AUTONOMOUS SPACECRAFT DOCKING USING MULTI-COLOR TARGETS

David P. Miller and Anne Wright
The MITRE Corporation
M/S Z459 7525 Colshire Dr.
McLean, Virginia 22102 USA
dpmiller@starbase.mitre.org or anarch@ai.mit.edu

Abstract

Spacecraft docking requires precision alignment. Traditionally this has been done by astronauts using alignment sights, or offset retro-reflective targets. More recently, teleoperated systems using the same sort of targets have been used. Most experiments with completely autonomous docking systems also rely on similar targets. These systems require high-acuity vision systems, and are computationally expensive. They are also subject to errors and misalignments from the tracking system getting confused by reflective debris, or other spacecraft markings. This paper describes a new docking system currently undergoing testing. This system uses three offset color markers. The markers use bright colors which are ordinarily not seen in an orbital setting. The system tracks the colors at frame rates, and can maintain its lock during changes in lighting conditions. The geometrical and tracking calculations are sufficiently simple that they can all be run on a low-cost commercial micro-controller. Experiments docking a mobile robot, and full 6-DoF experiments planned for a neutral buoyancy tank, are described.

1 Introduction

In-orbit repair of satellites has the potential to be a major cost-savings for the communications and intelligence industries. Quite often a satellite becomes inoperable simply because it has exhausted its fuel or batteries, or because it never correctly deployed in the first place. These repairs are relatively simple to undertake, the most challenging part is getting the repair-person to the satellite. Towards this end, NASA initiated the *Flight Tele-robot Servicer* program. This free flying high dexterity tele-robot would be able to change orbit, approach satellites and carry out repairs and refurbishment both simple and complex.

Unfortunately, the FTS program proved to costly, and was dropped by NASA a few years ago. But the need remains. The *Ranger* program being led by the University of Maryland, is an attempt to recreate the critical capabilities that would have been provided by the FTS, but at a small fraction of the cost. One of the places where cost cutting is taking place is in the communications system. The robot will be operated only when it is in line of sight of its primary ground station. Therefore operation time is reduced. To increase the capabilities of the system, it is desirable that time-consuming, but routine procedures be automated wherever possible. One such procedure is the approach and docking of the robot with the satellite which is to be serviced.

Any automation that is added to Ranger must follow the basic policies of the program. The systems added must be low-mass, low-power, and low-cost. This paper describes an automated docking system currently under design and test for use on the Ranger spacecraft. This system uses the existing Ranger camera and control system, and adds only a small micro-controller to the spacecraft's hardware. The system relies on color tracking, and can track on any unique color set in the environment. The system described in the paper uses a specific target to maximize the accuracy of the control feedback coming from the color tracker. However, any unique color patch on the target spacecraft could be used for guiding Ranger in for its rendezvous.

The remainder of this paper describes the target system used, the color tracking hardware and software, and the experiments that are currently in progress and planned for the near future.

2 Target Design

A docking target ideally can orient a robot in all six degrees of freedom. This provides the robot with

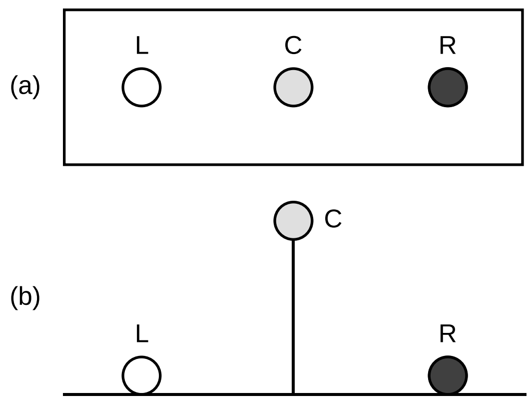

Figure 1: Target Arrangement as Seen (a) From Front (b) and Below

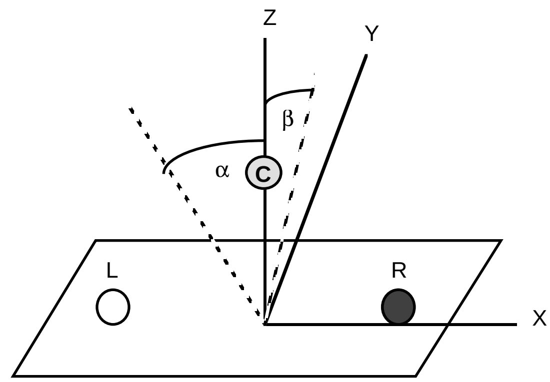

Figure 2: Coordinate System for Target

the feedback needed to completely and unambiguously plan its trajectory. In practice, precisely calculating the complete position and orientation of the robot is not necessary. Range information need be known only approximately, as some contact speed is usually needed to engage the docking mechanism, and docking speeds are normally quite low. Depending on the specifics of the docking mechanism, some error in rotation and translation can also be tolerated, as can errors in pitch and yaw. What is most important is that the errors be reduced to acceptable limits as the robot comes closer to the final docking position. The docking target and the target sensor must work together in this way.

