



AIAA 2005-5999

Hardware Simulation of Relative Navigation using Visual Sensor Information

Mark J. Monda and Hanspeter Schaub

Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0203

**AIAA Guidance, Navigation and
Control Conference**

August 15–18, 2005 / San Francisco, CA

Hardware Simulation of Relative Navigation using Visual Sensor Information

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 that has many possible applications including formation operation, 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 is used to measure the relative position, orientation, and motion between the vehicle and the target. Color statistical pressure snake algorithms are employed to track a visual target within the camera images in real time. The effectiveness of the nonlinear visual servoing algorithm is demonstrated through experimental hardware tests.

I. Introduction

Future generations of unmanned ground, air, and space vehicles offer many benefits over their manned counterparts, including reduced size, cost, complexity, and danger for the human operators. However, to be feasible replacements for current manned versions of such vehicles, future autonomous vehicle control systems must be able to manage the details of vehicle operation and control with minimal operator intervention. For example, an operator might instruct a formation of unmanned ground vehicles to move to a new location, and the vehicles would then autonomously navigate towards the destination, while maintaining formation and avoiding obstacles along the way. In short, they would perform all tasks required to complete the objective without additional commands from the operator.

However, before such levels of autonomy can be practically realized in a wide range of applications, the vehicle control systems' sensing capabilities must be improved. One area in which current sensing can be improved is the analysis of visual information. While humans rely heavily on vision for information about the surrounding environment, it has proven difficult to endow autonomous control systems with the same capability. However, visual sensing is a valuable ability because it can convey precise relative information. For example, when performing an aerial refueling operation, the pilot's vision gives him extremely accurate information about how he must maneuver the aircraft relative to the tanker. Similarly, multi-ground vehicle formation operation as well as UAV and spacecraft formation flight, rendezvous, and docking are other examples of relative navigation problems in which precise relative information from visual sensors would be extremely valuable.¹

The performance of visual sensors can be further improved by the use of appropriate estimation algorithms. Applying techniques such as Kalman filtering, unscented Kalman filtering, or particle filtering to visual sensor output data can offer many benefits including smoother, more continuous sensor information with lower noise and error levels.²

Over the last few decades several types of spacecraft simulators have been developed. Planar motion systems typically involve two translational and one rotational degree of freedom and are of interest to rendezvous and docking operations, as well as flexible manipulator research. Reference 3 contains an excellent survey of spacecraft simulators.

*Graduate Student, Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute

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

Simulators discussed in References 4, 5 and 6 float test hardware on polished flat surfaces. Here the planar frictionless space environment is used to evaluate force-free motion of flexible structures.

The new test bed being developed in the Autonomous Vehicle Systems (AVS) Laboratory at Virginia Tech differs from such frictionless test beds in that the physical spacecraft motion will be simulated through an autonomous ground vehicle. The purpose and goal of the proposed test bed is not to mimic frictionless planar motion, which is only an approximation of the coupled orbital motion a satellite would actually experience in flight. Rather, the UGV will have the motion prescribed through numerical software modules, much like a modern aircraft flight simulator. This motion modeling aspect is similar to the large Navy Research Laboratory spacecraft simulation facility where the 6 degree of freedom motion of a full-size spacecraft is simulated using a gantry crane.⁷ The goal is to provide the sensor platform with realistic physical motion. Further, using UGVs allows for relative motion tests to be conducted both indoors and outdoors, over large and small ranges. Such flexibility is impossible with hover tables or large crane structures.

The Marshall Space Flight Center's Flight Robotics Laboratory⁸ is a very sophisticated testing facility which allows different test hardware to interact through networking, similar to what is planned with this new test bed. The AVS test bed is much smaller in scale and differs in the use of unmanned ground vehicles to simulate vehicle motion, providing a broader base of applications ranging from underwater, ground, aerial and space-based vehicles. Further, the use of a software framework called UMBRA⁹ will ensure compatibility with other research facilities such as Sandia National Laboratories.

The initial goal of this test bed is to simulate the visual servoing of aerospace vehicles in near-planar motion. Statistical pressure snakes (also known as visual snakes) are used to visually track a target in a video stream. Related vision-based space-based robotic control is discussed in References 10 and 11. Here zero-gravity experiments are performed in a neutral buoyancy water tank. While providing 6 degrees of freedom, the range of this test bed is limited to several meters. Further, the visual tracking employs statistical sampling of the entire image for target colors. Such methods provide fast tracking if the target is the only object present with this color. The AVS test bed will provide a greater range of relative motion that can be simulated, and better lighting control by avoiding the need to suspend the test craft in water. Further, the visual snakes are more robust than statistical sampling in the presence of non-unique target colors. Also, the visual snake will provide a one-dimensional contour of the target which provides additional information about it.

An example of real-time visual control of ground vehicles is Reference 12 and 13. A modular test setup is created with an omnidirectional camera displaying sophisticated formation control. Targets are identified through statistical color models in YUV color space for robustness to lighting variations, similar to the proposed HSV use of visual snakes. However, image gradients are used to determine objects of interest, assuming a flat mono-color ground. The proposed snake-based visual control will investigate operating in unstructured environments without such assumptions.

This paper describes a hardware test bed which is being developed to investigate the performance of visual sensing and estimation algorithms in different types of relative navigation problems. This test bed consists of unmanned mobile robotic ground vehicles with onboard computers and visual sensors. The test bed can be used to simulate near-planar relative navigation problems for underwater, ground, air, and space vehicles. By programming the motion of the test bed vehicles to mimic the relative motion that would be encountered by the actual vehicles, the performance of the visual sensors can be tested in real-world test conditions. Section II. discusses the test bed setup, current status, and included hardware capabilities. Section III. discusses the visual sensing algorithm which has been implemented in real-time on the unmanned ground vehicle. Section IV. discusses the visual control strategy to regulate a visual tracking error. Finally, section V. illustrates hardware test results of the current test bed setup.

II. Hardware Test Bed Description

A. Simulation Setup

Consider a free-flying robotic assistant maintaining its position relative to the International Space Station by looking at simple visual features on the station's structure. Without requiring complex radio or optical beacons, the robot is able to maintain its position using only a stream of images from a video camera. Figure 1 illustrates such a concept. Developing this system is a very complex and challenging undertaking. In particular, for this space-based visual servoing example, performing hardware proof-of-concept tests will include simulating the complex relative orbital motion, simulating the three-dimensional attitude, as well as duplicating the intricate lighting, reflection and shadow variations. Physically simulating these tasks is a daunting challenge. However, using modern software and communication developments, a novel spacecraft simulation approach is being developed which blends the boundaries between reality and simulated components.

