JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 3, December 2006

A Modal Operations Paradigm for Robust Vision-based Astronaut Following

Timothy Wasserman*
University of Maryland, College Park, MD 20742, USA
Jamie A. Lennon†
Naval Research Laboratory, Washington, DC 20735, USA
and
Ella M. Atkins‡
University of Maryland, College Park, MD 20742, USA

Scenarios for manned missions to the Moon or Mars call for astronaut extravehicular teams to be accompanied by autonomous or semi-autonomous rovers. These rovers must be able to safely follow the astronauts over a variety of terrain with minimal assistance. We use a commercial color-based vision system as a high-speed sensory approach for tracking the astronaut and present a quantitative analysis of its accuracy based on a spectrum of known motion profiles. Visual tracking data is transformed into a sequence of waypoints that allow a rover to accurately follow the safe path the astronaut has previously traversed. To promote robust operation during exceptional situations, a modal operations paradigm is presented that enables the rover to recover autonomously when possible and with minimal supervision when required. We have outfitted a mobile robot with a camera connected to our color tracking system and characterized its ability to follow the astronaut target over a variety of motion profiles under nominal and exceptional circumstances. Results show the rover is able to autonomously follow and re-acquire the astronaut in most cases and is able to safely stop and request assistance otherwise.

I. Introduction

WHEN humans go to the Moon and Mars, they will undoubtedly take robots with them. Robotic assistants can greatly enhance crew productivity, either by autonomously completing tasks on their own or assisting the astronauts.[1] Astronauts on surface extravehicular activities (SEVAs) will certainly find rover assistants to be useful. A rover can carry cumbersome tools and heavy samples. It can also provide an additional measure of safety if it carries extra life support and communications equipment.

Such a rover will ideally have a multitude of operational modes. The astronauts may wish to ride on it and drive to a remote site. They may desire to direct the rover to autonomously approach a marker or beacon and perform some arduous task such as core sample extraction. The rover could be teleoperated by a crewmember at the base camp. At least some of the time, we expect that the SEVA crew will be walking over the Martian surface and will want the rover to follow them. For instance, if astronauts ride the rover to a remote site to take samples, they may wish to

Received 22 September 2005; revision received 1 May 2006; accepted for publication 22 June 2006.

*Graduate Research Assistant, Aerospace Engineering Dept./Space Systems Laboratory, University of Maryland, College Park, MD 20742, Student Member.

†Graduate Fellow, Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC 20735, Code 5515, Student Member.

‡Assistant Professor, Aerospace Engineering Dept./Space Systems Laboratory, University of Maryland, College Park, MD 20742, Associate Fellow.

disembark to look for interesting rocks. It will be very useful if the rover can follow them about, keeping a variety of tools handy, taking detailed *in situ* data on the samples before their removal, and then accepting the samples from the astronaut.

An astronaut in SEVA is already performing navigation and some obstacle avoidance as he or she traverses the Martian terrain. We wish to take advantage of advanced human perception capabilities to augment rover navigation. When "following" an astronaut, the rover does not need to perform extensive analyses of the surrounding terrain if it can rely on its human partner to pick a safe path. It needs only to record that path and then follow it while maintaining a safe distance between itself and the astronaut. How will it observe and record that path?

In this paper, we describe a modal operations model to enable a rover to robustly follow an astronaut. While the astronaut is successfully tracked, the rover's task is to follow the astronaut at a safe distance. This goal is accomplished by the progressive definition of waypoints along the astronaut's path that enable the rover to traverse the same terrain as the astronaut. We use a commercial color-based vision system (Cognachrome™) to track the astronaut. To calibrate the system and characterize tracker performance, we first deployed the vision system on a hardware-based simulator that enables known changes in relative position and velocity between the simulated astronaut and rover. A computer-controlled, rail-mounted cart with a 5 m range of translational motion provides changes in the distance between the two. A pan-tilt unit (PTU) generates perceived changes in relative heading. The "astronaut" marker was placed on the cart, while the "rover camera" was mounted on the PTU. We show that the vision system we are using accurately determines astronaut position under nominal conditions. However, given the possibility of highly dynamic lighting conditions, obstacles to occlude the astronaut target, and other exceptional circumstances, the color tracking system alone is insufficient to guarantee robust operation.

To handle such exceptions, we present a modal operations model that allows the rover to assess its circumstances once data is lost, and proceed to the last known waypoint, after which either the astronaut target is reacquired or else the rover asks for assistance. This contingency paradigm has been implemented and evaluated on a four-wheel rover with camera mounted to an onboard PTU. Results are presented for nominal tracking at 1-2 m following range and for a variety of exceptional situations, illustrating the rover's ability to follow and re-acquire its astronaut target as well as identify the need for help when required.

II. Background

Moon and Mars exploration is receiving much attention, and numerous activities are underway to define the missions and the technologies required to support these missions. Not too long ago, a Mars Reference Mission was defined, calling for at least three types of planetary exploration rovers.[2] The first class is for use in the near vicinity of the base camp and may be nothing more than wagons or carts. The second class is the unpressurized rover similar to the Apollo lunar rover that would be used for a six to eight hour sortie away from base. The third class is the mobile pressurized habitat, allowing crew to undertake ten-day missions far from the base. While this third sort is to have robotic arms for sample collection, it also has an airlock, so that astronauts may disembark. It may be desirable for both rovers of the second and third class to follow a mobile astronaut over the planetary surface. It is *not* desirable to occupy a SEVA astronaut or an astronaut at base with any more rover operations than necessary.[3] We would like for the rover to be able to follow the astronaut on its own.

NASA experiments using the Marsokhod rover have previously tested the "follow the astronaut" paradigm.[4] These tests show that the rover is useful only if it can keep up with the astronaut. Otherwise, the astronaut must spend a great deal of precious SEVA time waiting for the rover to catch up. This means the rover needs to travel at speeds of about 1.4 m/s. (This is an average walking speed for an unencumbered human on Earth, and it is unlikely that a suited astronaut will exceed it.) Earth-based vehicles heavily laden with sensors and computers have demonstrated relatively high autonomous traversal speeds (e.g., successful DARPA Grand Challenge entrants[5]). However, planetary surface robotic systems cannot yet achieve the same competence due to size/power, sensing, and computational resource limitations. Planetary surface rover research test beds typically navigate and avoid obstacles using stereo vision. Although navigation and human/astronaut interaction capabilities continue to improve,[6] typical traversal speeds have been reported of around 0.5 m/s,[7,8] too slow for following an astronaut at a "natural" gait. Even with capable sensors, speed is also limited by required path planning computations, which in complex terrain may still require slowing or stopping the rover while the environment and next goal waypoint are processed.

Efficient resource utilization—including computing resources—is one of the essential components of a successful autonomous system. Color blob tracking is a simple and fast vision process. While it is sensitive to certain lighting conditions, it can be surprisingly robust for a range of light intensities. For these very reasons, color vision processing was successfully used to track a hyper-redundant arm and estimate its pose in research done at Rice University.[9] It has also been used for the fast-acting agents that participate in robotic soccer competitions.[10] The Cognachrome Vision System™ was designed as a high-speed hardware-based solution to robust color tracking. It provides useful information about a given object, such as its centroid and the area of the image on the screen. In our simulator, we can relate the centroid to heading and the area to range. The Cognachrome™ System is able to process up to 60 video frames per second and minimizes processor overhead on the navigation computer. It has been proven to be fast, accurate, and robust in several robotics competitions.[11] It also does not require the astronaut to carry any emitters or equipment other than a small colored target.

Below, we characterize our modal operations paradigm to enable a rover to robustly follow an astronaut, appropriately reacting in cases where the astronaut cannot be "seen" by the vision system. Next, we characterize the performance of our vision system and describe a series of experiments with a rover and astronaut which demonstrate a suite of nominal and exceptional situations.

