulator Maneuver From Close Range

covariance and will reduce the vehicle's feedback gains when it has little confidence in its range estimates. Using only a single camera as opposed to a stereo system, this control will allow a UGV to approach a visual target of unknown size and stop a prescribed distance away. The control does require some knowledge of the inertial vehicle motion, which could be obtained from an IMU or the wheel encoders. The system modeled in these hardware experiments is a free-floating spacecraft approaching a visual target. All craft are assumed to be in inertial, non-orbiting environments, such as a scout craft controlling its relative motion with respect to an interplanetary craft. Future work will expand this strategy to incorporate time varying targets positions, as well as orbiting environments.

Acknowledgment

This research was supported by Sandia National Laboratories under Contract Number # 307618. We would also like to thank ORION International Technologies for providing a license to the UMBRA software framework.

References

[1]Junkins, J. L., Hughes, D. H., Wazni, K. P., and Pariyapong, V., "Vision-Based Navigation for Rendezvous, Docking and Proximity Operations," *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 3–7 1999, Paper No. AAS 99-021.

[2]Ivins, J. and Porrill, J., "Active Region Models for Segmenting Medical Images," *Proceedings of the IEEE International Conference on Image Processing*, Austin, TX, 1994, pp. 227–231.

[3]Perrin, D. and Smith, C., "Region-Based Strategies for Active Contour Models," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.

[4]Schaub, H. and Smith, C. E., "Color Snakes for Dynamic Lighting Conditions on Mobile Manipulation Platforms," *IEEE/RJS International Conference on Intelligent Robots and Systems*, Las Vegas, NV, Oct. 2003.

[5]Smith, C. E. and Schaub, H., "Efficient Polygonal Intersection Determination with Applications to Robotics and Vision," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, Aug. 2–6 2005.

[6]Philip, J., "Estimation three-dimensional motion of rigid objects from noisy observations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 1, 1991, pp. 61–66.

[7]Singh, A., "Incremental estimation of image-flow using a Kalman filter," *IEEE Proceedings of the IEEE Workshop on Visual Motion*, Vol. 9, No. 9, 1991, pp. 36–43.

[8]Spetsakis, M.E.; Aloimonos, J., "Optimal motion estimation," *IEEE Proceedings of the IEEE Workshop on Visual Motion*, Vol. 3, No. 3, 1989, pp. 229–237.

[9]Matthies, L., Szeliski, R., and Kanade, T., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences," Tech. Report CMU-RI-TR-88-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1988.

[10]Schaub, H., "Visual Servoing Using Statistical Pressure Snakes," Technical Report SAND2004-1868, Sandia National Laboratories, Albuquerque, NM, 2004.

[11]Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: active contour models," *International Journal of Computer Vision*, Vol. 1, No. 4, 1987, pp. 321–331.

[12]Perrin, D. P., Ladd, A. M., Kavraki, L. E., Howe, R. D., and Cannon, J. W., "Fast Intersection Checking for Parametric Deformable Models," *SPIE Medical Imaging*, San Diego, CA, February 12–17 2005.

[13]Perrin, D. and Smith, C., "Rethinking Classical Internal Forces for Active Contour Models," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, Dec. 8–14 2001, pp. 615–620.

[14]Malladi, R., Kimmel, R., Adalsteinsson, D., Sapiro, G., Caselles, V., and Sethian, J. A., "A geometric approach to segmentation and analysis of 3D medical images," *Proceedings of Mathematical Methods in Biomedical Image Analysis Workshop*, San Francisco, June 21–22 1996.

[15]Schermann, E. and Ebersole, T., "An Implementation of Color Statistical Pressure Snakes using HSV Color Space and Redefined Internal Energy Terms," CS 766 Project.

[16]Schaub, H., "Statistical Pressure Snakes based on Color Images," Technical Report SAND2004-1867, Sandia National Laboratories, Albuquerque, NM, 2004.

[17]Schaub, H. and Wilson, C., "Matching a Statistical Pressure Snake to a Four-Sided Polygon and Estimating the Polygon Corners," Technical Report SAND2004-1871, Sandia National Laboratories, Albuquerque, NM, 2003.

[18]Steger, C., "On the Calculation of Moments of Polygons," Technical Report FGBV–96–04, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München, Aug. 1996.

[19]Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, AIAA Education Series, Reston, VA, October 2003.

[20]Schaub, H., "Extracting Primary Features of a Statistical Pressure Snake," Technical Report SAND2004-1869, Sandia National Laboratories, Albuquerque, NM, 2004.

[21]Khalil, H. K., *Nonlinear Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 3rd ed., 2002.

[22]Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic System*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[23]Papanikolopoulos, N. P., Nelson, B. J., and Khosla, P. K., "Six Degree-of-Freedom Hand/Eye Visual Tracking with Uncertain Parameters," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, 195, pp. 725–732.