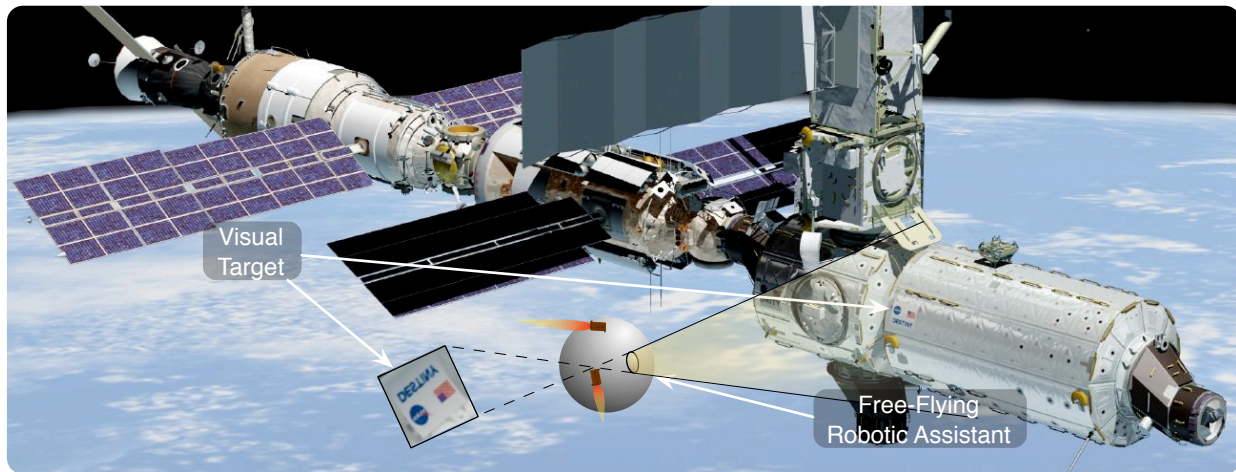


Figure 1: Illustration of a Free-Flying Robotic Assistant Visual Servoing of a Passive Optical Target on the Space Station.

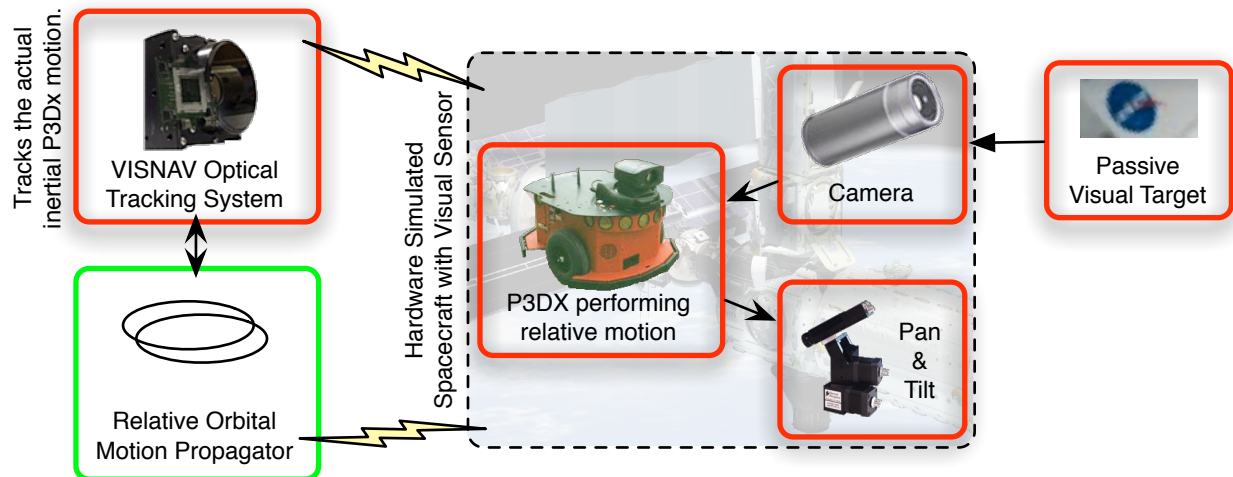
Numerical simulations are commonly employed to model new navigational sensing and control concepts. While such simulations provide important initial studies, hardware tests with critical sensing, communication and mechanical behaviors are typically required to fully develop these sensing and control methods. In practice the complexity of many research projects requires combinations of numerical simulation and hardware testing at an early stage. A hybrid hardware/software distributed simulation environment is being developed to investigate visual servoing of semi-autonomous vehicles.

Instead of a costly test bed which will simulate all environmental, physical, and communication aspects of an autonomous aerospace vehicle, a cost-effective relative motion test bed is being developed for scenarios where the relative motion dynamics are well understood. For example, an unmanned ground vehicle (UGV) is programmed to move relative to a target as if it were in orbit. Other aerospace vehicles could be modeled as well. The UGV acts thus as a sensor platform, providing the relative motion sensor with realistic relative motion. With on-board processing capabilities, the sensor data can be analyzed in real-time and a corresponding control solution computed. The physical response of the modeled aerospace vehicle is then simulated in software and the resulting vehicle relative motion prescribed through dedicated servo routines.