III. Modal Operations for Exception Handling

Although not perfect, color blob tracking is sufficiently accurate to enable a rover to follow its astronaut companion with very low processing overhead, as will be shown below. But there are many situations where the rover could find lighting conditions impossible to handle or simply lose sight of the astronaut. How shall it proceed in these all too-frequent cases?

A BIOLOGICAL PARADIGM - A possible answer lies in examining another multiagent expeditionary force—the family on vacation. Generally, a few members (the adults) fulfill the role of the astronaut, selecting areas of interest and leading the rest of the family to them. The children tag along behind, through complicated fields of mobile obstacles that bear a strong resemblance to their tracked targets. To assist their children in tracking them, many parents provide extra visual clues—a balloon of a certain color high overhead, or a distinctive hat worn on a tall adult. But as we all know, even these augmented color blob tracking methods can fail, and the child may find himself separated from the adult.

What the child does in this situation depends on how he has lost track of the adult and where he is at the moment. If the child cannot see the adult because the adult just walked behind the corner of a building, the child is sure that if he also continues around the corner of the building, he will see the adult again. This loss of signal is far less troubling than when the child, preoccupied by something else, looks around to discover that he has lost sight of the tracked object. He does not know where the adult has gotten off to, in what direction, or how far away he might be. If the child feels truly lost, he will engage in a behavior to aid in becoming found again. The behavior selected may depend on the environment. A child in an unknown, potentially hostile environment has probably been instructed to do one of two things. First, he can seek out a known safe spot or landmark—a police officer or a meeting place established in case of separation. Second, he can stay where he is so his parents can backtrack and find him. Broadcasting a distress signal is an option in both cases.

A child in a known or structured environment might feel comfortable searching for his parents. If separated in, for instance, a grocery store, a simple pattern search checking each of the aisles is likely to reveal the location of the parents. The child can use his knowledge of his parents' shopping patterns (starting at one end of the store and going through the aisles, moving slowly in the aisles as they shop) and his other beliefs (that they could also be in line at the checkout, that they won't leave without him) to aid in the search. A simple visual search can again be augmented by other means, particularly widely broadcasting a signal of some kind. The child might call out for his parents as he looks down the aisles, or might even ask the store owner to use the public address system to call them.

A rover can be programmed with such behaviors. Figures 1–2 show the logic we implemented to switch between "Acquire Target", "Follow", and "Stop" operational modes. The "Acquire Target" steps (Fig. 1) are autonomously executed both for the case where the rover loses its target while tracking and for the case when it disengages from another task and cannot immediately find its astronaut target. The goals of this procedure are to reacquire the target when possible or else characterize the circumstances by which the target was lost. Diagnostic information is collected (italicized in Fig. 1) and subsequently used in cases where the target is not re-acquired. As shown in the figure, the first

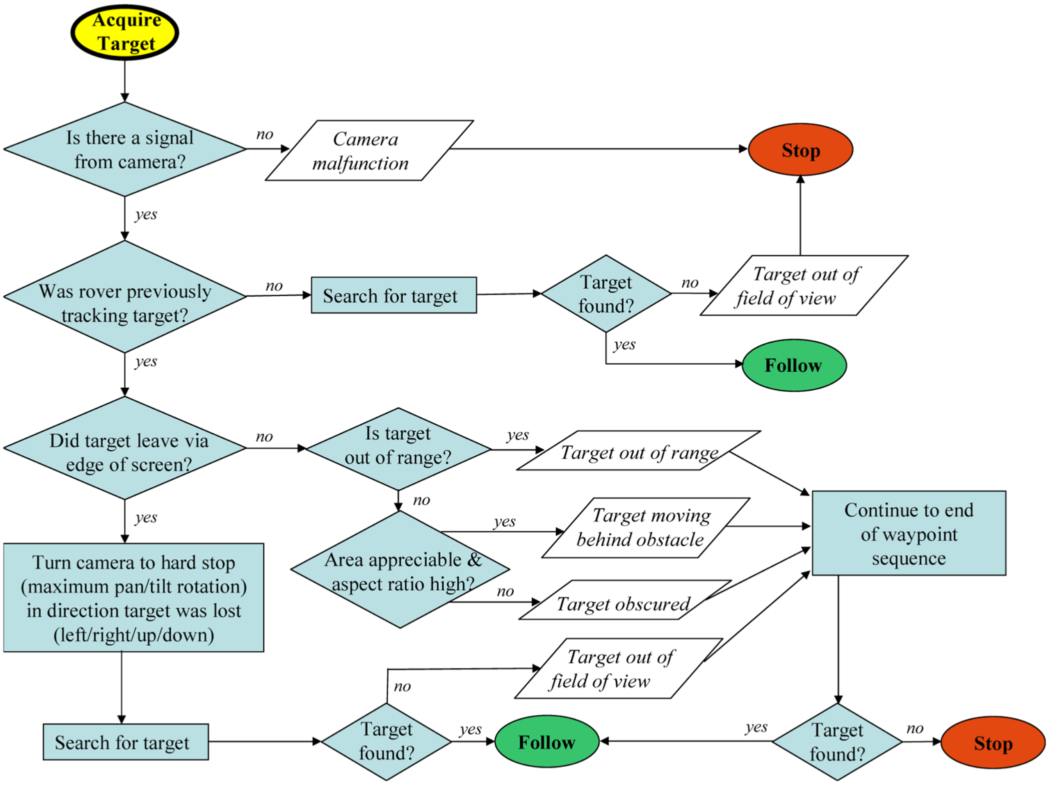

Acquire Target
Is there a signal from camera?
no
Camera malfunction
Stop
yes
Was rover previously tracking target?
no
Search for target
Target found?
no
Target out of field of view
yes
Follow
yes
Did target leave via edge of screen?
no
Is target out of range?
yes
Target out of range
no
Area appreciable & aspect ratio high?
yes
Target moving behind obstacle
no
Target obscured
Continue to end of waypoint sequence
yes
Turn camera to hard stop (maximum pan/tilt rotation) in direction target was lost (left/right/up/down)
Target out of field of view
no
Search for target
Target found?
yes
Follow
yes
Target found?
no
Stop

Fig. 1 Autonomous target acquisition sequence.

step is to check the camera hardware.[§] If either the camera or Cognachrome™ vision system is not working, tracking cannot occur and the system transitions to "Stop" (Fig. 2). Otherwise, if the target was not previously being tracked, an in-place search for the target is conducted with the onboard pan-tilt unit, after which either the rover begins to "Follow" the acquired astronaut or else transitions to semi-autonomous "Stop" mode. If the rover was tracking when its target was lost, it can utilize acquired data to help diagnose and recover from the loss-of-target situation. A target might, for example, move off the edge of the image plane. In this case, the rover can presume target loss was caused by fast relative motion or positioning in a "blind spot" (beyond pan-tilt unit joint limits). In such a case, the rover searches for the target until it reaches the end of its known traversal path (i.e., the target's recorded path). If the target did not move off the edge of the image plane, the history of its observed aspect ratio and area are used to classify the situation. The target is deemed out of range if the rover, traveling at its maximum speed, is unable to keep up with the target. Such a situation is flagged when target area progressively decreases to an indistinguishable value. Given a circular target, a high aspect ratio and appreciable area indicate the target is still partially visible but moving behind an obstacle, a transition we would expect to observe given our high data acquisition rate. These situations are noted, and the rover travels to the last observed target location, after which it either "Stops" or resumes "Following" the re-acquired target.

In cases where the rover cannot autonomously re-acquire its target (Fig. 1), it "Stops" and enters a semi-autonomous mode (Fig. 9) in which astronaut assistance is likely required. As shown in Fig. 9, diagnostic information collected during the target acquisition process (Fig. 1) is used to direct the rover's continued activities. The goal of this semi-autonomous operational mode is to minimize overhead for the rover's operator and its astronaut companion, but to

[§] A full fault detection sequence would also check the rover's servos and [redundant] sensors for proper operation. For this work, we focus on the vision system thus limit exception handling to vision system hardware failures. This simplification is also driven by practical concerns since our rover's sole source of feedback is its vision system.