For this project, the primary sensor is a color tracking system (described in detail in Section 3). This system can track the positions of three color blobs, and returns the image plane coordinates of the centroids of those blocks. To reduce dependence on getting precise position and orientation information out of the sensor, the sensor is part of a reactive feedback loop that controls the robot's motion [1, 2]. Basically, only relative information is needed (i.e., is the robot more or less to the left than it was a moment ago).

The target system we are using consists of three colored spheres (L, C, & R) against a white background. The balls L & R are mounted against the background separated by a distance of 2. The middle sphere is mounted on a white post of length 1. The post is mounted at the center of the line segment defined by the other two spheres (see Figure 1).

This arrangement makes it trivial to roughly calculate the robots position and orientation relative to the target. If the target coordinate system is set up with L & R at $(-1, 0, 0)$ and $(1, 0, 0)$ respectively then the robots roll is found by calculating the slope of line $\overline{LR}$. Assuming the robot is on or near the Z axis means that pitch and yaw can be determined by the position of sphere C relative to the center of the image. The robot's angular displacement off the Z axis, above or below the X axis can be approximated by

$$\beta = \sin^{-1} \frac{2\overline{cC}}{\overline{LR}}$$

where $\overline{cC}$ is the distance in image coordinates from ball C to the line defined by L & R and $\overline{LR}$ is the distance in image coordinates separating L & R (see Figure 2). Displacement off the Z axis to the left or right of the Y axis can be determined by comparing the lengths $\overline{LC}$ and $\overline{CR}$. If the angle is greater then $\pi/4$ then the balls will appear out of order. If the displacement is less then $\pi/4$, the angle can be calculated as:

$$\alpha = \sin^{-1} \frac{\overline{CR} - \overline{LC}}{\overline{LR}\sqrt{2}}.$$

An approximation of the distance from the target can be gathered by comparing the apparent separation of L & R with the separation found in a test image at a known distance. The exact function of image plane separation as a function of distance is dependent on the particular optics of the camera, but the function is roughly inversely proportional. Of course this measurement is only accurate when the robot is on or near the Z axis.

3 Vision Processing

In order for a robot to align itself to the target described in Section 2, the robot must be able to quickly track the positions of three colored balls. To accomplish this, we use the Cognachrome 2000 video processing system.

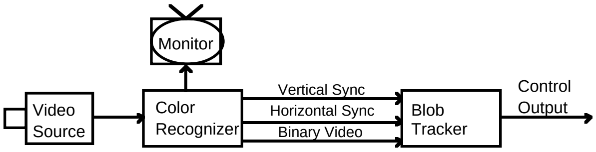

Figure 3: Block Diagram of Cognachrome 2000 organization

The Cognachrome 2000 was designed to be a visual sensor suitable for providing feedback in real-time control tasks, and particularly for control of small autonomous systems. Cost, compatibility, and parts replaceability were also important issues, so the system uses standard video input formats and many off-the-shelf components. I call it a visual sensor because it does not try to solve the general vision problem. Instead, it is optimized for high-speed, low-latency tracking of colored objects. Because it is specifically designed for this purpose, it is able to outperform more general systems and use less hardware on simple tracking tasks as are needed in spacecraft docking.

3.1 System Organization

The Cognachrome 2000 consists of two logically and physically distinct units – the color recognizer and the blob tracker. Figure 3 shows the block diagram for the system. The color recognizer inputs standard color NTSC video, which is the format used by standard video equipment in North America, and outputs a corresponding binary image and synchronization signals. The blob tracker captures these signals as timing information, decomposes the image into a number of spatially distinct "blobs," computes tracking information for the blobs, and takes appropriate action, such as transmitting navigation instructions over a serial port.

3.2 Color Recognizer

In order to create binary images from a color video signal, the color recognizer first digitizes the video into 24-bit digital RGB, then thresholds the digital RGB stream using a look-up table. As digital RGB, each pixel is expressed using three 8-bit numbers representing the amount of red, green, and blue present in the corresponding location in the image. If you think of the red, green, and blue components of a color as defining three orthogonal axes, the color value of each pixel defines a specific point inside an RGB cube. The look-up table partitions the cube into two regions, corresponding to the volume of the cube that may be occupied by the target color and the volume that will not. If such a partitioning can be defined which completely separates the target color region from the region of the background colors, the resulting binary images will be uniformly one for the pixels corresponding to the target object and uniformly zero everywhere else. For ease of reference, the pixels in the target partition will be referred to as "white," and the pixels in the background partition will be referred to as "black," as this is how they appear in the debugging output of the system. For instance, if orange is the target color, an orange square on an arbitrary background will appear as a white square on a black background to the blob tracker. The Cognachrome 2000 actually has three independent color channels in the look-up table, so three distinct colors can be tracked in parallel.