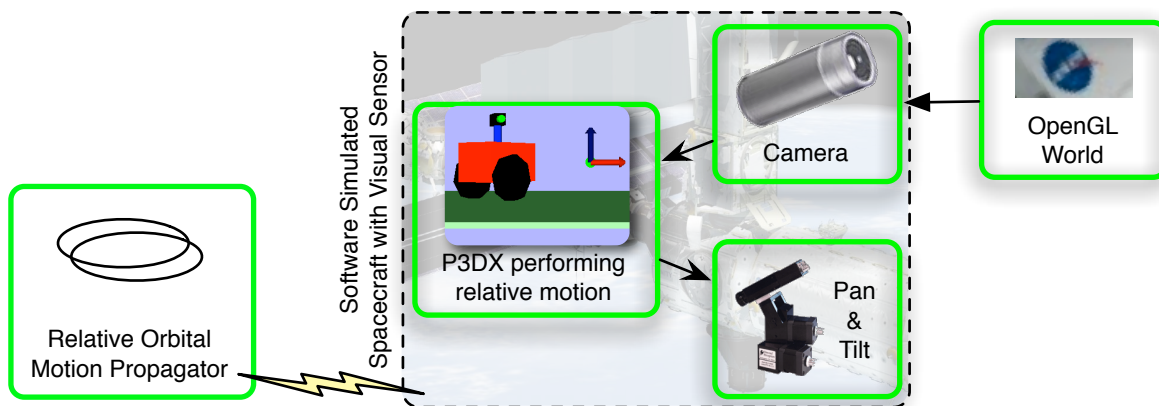
Figure 2(a) illustrates a sample setup where a free-flying robotic assistant visually servoing on another spacecraft is simulated in a hybrid hardware/software scenario. A physical camera is mounted on the UGV, and images are captured with a frame-grabber card. After processing the visual data and computing target tracking errors, the station-keeping control solution is sent to the simulated spacecraft module which computes the relative motion response. The vehicle motion will provide planar (x, y) motion, but cannot provide independent camera heading angles. By adding a pan-and-tilt unit to the vehicle, the camera can be pointed independently from the vehicle heading. Further, using the tilt mode, small out-of-plane attitude motions can also be simulated. To compute the spacecraft orbital response due to a control input, the actual inertial position of the UGV must be known. An optical tracking system called VISNAV¹ provided by StarVision Inc.¹⁴ will be used. This system is capable of tracking active optical beacons both indoors and outdoors at ranges of up to 50 meters.

The current hardware test setup consists of the operational UGV with a camera mounted on it. Elementary wireless communication has been implemented to operate the vehicle untethered. The pan-and-tilt unit has not been installed currently. Also, the relative orbit propagator module has not been implemented. The current hardware setup is capable of visually servoing the UGV on a target without any communication, video or power connections. This wireless operational mode is an important milestone to developing the full aerospace vehicle simulation capability.

Figure 2(b) illustrates a parallel development of this relative motion test bed. The UGV is being modeled through a software module, as well as the camera and hardware pan-and-tilt units. This software simulation of the test bed will enable researchers to first test sensing or control strategies in software before implementing them on hardware. A key component of this test bed development is the use of the Sandia National Laboratories developed UMBRA framework.⁹ UMBRA allows diverse behaviors such as vehicle dynamics, sensor modeling, communication infrastructures, vehicle path planning, etc., to be encapsulated in compiled modules. These modules can then be invoked during run-time and interact with other modules through flexible connection protocols. UMBRA modules have been created to encapsulate



a) Physical Vehicle Simulating Relative Motion of Aerospace Vehicle



b) Software Simulated Vehicles Servoing in a Virtual Environment

Figure 2: Test-Bed Illustration Where an Autonomous Ground Vehicle is Used to Mimic Near-Planar Spacecraft Motion Relative to a Target Craft.

the UGV control, video frame-grabber, visual processing routine, as well as the visual servo control computation. As a result, the exact same visual servo module can control either the physical or simulated vehicle by redirecting the servo module output. Hybrid simulation scenarios are already being tested where real camera images are driving virtual vehicles.

B. Autonomous Ground Vehicle

The UGV used in this test bed to investigate visual sensor performance in relative navigation problems is based on the Pioneer 3-DX mobile robotic ground vehicle platform manufactured by ActivMedia Robotics shown in Figure 3. This platform features a serial interface which can be commanded from an onboard PC-104 computer, or an externally mounted computer. Several types of sensors interface directly with the vehicle's control system, and are then accessed through the serial communication protocol. These include encoders on each of the two independently-driven, differential-steer drive wheels, bump sensors, sonar sensors (5 meter maximum range), and a digital 3 axis compass/roll/pitch magnetometer.

The onboard PC-104 computer is based on a Pentium III processor. It is currently equipped with a wireless ethernet card, a frame grabber card, and a USB card. UMBRA modules have been created to send and receive data wirelessly between the Pioneer vehicle and a workstation. It is noted that most every type of computer card is available in the PC-104 form factor, and therefore many additional cards and capabilities could be added (such as a GPS receiver, a data acquisition card, etc.). The PC-104 communicates with the vehicle's control system serially, using the same protocol



Figure 3: ActivMedia Robotics Pioneer 3-DX robotic vehicle with digital camera system

that an external computer would use.

The primary software running on the PC-104 is the UMBRA simulation framework, which is a C++ based, modular, real-time software framework. This simulation framework was developed at Sandia National Laboratories to model the interaction of complex systems. It is well suited for applications in which multiple modules must share data and operate in real-time.⁹ Essentially, Umbra acts as a front-end for all of the onboard computer functions, with each encapsulated as a module. The serial communication with the vehicle, the visual sensing and estimation, the vehicle dynamics, and the vehicle control inputs are all encapsulated as individual modules. Using this UMBRA framework, hybrid hardware-in-the-loop simulation scenarios are being investigated. The current setup has a Pioneer UMBRA module which interacts with the P3-DX vehicle and engages the various servo modes. Further, an UMBRA module was created to interact with the PC-104 frame grabber card. The output of this module is an image structure which is used by the visual processing algorithm. Further, various UMBRA modules to monitor and test the P3-DX hardware modes are being developed. The hardware results presented later in this paper should be considered early initial results, and are not the final test bed capability that is planned.

Using the wireless LAN card on the PC-104 computer, the onboard computer can be networked with any wireless Ethernet network. This allows all of the test bed vehicles to communicate with each other or with a base workstation. In addition, by using a remote administration program, the PC-104 can be controlled and operated wirelessly from a base station computer. This remote access capability greatly facilitates testing and operation of the vehicle.

The pan-and-tilt unit is currently only controlled by a workstation and is not mounted yet on the P3-DX. However, an UMBRA interface module has been created here as well to direct the heading of this device. Using the pan-and-tilt unit along with the P3-DX degrees-of-freedom, the near-planar motion of spacecraft will be simulated. When performing rendezvous and docking operations in space, or performing general close proximity operations, it is the orbit plane motion that is strongly coupled. The hardware described in this section is sufficiently close to near-planar motion with two translational and orientational degrees of freedom. Future test bed expansions could rotate and/or lift the camera to provide additional relative motion simulation capabilities.