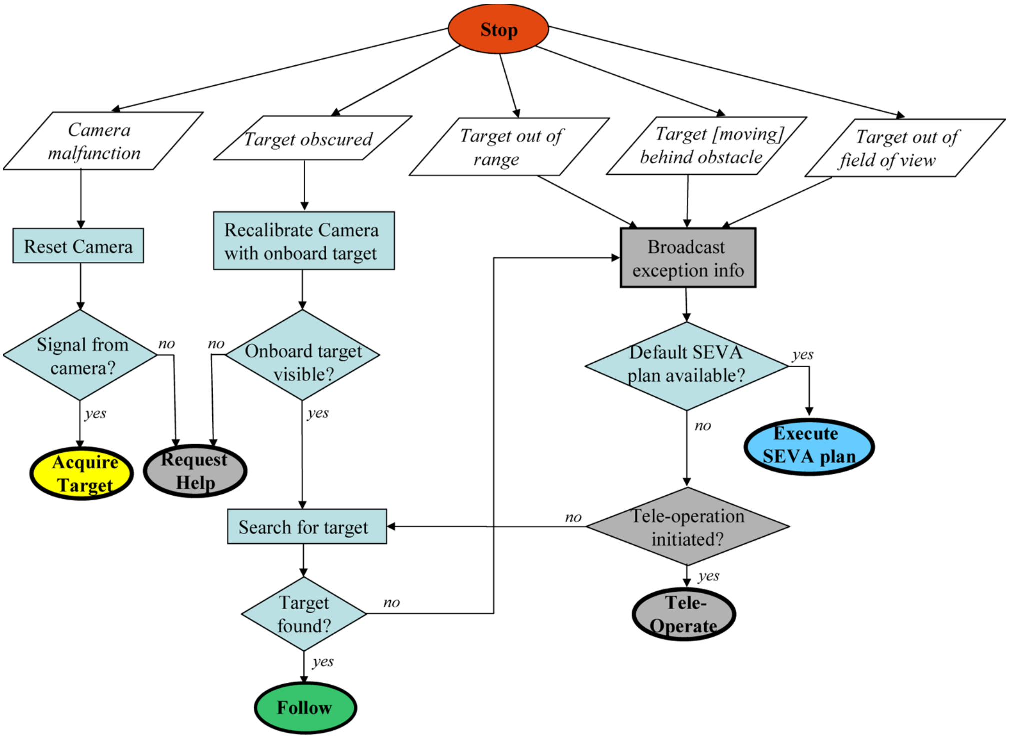


Fig. 2 Semi-autonomous exception handling (user interaction in grey boxes).

enable continued operation rather than mission failure due to a lost target. The most severe exception is a camera malfunction. If a reset doesn't remedy the situation, help is requested. An obscured target might result from changes in lighting or a Martian dust storm. An onboard calibration target at a known location could allow the rover to approximately recalibrate to the new conditions, after which it either continues to search for the target or asks for help when recalibration fails.

In cases where the target is out of range, out of the field of view, or behind an obstacle, a message is broadcast to the astronaut and operator indicating the exception situation and the possible need for assistance. Such a message can cue the astronaut to return to a location where the rover can again have a clear view, or it can indicate to a remote supervisor the need to intervene by providing a new SEVA plan (given slow-speed autonomous navigation capability) or direct teleoperated control commands. Figure 2 indicates this sequence. If an SEVA traversal plan is available, that plan is initiated (e.g., travel to a specific waypoint: a "safe meeting place" or the "customer service" desk), or if the operator has requested direct control, the rover switches to teleoperation mode. With no input, the rover loops between the PTU-based target search and exception re-broadcast tasks until either the astronaut returns to where the rover can again follow or the operator issues commands. Since we presume the astronaut has been alerted to the rover's loss of target situation, we also presume the astronaut will take care to "reappear" at a location the rover can reach via a direct route. Without this presumption, the rover would need to ask the operator to validate its path rather than just re-initiating the "Follow" operational mode.

IV. Simulator-based Vision System Performance Characterization

To initially calibrate and characterize nominal color tracking accuracy, we constructed a hardware-based simulator that could independently measure the parameters provided by our color tracking system. The simulator (Fig. 3) consisted of a camera mounted on a computer-controlled pan-tilt unit (PTU) and a computer-controlled cart mounted on a rail. The camera on the PTU simulated a camera on the rover. The colored object to be tracked (a green ball, in this case) was placed on the cart to simulate a target on the astronaut. We assumed that the astronaut would either wear some sort of appropriate tracking target on his spacesuit, or that the spacesuit's rounded helmet can

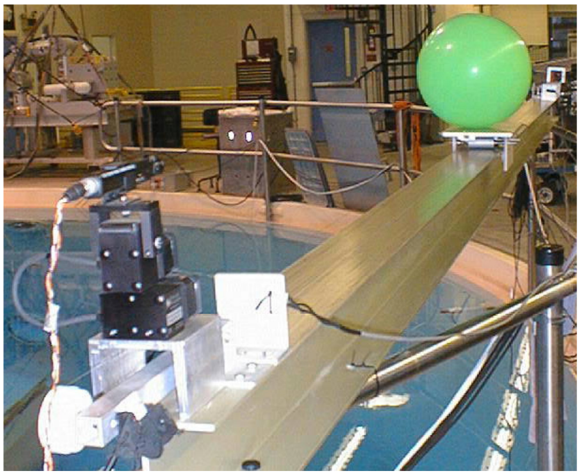


Fig. 3 Rail cart + PTU motion simulator.

be separately tracked. Figure 4 shows the schematic representation of the simulator used to calculate the Table 1 Denavit-Hartenburg (DH) parameters[12] that together describe the coordinate transformation ${}^0_4T$ between astronaut and camera coordinates. ${}^0_4T$ and its inverse specify the expected perceived location of the tracked object as a function of θ_2, θ_3, and δ_1. The "actual" coordinates fed back from the PTU/rail cart system are compared to measurements from the vision system to calibrate and validate the vision system. The cart was moved radially to model changes in range r between the rover and astronaut. We can assume that the range r from the camera to the target is approximately equal to the horizontal distance δ_1 between them via the small angle assumption. The difference in height between elevations a and $c + d$ shown in Figure 4 is a mere 2.4 cm, while the minimum horizontal distance δ_1 considered in our experiments was 1.0 m.

The velocity of the cart simulates the relative velocity between the two. If we wanted to show the astronaut and the rover traveling together at constant velocity, the cart should be remain at a constant range with zero velocity. If the astronaut were to slow down or the rover to speed up relative to the previous constant velocity, the cart would move towards the camera, simulating the rover closing in on the astronaut. The reverse case—the astronaut walking more quickly or the rover slowing down—would be indicated by moving the cart away from the camera at a velocity

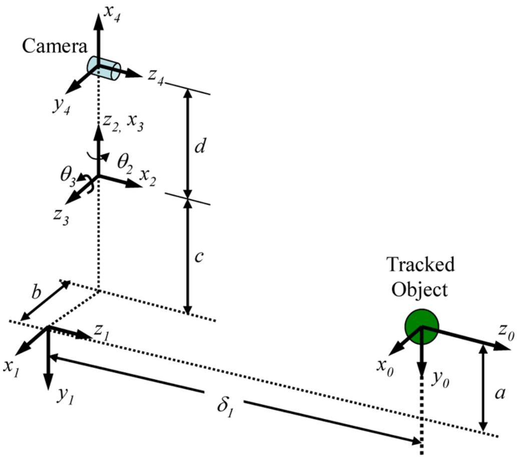


Fig. 4 Simulator link coordinate frames.