Sensing the actual inertial motion of the P3-DX is crucial if the vehicle is to simulate the motion of an aerospace vehicle. For example, the spacecraft relative motion dynamics depend on the location of the spacecraft relative to the target. The P3-DX is capable of adding differential GPS receiver units. However, such motion sensors would only operate outdoors. Instead, the first inertial motion sensing installation will use the VISNAV sensors made by StarVision.¹ A position sensitive photo-detector is used to detect incoming light and estimate a relative position and direction vector. Compared to using charge-coupled device (CCD) cameras at only tens of Hertz, update rates of hundreds of Hertz are possible.

III. Visual Sensing using Statistical Pressure Snakes

The visual sensor used in these relative navigation problems consists of a digital camera, the frame-grabber card, and the on-board PC-104 computer. The computer processes the images using a recent real-time statistical pressure snake algorithm.¹⁵ 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.¹⁶ Also

referred to as a visual snake, the parametric curve is of the form

$$S(u) = I(x(u), y(u))', \quad u = [0, 1] \quad (1)$$

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 energy function is expressed as:¹⁶

$$E = \int_0^1 [E_{\text{int}}(S(u)) + E_{\text{img}}(S(u), I)] du \quad (2)$$

where E_{int} is the internal energy defined as

$$E_{\text{int}} = \frac{\alpha}{2} \left| \frac{\partial}{\partial u} S(u) \right|^2 + \frac{\beta}{2} \left| \frac{\partial^2}{\partial u^2} S(u) \right|^2 du \quad (3)$$

and E_{img} is the image pressure function. The free weighting parameters α and β enforce tension and curvature requirements of the curve $S(u)$.

The active deformable models can be divided into two groups:¹⁷ parametric models (snakes)^{16,18} and level-set models (geometric contours).¹⁹ 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.²⁰ 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 new energy function is given by

$$E = \int_0^1 [E_{\text{int}}(S(u)) + E_{\text{pres}}(S(u))] du \quad (4)$$

where the pressure energy function E_{pres} is

$$E_{\text{pres}} = \rho (\partial S / \partial u)^\perp (\epsilon - 1) \quad (5)$$

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.¹⁸ This simplified formulation includes an even snake point spacing constraint. The resulting algorithm does not contain the difficult to tune tension and curvature forces 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.²¹ Further, to provide robustness to lighting variations, Schaub and Smith propose a new image error function:²²

$$\epsilon = \sqrt{\left(\frac{p_1 - \tau_1}{k_1 \sigma_1}\right)^2 + \left(\frac{p_2 - \tau_2}{k_2 \sigma_2}\right)^2 + \left(\frac{p_3 - \tau_3}{k_3 \sigma_3}\right)^2} \quad (6)$$

where p_i are local average pixel color channel values, τ_i are the target color channel values and σ_i are the target color channel standard deviations. The gains k_i are free to chosen. The image RGB colors are mapped into the Hue-Saturation-Value color space illustrated in Figure 4. By choosing appropriate gains k_i , the visual snake can track targets with significant variations in target saturation and shading. The resulting visual snake implementation is referred to as the color statistical pressure snake and is currently operational in the Virginia Tech AVS Lab. The promising tracking performance illustrated in Figure 5 has also been independently verified in Reference 23.

Figure 5 shows two image captures where the same visual snake algorithm is tracking two different types of targets. In Figure 5(a) the operator selected the blue target square and the snake (purple outline) snaps to the visual boundaries. 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. Note that the visual snake routine is not confused by the image being partially

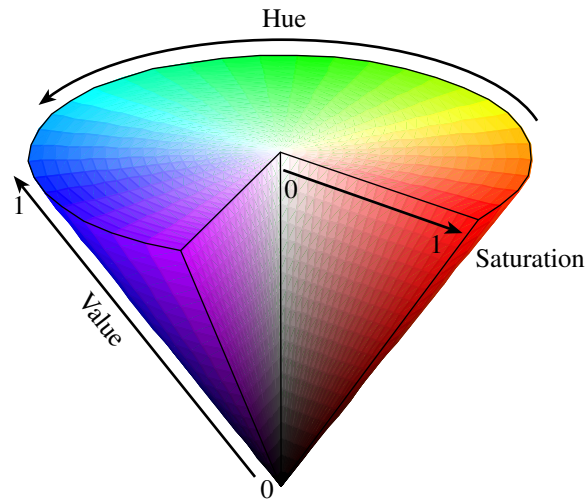


Figure 4: Conic Illustration of the Hue-Saturation-Value (HSV) Color Space.



a) Visual Snake Tracking a Partially Obscured Square Target and Estimating the Corner Locations²⁴ **b)** Visual Snake Tracking a Yellow Suit-Case Outdoors with Severe Lighting Variations¹⁵

Figure 5: Examples of the Identical Visual Snake Algorithm Tracking Different Targets. Each target is selected by double-clicking on it within the image.

obscured by a pen or fingers. A traditional corner or straight-line detection algorithm would have difficulties tracking this target. Further, using the visual snake contour line it is possible to extract additional target features such as target principal image axes direction and size, centroid location,²² and even target corners.²⁴

An open research question involves developing efficient automatic snake parameter k_i selections. The HSV color space has singularities with near-gray-scale colors. The implemented automatic gain selection algorithm provides a reasonable target selection performance. However, further enhancements to this important gain selection process are expected. The current implementation is able to select targets by having the user double-click on the computer screen. This simple activations is promising for applications where the end user is not extensively trained in computer vision.

Given a robust snake solution, the target contour can be exploited to yield more information than simply the target image centroid. Besides rapidly determining the target principal axis size and orientation,²⁵ the contour can also be used to determine corner points (see Figure 5(a)).²⁴ These snake features could be exploited in on-board filtering and estimation algorithms to determine additional relative motion information besides the typical target centroid heading information.

Finally, it should be noted that this test bed development is not tied to the color statistical pressure snake solution. As the active deformable model algorithms mature and become more efficient, other efficient segmentation routines could be used. However, the current statistical pressure snake implementation has proven to be sufficiently fast and robust to control the relative motion of unmanned vehicles.

IV. Visual Servoing Control Law

The current P3-DX UGV is capable of tracking and following a passive visual target using a statistical pressure snake. The operator first selects the target on the screen by double-clicking on it. The tracking control mode is then engaged and the vehicle will attempt to maintain a fixed heading and distance relative to the target. A terrestrial application of this setup is to have one UGV follow another UGV vehicle through visual servoing. Of interest is developing robust visual sensing solutions that can handle both lighting color and intensity variations. The results will have a direct impact on visually servoing spacecraft as well.

Given the target contour provided by the statistical pressure snake algorithm, the 0th moment (area), 1st moments (target centroid coordinates) and 2nd moments (target area inertia) are efficiently computed using Green's theorem.^{26,25} The target heading is then computed using the horizontal target centroid coordinates x_c . The target distance is computed indirectly using the major principal inertia I_1 and the target area. If the target is known to be a rectangle, then the semi-major principal axis length l (red axis in Figure 5) is determined through:²⁵

$$l = \sqrt{\frac{3I_1}{A}} \quad (7)$$

Figure 6 illustrates a simplified model of a pin-hole camera where f is the focal length and z is the depth of the target object. Using the geometric law of similar triangles, we find that

$$\frac{h}{f} = \frac{d}{z} \quad (8)$$

must be satisfied. If the snake algorithm has provided us with the semi-major principal axis image size h , then the depth of the target can be estimated using

$$z(h) = \frac{df}{h} = \frac{\xi}{h} \quad (9)$$

where $\xi = df$ is a constant for this specialized target tracking problem. If the dimension d is known in advance for the target, then ξ can be determined through a simple calibration procedure. By placing the object a known distance z_0 away from the camera, and measuring the corresponding pixel dimension h_0 , the calibration parameter ξ is found through

$$\xi = df = h_0 z_0 \quad (10)$$

directly.

This depth measurement strategy is commonly used in the robotic vision community and has well-known limitations. If the target plane is not perpendicular to the camera bore-sight axis, then the target semi-major principal axis will have a perceived foreshortening. This will make the target appear to be further away than it actually is. However, this axis foreshortening appears only in the axis modeling if second order and higher terms are included. This makes this distance measurement relatively insensitive to target plane orientation miss-alignments.²⁵

If the target size is not known, then the vehicle can be controlled to maintain the initial separation distance. This mode has many practical applications where it is not necessary to servo a specific separation distance, but rather the current separation distance is to be maintained. In this case the calibration value ξ does not have to be determined. The transient tracking performances to regulate on a stationary target will be different for false ξ values, but the UGV will still maintain the same separation distance once the target comes to rest.

To control the motion of the unmanned vehicle relative to its visual target, kinematic-based velocity steering commands are generated. For the scenario where the UGV is to track a target heading and separation distance, the two target errors can be controlled individually. Thus, to simplify the following control discussion, consider x_d to be either a desired heading or separation distance target state. Then $\delta x = x - x_d$ is the visual target tracking error. The nonlinear velocity steering command \dot{x} is

$$\dot{x} = -\gamma f(\delta x) \quad (11)$$

where $f()$ is an odd, smooth bounding function and $\gamma > 0$ is a feedback gain. Let c be a finite constant, then $f(\delta x)$

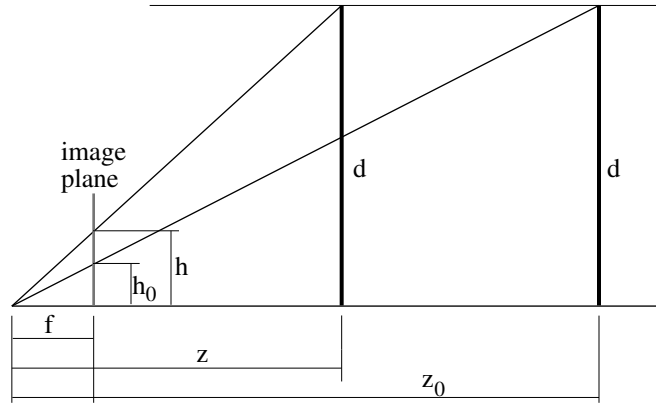


Figure 6: Illustration of Simplified Pin-Hole Camera Model Used for Depth Extraction

must satisfy

$$\lim_{\delta x \rightarrow \pm\infty} f(\delta x) = \pm c \quad (12a)$$

$$f(0) = 0 \quad (12b)$$

$$f(\delta x)\delta x > 0 \quad (12c)$$

For example, $f(x) = \arctan(x)$ could be used. The purpose of $f(\cdot)$ is to smoothly bound the UGV speed. If the target state error δx is very large, then the servo command should request a maximum response to reduce this error (saturated control). Once δx is reduced, then the control will smoothly reduce the speed command to zero. This simple visual velocity-based steering law is effective in regulating the tracking errors relative to a constant target. The current form is essentially a smoothly saturated proportional feedback control. The gain γ controls the end-game convergence, while the γc determines the maximum commanded speed.

If the target is moving, then asymptotic convergence is not expected. An advantage of this elementary visual control strategy is that it is very simple to implement. Compared to position based visual servo control methods which require extensive camera calibration before use, this velocity-based steering law only requires a very simple calibration before use in the field. Even without any calibration, this control is able to maintain an initial separation distance. Future work will investigate adding robustness enhancements to improve visual tracking for moving targets, or refining the calibration constant γ on the fly. The purpose of the current visual control strategy is to regulate tracking errors δx and drive them to zero.

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

$$V(\delta x) = \frac{\delta x^2}{2} \quad (13)$$

If the target is stationary, then $\dot{x}_d = 0$ and the Lyapunov rate

$$\dot{V} = -\gamma\delta x f(\delta x) \leq 0 \quad (14)$$

is negative definite in δx . Thus, for inertially fixed targets, the visual guidance law in Eq. (11) provides asymptotic convergence in δx .

Using the $f(\cdot)$ function in the steering law allows for vehicle speed limitations to be smoothly approached. The gain γ determines the convergence rate for small tracking errors.

However, if the target is moving, then the visual servo strategy in Eq. (11) will not yield asymptotic convergence. Let us assume that the target is moving with $\dot{x}_d = \text{constant}$. The closed-loop tracking dynamics are

$$\delta\dot{x} + \gamma f(\delta x) = -\dot{x}_d \quad (15)$$

With \dot{x}_d being constant, the \dot{V} expression is no longer negative definite and the visual servoing strategy is not Lyapunov stable. For finite \dot{x}_d values and assuming

$$\lim_{\delta x \rightarrow \infty} f(\delta x) > \dot{x}_d \quad (16)$$

\dot{V} does show that the δx tracking errors must remain finite. Because the tracking error δx is bounded, so is $V(\delta x)$. Thus, the integral of \dot{V} from $t = 0 \rightarrow \infty$ exists and is bounded.

$$\int_0^\infty \dot{V} dt = V(t_\infty) - V(t_0) = \text{finite} \quad (17)$$

Barbalat's theorem²⁷ states that if Eq. (17) exists and \dot{V} is uniformly continuous on $[0, \infty)$, then $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$. To prove uniform continuity of \dot{V} it is sufficient to show that \ddot{V} is bounded.²⁸ Taking the second derivative of \dot{V} we find

$$\ddot{V} = (\gamma f(\delta x) + \dot{x}_d)(\gamma(f(\delta x) + f'(\delta x)\delta x) + \dot{x}_d) \quad (18)$$

with $f'(\delta x) \equiv \frac{\partial f}{\partial(\delta x)}$. Thus, \ddot{V} is bounded if $f'(\delta x)$ is bounded. Using the $f()$ conditions in Eq. (12) this is satisfied and we conclude that $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$.