Table 1 Rail cart simulator DH parameters.

Frame	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	a	δ_1	0
2	90°	b	c	$90° + \theta_2$
3	90°	0	0	$90° + \theta_3$
4	90°	d	0	0

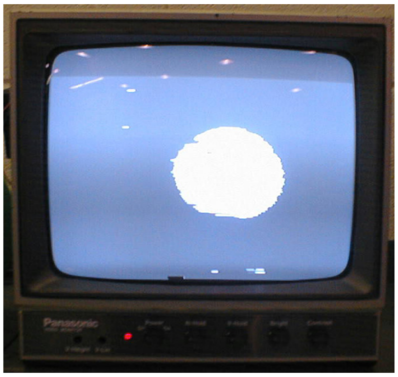


Fig. 5 Video after Cognachrome™ processing.

equal to the difference in the rates of the astronaut and rover. A motor mounted at one end of the rail controlled the cart's motion. The transmission device was a combination of chained gears and belt-driven rollers.

The PTU panned to simulate changes in heading θ_2, moving in the opposite direction of the desired astronaut movement. For example, if the astronaut was to move +30°, the PTU panned −30°. From the camera's point of view, it seemed as if the tracked object had moved itself +30°, which is the intended result. The PTU was also used to tilt, indicating a change in astronaut elevation. While a rover cannot, of course, change its elevation independent of terrain, we assumed that an astronaut cannot, either. If the astronaut's elevation is changing, then the astronaut is going up or down a hill. The rover will follow the astronaut up or down the hill shortly. But keeping the astronaut on-camera for range and heading estimation is still desirable. Monitoring elevation enables camera pointing, allowing the rover to follow the astronaut with its "eyes" until its body catches up. The COTS PTU responses were highly repeatable with no observable overshoot.

The Cognachrome Vision System™ implemented the color blob tracking used to determine astronaut range and heading. It filtered out background noise and identified only the tracked object. Figure 4 illustrated the visually-cluttered area around the simulator which includes reflective surfaces and multiple light sources. Figure 5 shows the camera view after processing by Cognachrome™. Only the tracked object—the ball—remains. We chose a spherical object so that its aspect ratio would remain 1.0 as long as the object remained entirely on the screen, enabling a change in aspect ratio to signal an exception condition. Experience has shown the color tracking works best with highly saturated ("neon") colors.

A. Vision System Error Analysis

To characterize the static error and noise in the vision system, Cognachrome™ data for the area of the tracked object and its centroid were recorded for known ranges. Initial tests were conducted with a 3.5 mm camera lens and a 6" diameter ball. We wanted results that could be generalized to any camera lens and any size target object, so we computed the percent of the screen that the tracked object's image occupied. The tracked object occupied between 60% and 0.03% of the screen in this test series. For a non-dimensional percentile measure of error, we computed:

$$e = \frac{\sigma \times 100\%}{\bar{x}} \tag{1}$$

where σ was the standard deviation of the reading and $\bar{x}$ was the average reading.

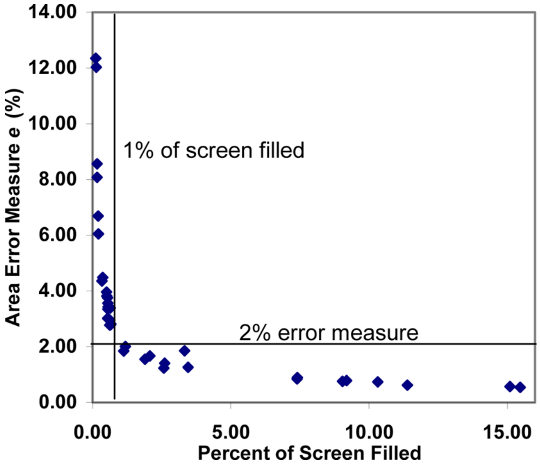


Fig. 6 Percent of screen filled vs. error measure (*e*) for area.

When the target object occupied at least 1% of the screen, this error in the area data was kept below 2% (Fig. 6). After this point, error in the area data spiked quickly, rising up to 12%. In general, centroid data was much more consistent than area data (Fig. 7), as expected since area is more susceptible to noise. Centroid error remained very low—less than 0.5%—even when the tracked object occupied 0.25% of the screen. When the tracked object became extremely small (10 pixels or less) on the screen, it approximated the size of noise picked up by the system, and sometimes the centroid of the noise was reported instead. These results are encouraging. When the astronaut is far away from the rover (but still close enough that his target object occupies more than 0.25% of the screen), the rover can still accurately determine in which direction to move to catch up. As the rover gets closer to the astronaut, it is better able to determine their separation, and it is at these closer ranges that better accuracy is needed for safety.

When the tracked object fills the screen, we are no longer able to determine if it moves closer. To maximize the range at which our area data is accurate, we would like this "filled screen" condition to occur when the object is as close to the camera as we would ever want it to get while astronaut following is enabled. (There are obviously cases where the astronaut might want to approach or even board a stopped rover; we are not considering those. We assumed the astronaut can turn off the following behavior at will). We chose 1.0 m as a "danger range". If the rover detects

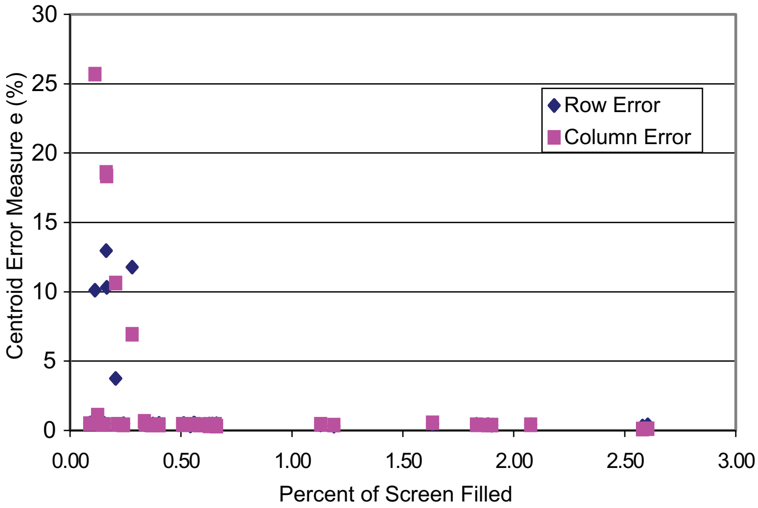


Fig. 7 Percent of screen filled vs. centroid row and column error measure *e*.

that the astronaut is within 1.0 m, it will stop immediately. By choosing an appropriate lens and tracked object, we were able to frame the object image so that it just touches the edges of the frame when the object was 1.0 m from the camera and centered in the field of view.

The inverse of ${}^{0}_{4}T$ allowed us to independently measure the position of the ball relative to the camera for any commanded values of θ_2, θ_3, and δ_1. We desired to compare these expected values to actual experimental data. This first required converting between Cognachrome's™ measurements, which were in pixels, and distance. Cognachrome's™ pixelation and the size of the digital CCD chip provided this conversion. Next, we related CCD displacements to actual distances via the simple pinhole camera equation:

$$\frac{d_{ball}}{r} = \frac{d_{image}}{f} \tag{2}$$

where d_{ball} is the diameter of the target ball, r is the range from the camera to the center of the ball, d_{image} is the diameter of the target image, and f is the camera focal length. From this, we found the camera's field of view in both pitch and yaw. We also were able to determine the range for which the top and bottom of the target should just fill the screen vertically, the "danger range" described above. The theoretical value was 0.95 m, very close to our 1.0 m estimate. Range is inversely proportional to the inverse of the "area measure" provided by Cognachrome™, which is the square root of the number of target object pixels (the area).

Next, we took experimental data to adjust and verify our approximate theoretical calibrations. We took data for a series of known ranges and headings and generated equations correlating Cognachrome™ data for area and centroid with range and heading. The coefficient found for the range-area measure relation was within 10% of the theoretical value. Pan and tilt calibrations were more involved. First, camera offset had to be considered. We took the row or column number given by the Cognachrome™ for zero degrees and put it into the theoretical expression, giving a correction factor. The correction factor was used to adjust the intercept of the theoretical equation to zero. Then the theoretical and experimental equations could be compared. Pan equations had 10% error between the theoretical and experimental values for their slope and 9% error in intercept. Tilt equations matched better, with only 2% error in slope and 3% error in intercept values.

B. Calibration and Testing with the Simulator

The data collected for each range and heading confirmed the derived equations that would relate Cognachrome™ output to object position. We wanted to verify that these equations, created from theory and checked with data taken from static readings, would still hold when the system was in motion. To this end, we created two test trajectories to represent an astronaut moving around the stationary rover on a flat surface. Each isolated one of our coordinate system components, range and heading. In the range test, the tracked object moved away from the camera and then back towards it, as if the astronaut were walking away from, and then towards, the rover. In the heading test, the object stayed a fixed distance from the camera, but the camera panned back and forth, as if the astronaut were circling the rover. Both tests showed similar trends with respect to velocity and length of test path. Results were always best for longer, smoother paths.

For heading change tests, range was held constant as the pan angle and pan speed were varied. Tests were done at ranges from 1.0–5.0 m. Pan angles ranged from 3.0–9.0 degrees, and pan speeds were varied to simulate an astronaut walking at 0.10–1.00 m/s. The PTU's zero position was dead ahead; we had it pan from the positive pan angle to the negative pan angle for a symmetric motion. We found that longer, smoother pan maneuvers resulted in less offset error. Figure 8 shows an example results from a test done with 3.0 m separation, a pan angle of ±9.0 degrees, and an astronaut walking speed of 0.5 m/s. The heading tracking was much less sensitive to changes in speed than the range tracking. When the astronaut's walking speed was increased to 1.0 m/s, there was scarcely any change at all in the results. This indicated that heading tracking performance depends more on a trajectory's smoothness than on its speed.

The next series of tests involved change in range only. Changes in range varied from 0.5–4.0 m, and speeds of 0.05–0.32 m/s were used. Figure 9 shows the cart traversing the rail from 1.0 m to 3.0 m at a speed of 0.10 m/s. Data from the motor encoders is overlaid with the Cognachrome™'s estimation of actual tracked object location. The Cognachrome™ data is the upper of the two waveforms in all figures.

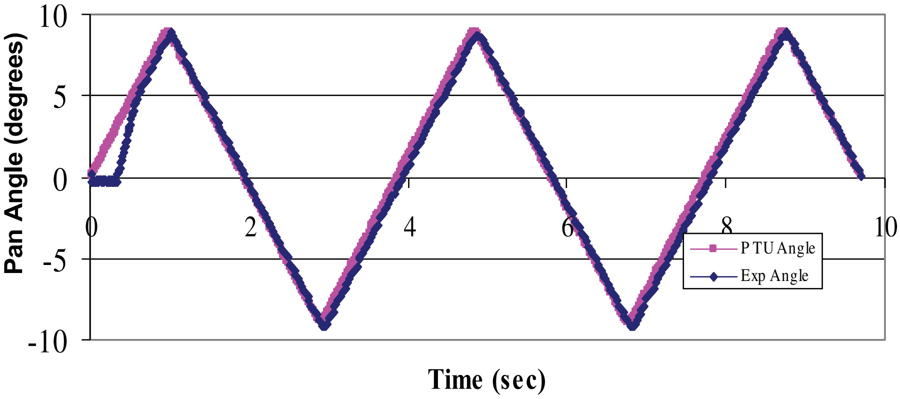


Fig. 8 Simulator nine degree pan test.

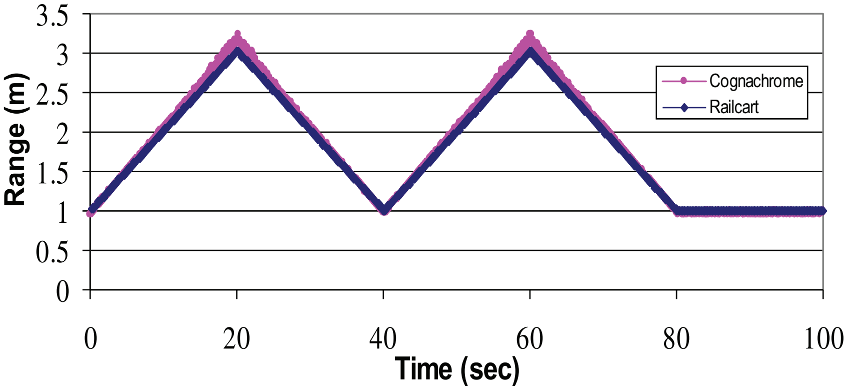


Fig. 9 Simulator range test.

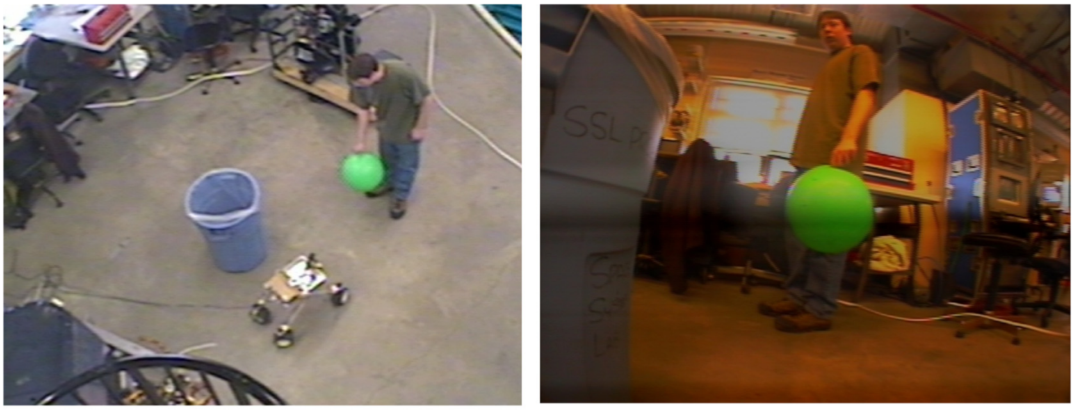
V. Rover Following Experiments

To verify the ability of our color tracking system to follow its target under nominal and exceptional situations, we mounted our camera to a rover atop an onboard PTU. Below, we describe the experimental setup and results from our rover following tests that focus on following performance and the use of the Fig. 1 and Fig. 2 autonomous target acquisition and exception handling protocols.

A. Rover Experimental Platform

The rover used for our tests (Fig. 10a) is open-loop controlled via RS/232 link and actuated with ten high-torque R/C servos that independently drive and steer each of four wheels (8 servos) as well as the PTU (2 servos). Prior to vision system tests, a dead reckoning controller was implemented on the rover and manually tuned for the concrete lab floor. With this smooth but somewhat low-friction surface, dead reckoning was determined an acceptable form of navigation over short distances (~3 meters or less), but slippage errors accumulated significantly for longer open-loop traversal distances, particularly when turns were required. Given accumulating dead reckoning errors, a traversal "map" incrementally constructed by the rover is not expected to be accurate relative to inertial coordinates. However, for our tests, the rover was instructed to follow its visual target (astronaut) at a distance of ~1 m, a range well within the capabilities of our dead reckoning system as will be demonstrated below.