The condition \dot{V} is satisfied if

$$\dot{V} = 0 \quad \begin{cases} \delta x = 0 \\ \gamma f(\delta x) + \dot{x}_d = 0 \end{cases} \quad (19)$$

Looking at the closed loop tracking dynamics in Eq. (15), it is evident that $\delta x = 0$ cannot remain true for a finite amount of time with a constant target rate \dot{x}_d . Instead, the steady-state tracking error is determined through

$$\gamma f(\delta x_{ss}) + \dot{x}_d = 0 \quad (20)$$

Solving for the steady-state tracking error due to a constant target motion \dot{x}_d , we find

$$\delta x_{ss} = -f^{-1}\left(\frac{\dot{x}_d}{\gamma}\right) \quad (21)$$

Thus, for constantly moving targets the tracking errors will have a finite value.

V. Hardware Visual Servoing Results for Free Inertial Motion Simulation

This section presents hardware visual tracking results which use the visual servo control strategy outlined in section IV. The UGV simulates the free inertial motion of a craft. For example, this test could simulate a floating robotic assistant visually servoing on a mother spacecraft in interplanetary space. Orbital dynamics or flight dynamics behavioral modules have not been created yet. Given the visual tracking errors, the servo commands will drive the vehicle. If this were a spacecraft, then it is assumed that an IMU is onboard to measure the actual inertial response and a velocity sub-servo system would be present to implement the commanded velocity commands. In an expanded relative motion test bed IMUs could be used to measure the vehicle motion which would then servo the velocity command. This would simulate all required spacecraft motion sensing.

The visual target is a green rectangle surrounded by a red border as is seen in Figure 7. The camera/target calibration parameter ξ was determined at the desired 1 meter target separation distance. Thus, the pin-hole camera-based distance measurement will be the most accurate around the desired 1 meter separation distance.

The visual snake automatic gain selection is set to track hue colors only here. This provides the largest robustness to color intensity variations. However, this also makes the visual snakes more susceptible to spilling over onto other green areas in the image background. This is why the red border is provided around the green target to help contain the snake. As is seen in the hardware results below, this strategy does provide effective visual tracking of the target when the human person guiding the target can move it smoothly at will. This scenario is referred to as a collaborative, passive visual target. No active communication is occurring between the visual target and the sensor (camera) as with optical beacons. The visual target is collaborative because it has a known shape and size. If arbitrary visual targets are to be tracked, then the hue-only visual snake mode would not be used. Instead, a blend of gains across the HSV space will be used to track the target. This mode is currently less robust to lighting intensity variations, but is less easily fooled by other image components with a similar hue. Such a target is referred to as a non-collaborative, passive visual target.

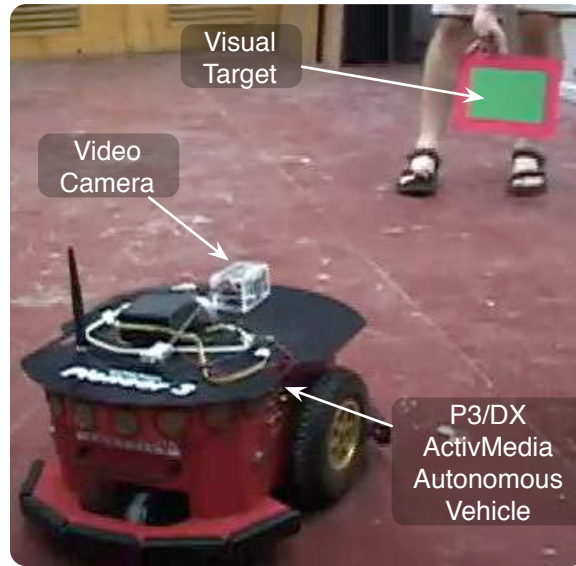
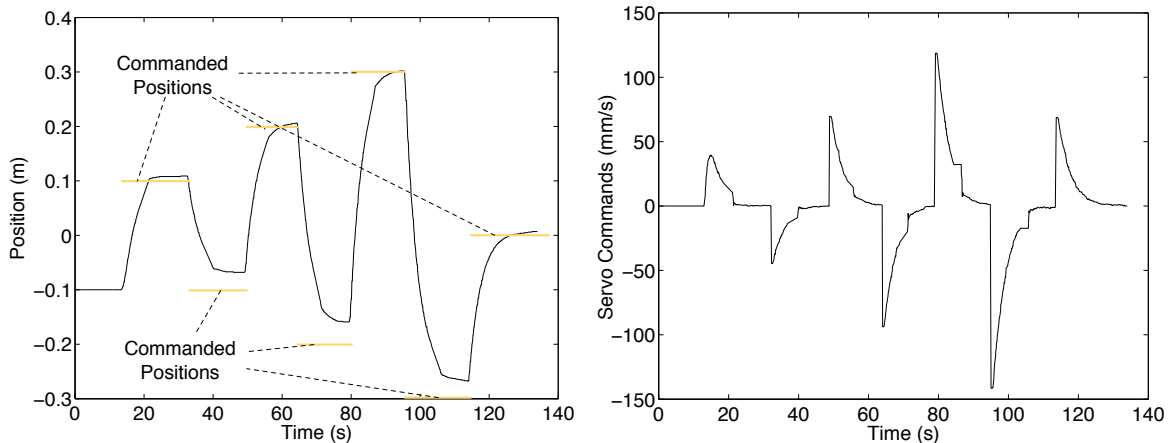


Figure 7: Illustration of P3-DX following a rectangular target which is manually moved through a laboratory.



a) Change in Actual Vehicle Position Relative to Calibration Position (positive is closer to target, negative is further away)

b) P3-DX Forward/Backward Servo Commands

Figure 8: Visual distance servoing simulation results where the commanded separation distance is changed in discrete steps.