For these experiments, the rover's sole sensory input is astronaut (target) location. In addition to the onboard camera, a camera was mounted overhead to measure rover performance independent of the onboard data source. Figures 10a and 10b show typical overhead and onboard views, respectively. Control of the PTU was decoupled from

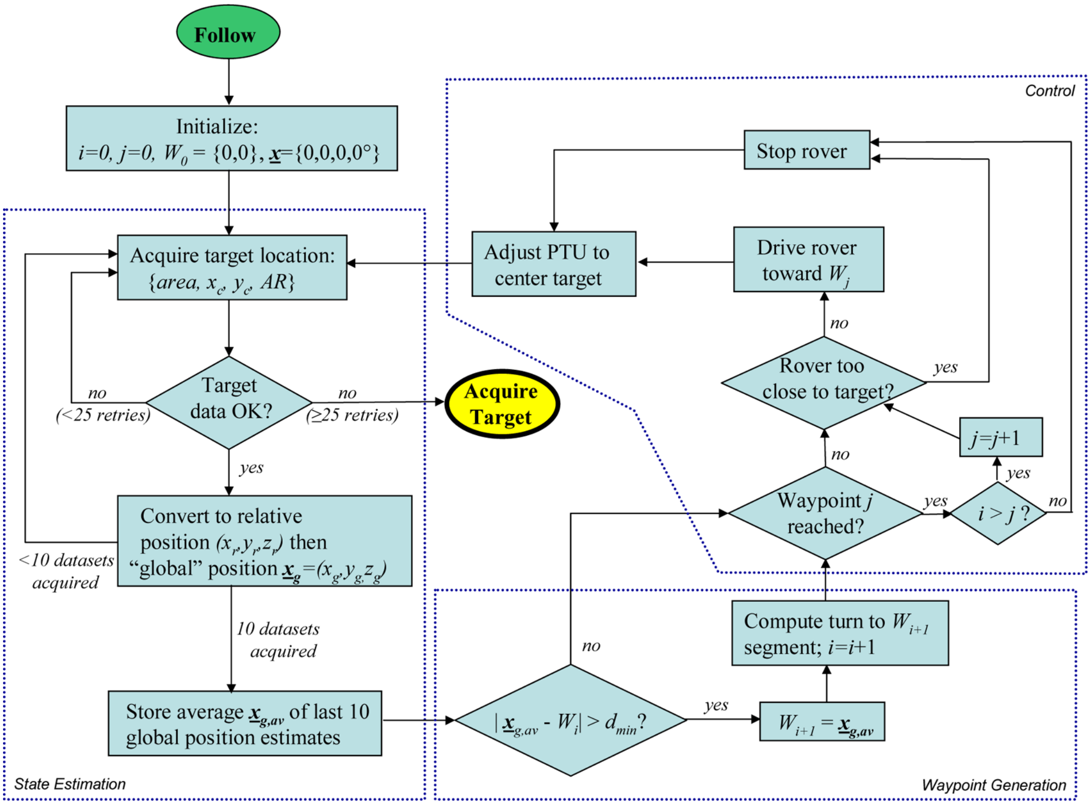


a) Overhead View

b) Onboard View

Fig. 10 Onboard rover and overhead camera views.

the vehicle dead reckoning controller so the vehicle could independently follow the traversal path while the PTU tracked or searched for the target. Although no onboard sensor data is available to determine inertial heading, target centroid plus commanded pan angle and rover heading changes enable determination of the target's relative bearing. Range to the target is determined by color blob (target) area. As described above, our rover makes the assumption the astronaut is following a traversable (obstacle-free) path, and that it must follow this same path rather than take



Follow
Control
Initialize:
i=0, j=0, W_0 = {0,0}, $\underline{x}$={0,0,0,0°}
Stop rover
Acquire target location:
{$area$, x_c, y_c, AR}
Adjust PTU to center target
Drive rover toward W_j
no
Rover too close to target?
yes
no
(<25 retries)
Target data OK?
no
(≥25 retries)
Acquire Target
j=j+1
yes
no
yes
Waypoint j reached?
yes
i > j ?
no
Convert to relative position (x_r,y_r,z_r) then "global" position $\underline{x}_g$=(x_g,y_g,z_g)
<10 datasets acquired
10 datasets acquired
Compute turn to W_{i+1} segment; i=i+1
no
Store average $\underline{x}_{g,av}$ of last 10 global position estimates
$|\underline{x}_{g,av}$ - $W_i| > d_{min}$?
yes
W_{i+1} = $\underline{x}_{g,av}$
State Estimation
Waypoint Generation

Fig. 11 Astronaut following procedure.

"short-cuts" to reduce path length. A dynamic waypoint map was generated in real-time so that the rover followed the astronaut's actual path rather than moving straight toward the astronaut at each time step. Figure 11 outlines the processes executed during all "Follow" operations. Position vectors are defined as $\underline{x} = \{x, y, z\}$ plus heading ψ, while each waypoint W_i is defined by (x_i, y_i). Although the rover traverses a 2-D surface, identification of the target z coordinate enables the PTU to track vertical target displacement.

The Fig. 11 astronaut following algorithm has three phases: state estimation (relative navigation), waypoint generation, and control. In the figure, indices i and j denote current astronaut and rover waypoint locations, respectively. First, data is collected from the onboard camera, including target *area*, centroid $\{x_c, y_c\}$, and aspect ratio AR. Target data is validated based on *area* and AR. If no valid data is found in 25 retries, the target is considered lost and the target acquisition procedure (Fig. 1) is invoked. Otherwise, the image plane information is converted to relative then global position coordinates, where "global" coordinates are defined from the rover's startup position and heading. Since data is rapidly acquired (∼30 Hz) and the rover dynamics are relatively slow, we apply a primitive filter that averages 10 datasets into a single position estimate $x_{g,av}$. If this estimate is sufficiently far from the previously stored waypoint W_i, a new waypoint W_{i+1} is generated, and a smooth turn is inserted to connect path segment $\overline{W_{i-1}W_i}$ to $\overline{W_i W_{i+1}}$. The control software executes next. If the last waypoint W_j has been reached, the rover is driven (and turned) toward the next waypoint W_{j+1}. In this implementation, if either no more waypoints are available ($i = j$) or the rover is too close to its astronaut companion, it will temporarily halt. Based on any commanded turn and the previously acquired target centroid, the PTU will be adjusted to center the target on the image plane. This process continues until interrupted by the user or until the target is lost.

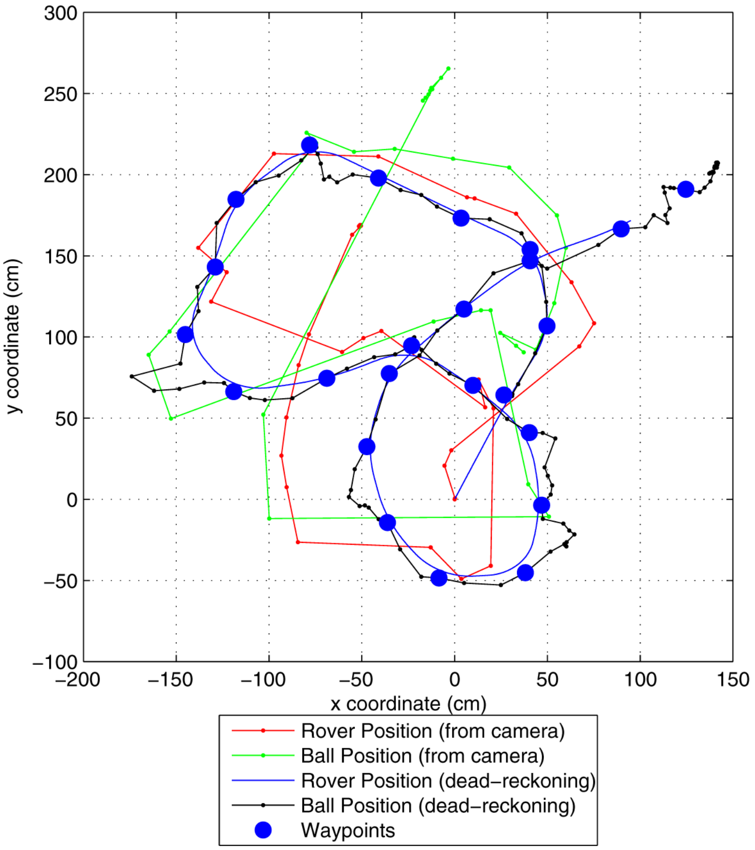
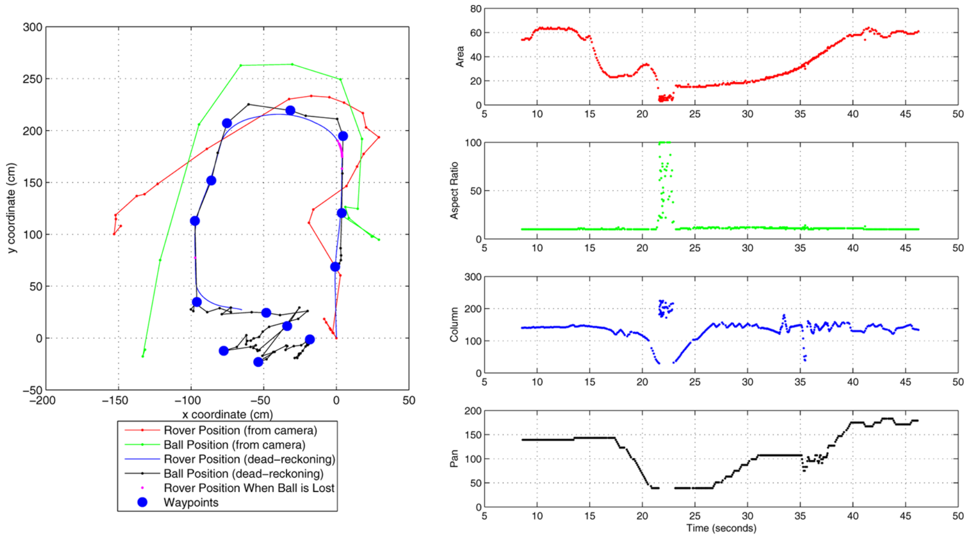


Fig. 12 Nominal following test.



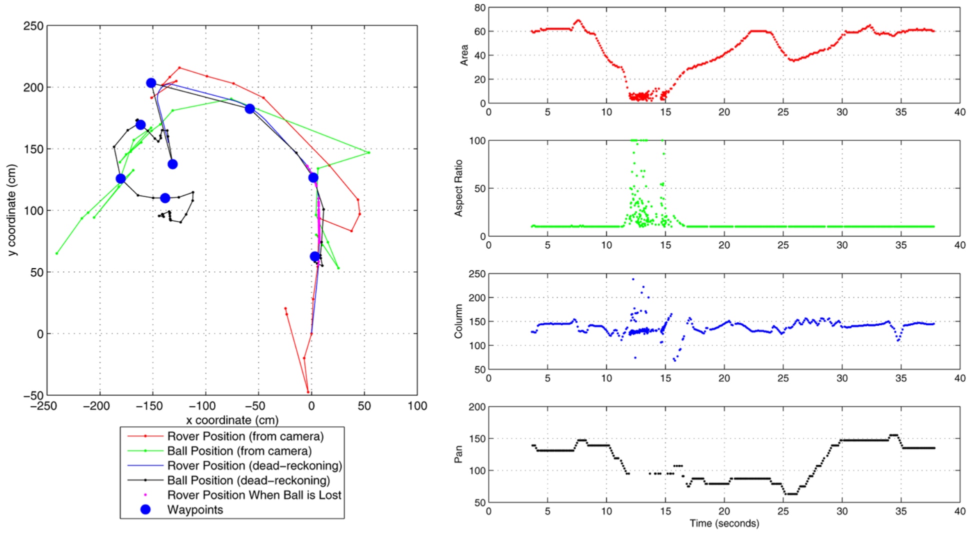
a) Overhead Traversal Map

b) Cognachrome™ + Pan: Target moves out of range at *t* =22 sec

Fig. 13 Astronaut makes a U-turn.

B. Experimental Results

Results are presented that illustrate rover performance during following (Fig. 11) and target acquisition (Fig. 1) operations. All tests were performed in the laboratory environment shown in Fig. 10, and plots indicate the rover's dead reckoning map data plus (when indicated) "actual" astronaut and rover paths as determined from the overhead video. Figure 12 and **Video 1** illustrate a nominal following sequence. The rover begins at station (0, 0), defining the inertial coordinate system from its initial position and heading. The rover places regularly-spaced waypoints on the observed target path (indicated by blue dots in the figure) and follows a path between these waypoints approximately 1 m behind the astronaut. Generally, the rover believes it has accurately followed the astronaut, with good agreement between dead-reckoning estimates of ball position and rover position over time. The most significant exception



a) Overhead Traversal Map

b) Cognachrome™ + Pan: Target moves behind obstacle at *t* =12 sec

Fig. 14 Astronaut moves behind an obstacle.

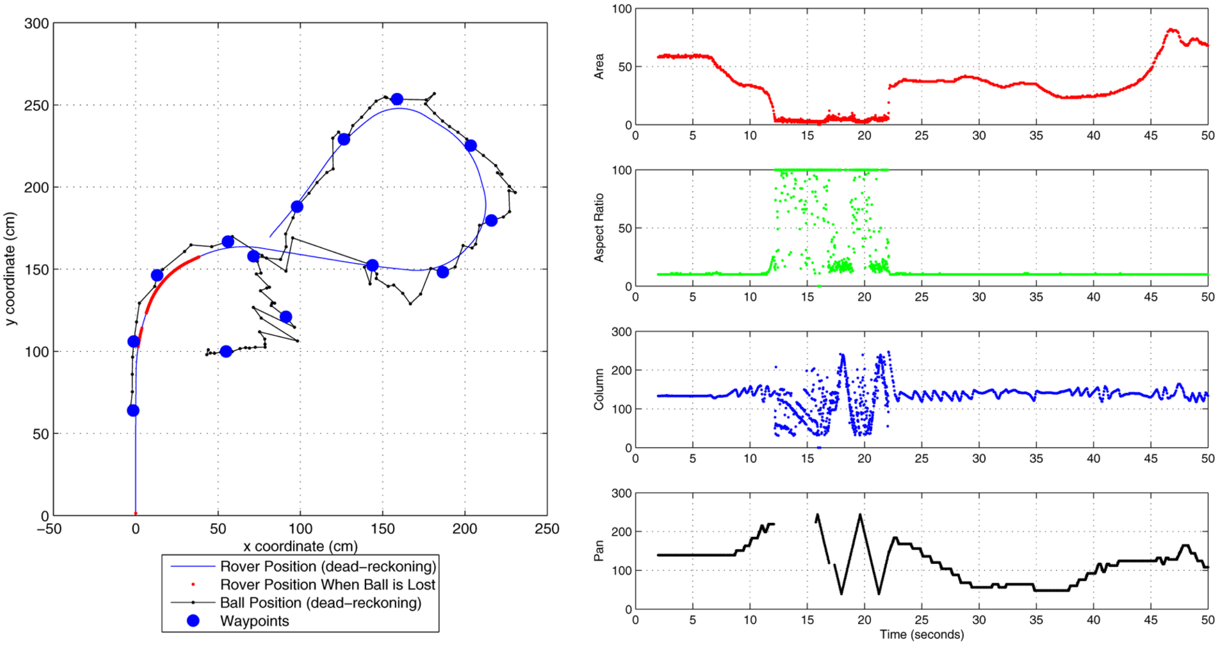
is where the astronaut makes a sharp turn near coordinates (−150, 100). Around this area, the waypoint spacing $d_{min} = 0.6\,\text{m}$ is sufficiently large that no intermediate waypoints are inserted. While d_{min} could be decreased to increase rover adherence to the apparent path of the astronaut, this may not always be desired. A lower d_{min} would force the rover to take sharper turns, which slows it down, contributes to dead-reckoning errors, and increases the probability that the astronaut will drift outside of pan range. In addition, the rover's planned trajectory becomes more influenced by noise in the astronaut's apparent position. This illustrates the importance of the d_{min} value to both rover and astronaut: the astronaut must make sure areas within d_{min} of the traversed path are clear.