A. Distance Servoing Experiment

To test the distance servoing capability of the visual servo strategy, the green target is placed vertically on the wall and the target/camera parameter ξ is calibrated at a 1 meter distance. After engaging the visual servoing, the commanded separation distance are changed at discrete intervals. The vehicle will then move forward, or backward, to achieve the desired separation distance to the visual target.

Figure 8(a) shows the change in vehicle position relative to the initial 1 meter separation distance. Positive changes indicate the vehicle moving forward. The vehicle motion is determined here through the wheel encoders. Because the vehicle is only performing simple forward and backward motions on a non-slippery surface, these encoder derived position measurements are expected to be sufficiently accurate. The test bed currently does not have method to externally track the inertial motion.

The initial commanded separation distance is 1.1 meters, or -0.1 meters from the calibration distance. Then separation distances of 0.9, 1.1, 0.8, 1.2, 0.7, 1.3 and 1.0 meters are commanded. Each time the vehicle responds quickly and approaches the desired target separation distance. However, note that as the separation distances differ increasingly from the 1 meter calibration distance, the measured target separation distance begins to be less accurate. In particular,

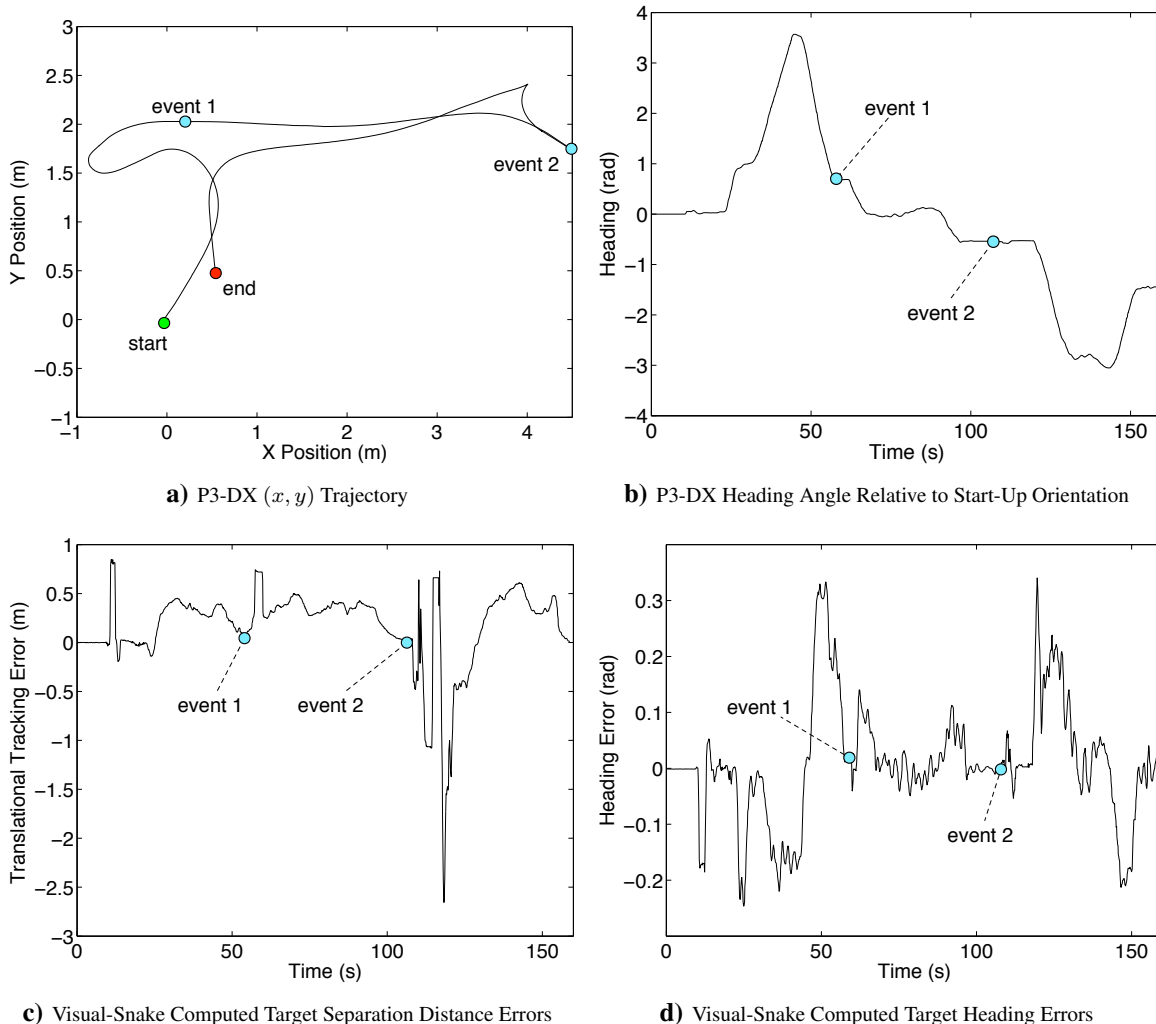


Figure 9: Experimental Results of the 2D Tracking Test of a Randomly Moving Visual Target in a Visually Clutter Laboratory Environment.

as the vehicle moves further away from the target, the pin-hole camera model is becoming less valid. Computing the distance using the visual snake semi-major principal axis provides increasingly less accurate separation distance calculations as the separation distance differs more from the calibration distance. At the end of the experiment the original calibration separation distance is commanded. Here the vehicle converges very well to this desired separation distance. The commanded vehicle servo speeds are shown in Figure 8(b).

This hardware test demonstrates that the visual servoing strategy can successfully track a calibrated separation distance. Other separation distances can also be commanded, but with more coarse final positioning. Consider the scenario where one vehicle is to track a lead craft at a given separation distance, for example, a convoy of UGVs. This separation distance tracking method is a promising solution to achieve such visual servoing.

B. 2D Visual Tracking Experiment

In the second hardware visual servoing experiment, the P3-DX vehicle tracks the same green visual target. However, in this scenario the target is smoothly moved by a human as shown in Figure 7. The identical visual snake settings (hue-only tracking etc.) and the visual servo control gains are used. The human is essentially mimicking a lead vehicle moving through the laboratory, and the P3-DX must maintain a 1 meter separation distance while directly pointing at the target. The visual servo control supplies both heading and vehicle speed commands to achieve this.