To validate the rover's following capability, for this nominal test we also processed the overhead video. As expected, dead reckoning errors accumulated over time, resulting in significant discrepancies (>1 m) near the end of the test between dead reckoning and "real" datasets from the overhead camera. Despite these differences, however, this data does show the rover did actually follow the astronaut's path reasonably well, except for the area near the (−150,100) corner where even dead reckoning indicated the rover planned a "short cut" as discussed above.

Figures 13–15 illustrate cases in which the target was lost and reacquired during the test. In Fig. 13 and **Video 2**, the astronaut executes a U-turn maneuver that makes the target leave the rover's field of view. This occurs because the pan servo prevents the rover from looking behind itself, and the rover must proceed along the astronaut's trajectory until it is able to make the U-turn itself. Figure 13b shows the vision data and pan angle commands for this test. As can be seen, area and aspect ratio are clear indicators that the target has left the image plane, with the pan commanded to its minimum angle (left turn) limit until the rover has itself rounded the corner at which time it is able re-acquire the image. In this case, target re-acquisition occurs solely within the Fig. 1 protocol, since the rover is able to again view the astronaut before reaching the end of its known waypoint sequence.

Figure 14 and **Video 3** show a test in which an obstacle (a large trash can) was placed in the center of the traversal area. In this case, the astronaut moves behind the obstacle, with a rapid period of area decrease followed by loss of target as indicated by the low area and high aspect ratio data. The rover again continues the traversal to its last known waypoint, by which time it travels around the obstacle and again finds the target.

In the last test (Fig. 15 and **Video 4**), the astronaut again goes behind an obstacle, but remains out of view until after the rover stops at its last known waypoint. In this case, the rover cannot acquire the target while continuing



a) Overhead Traversal Map

b) Cognachrome™ + Pan: Rover searches for ball from t =16-23 sec

Fig. 15 Search for the astronaut.

its traversal, thus it executes a "stop" command (Fig. 2). Since we did not provide any input, the rover repeatedly broadcasts its situation and executes its target search pattern (a back and forth sweep as shown in Fig. 15) until the astronaut moves back into the image plane. Once the target is reacquired, the rover switches to normal following mode and continues its traverse.

Our rover tests focused on the suite of autonomous behaviors present in our modal operations model except for the broadcast of exception conditions. We have experimentally validated the target acquisition sequences through extensive testing, and hope to explore in more detail the interaction between the rover and its operator/astronaut companion in future work. Perhaps the most significant finding from these tests was that the rover could, in fact, successfully follow and reacquire its target despite poor dead reckoning accuracy and no feedback beyond color tracking. This mode of operation can enable extremely low-cost, light-weight "follower" rovers to team both with astronauts and other better-equipped rovers as companions.

VI. Conclusions

In this paper, we have presented an application of color blob tracking to the astronaut following problem. The method is fast and capable of accurately tracking an object moving at human walking speeds. For a Moon or Mars rover equipped with cameras as its primary sensor suite, visual tracking is an attractive option. The position information gleaned from the camera(s) can be used in a control system to enable the rover to follow the astronaut over unknown terrain. In this way, we take advantage of the astronaut's superior navigation and planning abilities and free the rover from a time-consuming analysis of the local terrain and its hazards. Since the path the astronaut has taken is known to be safe, the rover can follow at velocities on the order of human walking speed, allowing it to keep pace with its human teammate. Should the rover lose sight of the human, either because of lighting conditions or because of other factors, mode-switching operations can enable it to determine its next best course of action.

We have installed the color tracking system on a prototype rover with PTU camera mount system. We have implemented the software necessary to build and follow waypoint-based following trajectories with dead reckoning navigation, sufficient for our relatively close (1–2 m) astronaut-rover separation distances. Software to implement the modal operations paradigm for robust recovery from exceptional situations has been demonstrated, indicating the rover can in fact either re-acquire its target or safely request assistance from its astronaut companion.

We have plans to further test the spectrum of target loss situations with our rover, both in the controlled lab environment and outdoors where lighting conditions are much more variable. During outdoor testing, we can also utilize GPS to provide data to independently assess the rover's following capabilities. Long-term, we intend to incorporate this knowledge into a planning/scheduling agent controlling high-level autonomous and semi-autonomous behaviors of which following is an important subset.

Although our target application was astronaut following on a planetary surface without GPS, the modal operations paradigm is not unique to an extraterrestrial environment or an astronaut target. A low-cost rover could track a "leader" rover equipped with more sophisticated high-speed navigation capabilities. Reduced time and cost could also be realized with Earth-based robotic platforms acting as high-speed followers of manned or unmanned vehicles, particularly in obscured areas where GPS coverage is not available. Application to three-dimensional space or undersea environments without GPS is also possible, provided the vision system is extended (e.g., to a stereo pair) to resolve ambiguities in six degree-of-freedom target motion.

References

[1] Fong, T. W. and Nourbakhsh, I., "Interaction Challenges in Human-Robot Space Exploration," *Interactions*, Vol. 12, No. 2, Mar. 2005, pp. 42–45.

[2] Hoffman, S. J., and Kaplan, D. I., (eds). *Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team*. NASA Special Publication 6107, Johnson Space Center: Houston, TX, July 1997.

[3] Washington, R., Golden, K. et al., "Autonomous Rovers for Mars Exploration," *Proceedings of the 1999 IEEE Aerospace Conference*, 1999.

[4] Kosmo, J., Trevino, R., and Ross, A., *Results and Findings of the Astronaut-Rover (ASRO) Remote Field Test Site*. JSC Doc. No. 39261, Johnson Space Center: Houston, TX, Mar. 1999.

[5] Patterson, D. A., "President's Letter: Robots in the Desert: A Research Parable for our Times," *Communications of the ACM*, Vol. 48, No. 12, Association for Computing Machinery (ACM), New York, NY, 2005, pp. 31–33.

[6]Nourbakhsh, I., Kunz, C., Fluckiger, L., Schreiner, J., Ambrose, R., Burridge, R., Simmons, R., Hiatt, L., Schultz, A., Trafton, J., Bugajska, M., and Scholtz, J., "The Peer-to-Peer Human-Robot Interaction Project," *Proceedings of AIAA Space 2005,* Long Beach, CA, 2005.

[7]Simmons, R., Krotkov, E. et al., "Mixed-Mode Control of Navigation for a Lunar Rover," *Proceedings of the SSI/Princeton Space Manufacturing Conference*, 1995.

[8]Krotkov, E., Simmons, R., Cozman, F., and Koeing, S., "Safeguarded Teleoperation for Lunar Rovers: From Human Factors to Field Trials," *IEEE Planetary Rover Technology and Systems Workshop*, Apr. 1996.

[9]Magee, K., *Vision Based Logic and Neural Network Approach to the Control of Hyper-Redundant Robot Manipulators*. Ph.D. Dissertation, Rice University, Houston, Texas, May 1994.

[10]Bowling, M., and Veloso, M., "Motion Control in Dynamic Multi-Robot Environments," *Proceedings of the1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, November, 1999.

[11]Bailey, B., Reese, J. et al., "Robots with a Vision: Using the Cognachrome Vision System," *Circuit Cellar Ink,* Issue 92, March 1998.

[12]Craig, J. J., *Introduction to Robotics,* Second Edition, Addison-Wesley, Reading, Massachusetts, 1989.

Amy Pritchett
Associate Editor