If the target is moving at a constant rate, the current visual control strategy is shown to have a steady-state separation distance bias in Eq. (21). Thus, while the target is moving, the separation distance errors are not expected to

decay to zero.

The experimental results are illustrated in Figure 9. The vehicle motion within the laboratory is computed using the wheel encoder information. Figure 9(a) shows the resulting trajectory. After starting up, the P3-DX performed a slow left, then a sharp right turn, to follow the visual target. The vehicle heading angle relative to the startup orientation is shown in Figure 9(b). The sensed (computed through the visual snake information) separation and heading tracking errors are illustrated in Figures 9(c) and 9(d).

The start and stop points are marked on the trajectory plot in Figure 9(a), as well as two special events that occurred. At event 1 the green target was rapidly removed from the P3-DX field of view. This causes the visual snake to lose target lock. The control software recognized this event and automatically put the visual servo routine into stand-by mode. After the target was made visible again to the P3-DX camera, the visual snake routine was able to reacquire a target lock and the control resumed. As the snake acquires the target, the relatively slow onboard computer of the P3-DX will take several frames to fully track the green square. This causes erroneous separation distance transients which are visible in Figure 9(c).

The second event occurred around 110 seconds into the simulation where the target was held still to let the vehicle converge to the desired separation distance and heading. After this, the target was removed again to temporarily disengage the visual servo control. When presenting the target again, it was waved and rotated in front of the camera for several seconds to let the visual snake reacquire target lock. During this period visual snake convergence transients occurred which resulted in the short period of erratic separation error measurements. After the target is held steady, the smooth visual control continued. These two events help illustrate the robustness of the current visual servo implementation.

VI. Conclusion

A hardware simulation test bed designed to test the performance of visual sensing and control methods in relative navigation problems has been developed. This test bed is intended to eventually be used to simulate unmanned ground, air, and space vehicle relative navigation problems. The UGV provides the sensor or test hardware with physical motion which is equivalent to that of the aerospace vehicle being tested. By using mobile robotic ground vehicles, very large ranges of motion and extremely varied lighting conditions can be simulated. The implemented visual tracking algorithm employing color statistical pressure snakes is discussed. A simple but effective and easy to tune visual servo control is presented which is shown through hardware experiments to function well with the visual sensing strategy. Experimental results are discussed where a craft is kinematically commanded to point at a visual target, and maintain a 1 meter separation distance, even while the visual target is moving in an unstructured laboratory environment.

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

- ¹Junkins, 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.
- ²Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic System*, Chapman & Hall/CRC, Boca Raton, FL, 2004.
- ³Schwartz, J. L., Peck, M. A., and Hall, C. D., "Historical Review of Air-Bearing Spacecraft Simulators," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 26, No. 4, 2003, pp. 513-522.
- ⁴Schubert, H. and How, J., "Space Construction: An Experimental Testbed to Develop Enabling Technologies," *Proceedings of the Conference on Telem manipulator and Telepresence Technologies IV*, IEEE, Piscataway, NJ, 1997, pp. 179-188.
- ⁵Matunaga, S., Yoshihara, K., Takahashi, T., Tsurumi, S., and Ui, K., "Ground Experiment System for Dual-Manipulation-Based Capture of Damaged Satellites," *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE, Piscataway, NJ, 2000, pp. 1847-1852.
- ⁶Miller, D., Saenz-Otero, A., and et. al., J. W., "SPHERES: A Testbed for Long Duration Satellite Formation Flying in Micro-Gravity Conditions," *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, Univelt, San Diego, CA, 2000, pp. 167-179.
- ⁷Creamer, G., Pipitone, F., Gilbreath, C., Bird, D., and Hollander, S., "NRL Technologies for Autonomous Inter-Space Rendezvous and Proximity Operations," *Advances in the Astronautical Sciences, The John L. Junkins Astrodynamics Symposium*, Vol. 115, American Astronautical Society, 2003, pp. 233-250.
- ⁸Roe, F. D., Mitchel, D. W., Linner, B. M., and Kelly, D. L., "Simulation Techniques for Avionics Systems - An Introduction to a World Class Facility," *Proceedings of the Flight Simulation Technologies Conference*, AIAA, Reston, VA, 1996, pp. 535-543.
- ⁹"Analysis of Robotics Technology - Umbra," 2005, <http://www.sandia.gov/isrc/umbra.html>.
- ¹⁰Atkins, E. M., Lennon, J. A., and Peasco, R. S., "Vision-based Following for Cooperative Astronaut-Robot Operations," *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2002.

- ¹¹Lennon, J. and Atkins, E., "Color-based Vision Tracking for an Astronaut EVA Assist Vehicle," *Proceedings of the SAE International Conference on Environmental Systems (ICES)*, Orlando, FL, July 2001.
- ¹²Das, A. K., Fierro, R., Kumar, R. V., Southall, B., Spletzer, J. R., and Taylor, C. J., "Real-Time Vision-Based Control of a Nonholonomic Mobile Robot," *IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 1714–1719.
- ¹³Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. J., "A Vision-Based Formation Control Framework," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 813–825.
- ¹⁴"StarVision Technologies Inc." 2005, http://vesuvius.jsc.nasa.gov/er_er/html/sprint/.
- ¹⁵Schaub, 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.
- ¹⁶Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: active contour models," *International Journal of Computer Vision*, Vol. 1, No. 4, 1987, pp. 321–331.
- ¹⁷Perrin, 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.
- ¹⁸Perrin, 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.
- ¹⁹Malladi, 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.
- ²⁰Ivins, 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.
- ²¹Smith, 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.
- ²²Schaub, H., "Statistical Pressure Snakes based on Color Images," Technical Report SAND2004-1867, Sandia National Laboratories, Albuquerque, NM, 2004.
- ²³Schermann, E. and Ebersole, T., "An Implementation of Color Statistical Pressure Snakes using HSV Color Space and Redefined Internal Energy Terms," CS 766 Project.
- ²⁴Schaub, 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.
- ²⁵Schaub, H., "Extracting Primary Features of a Statistical Pressure Snake," Technical Report SAND2004-1869, Sandia National Laboratories, Albuquerque, NM, 2004.
- ²⁶Steger, C., "On the Calculation of Moments of Polygons," Technical Report FGBV-96-04, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München, Aug. 1996.
- ²⁷Khalil, H. K., *Nonlinear Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 3rd ed., 2002.
- ²⁸Slotine, J. E. and Li, W., *Applied Nonlinear Control*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991.