In arbitrary environments, it is often difficult to define such a partitioning exactly. Therefore it is important to select target colors which are maximally different from the background, for instance bright primary colors tend to work particularly well. The choice of target color should also take into account the particular nature of the environment in which the system will be used. For instance, while green may be a fine choice for indoor use, it does not work well as a target color outdoors near grass and trees. In the environment of space, there are few colors other than black and white which occur naturally, so the problem is in some ways easier. However, the harsh lighting conditions may still cause the targets to be transiently obscured, as a colorful target is indistinguishable from the blackness of space if it is in a shadow and from the white of the spacecraft if it is in a glare.

3.3 Blob Tracker

Whereas the color recognizer board is a custom design, the blob tracker is implemented using a commercially available processor board using the Motorola 68332 micro-controller. The 68332 is particularly suitable for the task of capturing the timing of binary video because it includes an independent microcoded Timer Processor Unit (TPU) which communicates with the processor using shared memory and interrupts. The task of capturing the timing signal and converting it to spatial data is performed by the TPU. The raw data available to the processor is the row and column numbers of black-to-white and white-to-black transitions in the binary image. I call this format "endpoint format" because it only records the endpoints of each horizontal white segment in the image, the collection of which corresponds to all the white area in the image. For typical images, this raw data format is more compact and can be more efficiently processed into distinct blobs than the more commonly used formats of a pixel array, which represents every pixel in the image as a number in a grid, or a list of all the white pixels.

Given the set of endpoints in an image, the blob tracker groups the endpoints into blobs on the basis of spatial adjacency [3]. Generally, each blob corresponds to an object of the target color in the real world, though a single physical object may be split into multiple blobs if there is a line of another color cutting completely through it. The system's ability to distinguish between an arbitrary number of blobs aids in noise immunity because even in a noisy image heuristics can usually be used to distinguish which blobs are interesting and which are not. Common heuristics used are that the expected size of the target blobs are larger than the others or that location of the target blobs should be close to where they were during the last frame because their maximum speed is limited. Features of the target blobs can then be computed and used to make control decisions. Such features as area, position, orientation, aspect ratio, and relative position to blobs of other colors can easily computed from the endpoints list.

3.4 Performance Specifications

The primary performance specifications for this system are resolution, frame rate, and latency. The spatial resolution of the tracking data is 256 rows by 250 columns. The frame rate of NTSC video is 30 or 60 Hertz depending on whether interlace is used or not. Most video cameras do not actually use interlace, and therefore provide frames at 60 Hertz, or 16.7 ms/frame. The frame rate of the system depends on how long end of frame processing takes. Converting from endpoint to blob data typically takes 4 ms per frame and happens in parallel with frame acquisition as data is copied from the TPU ram to the processor ram. Once the end of the frame is reached any remaining data conversion is completed, the blob data is processed, and control action is taken. If this end of frame processing takes 12 ms or less, the effective frame rate will be 60 Hz. If the processing takes longer, frames will be missed and the frame rate will decrease. The processing time available is 28 ms for 30 Hz, 44 ms for 15 Hz, etc. The latency calculations are very similar. The length time from the beginning of the frame to the completion of the control task is 16.7 ms plus the end of frame processing time. The processing tasks attempted so far have typically fallen within the 12 ms processing window, and therefore have a 60 Hz frame rate and have 17 to 29 ms of latency, though it remains to be seen precisely how long the final processing for this project will take.

3.5 Hardware

The color recognizer and blob tracker boards stack together for a total size of 2.5"x6.25"x1.25". The color recognizer board consists of a Sony NTSC to RGB converter module, a RAM to implement the look-up table, an 8K serial EEPROM for non-volatile storage of parameters, interface circuitry, and connectors for video in, video out, and power. The blob tracker board is a stand-alone Motorola 68332 board available from Vesta Technologies. The 68332 board typically has 128K of ROM, and 256K of RAM, and runs at 16MHz. The processor has no disk or other secondary storage. The primary means of communicating with the processor is over an asynchronous serial port. The processor runs a custom real-time operating system for embedded controllers written by Anne Wright, and provides the control signals for the color recognizer as well as processing the images and using the processed data to provide control output.

4 Experiment Design

The eventual goal of this work is to allow the Ranger spacecraft to be able to autonomously rendezvous and dock with the target spacecraft on which it is to work. Prior to the flight experiment, there are three sets of ground tests that we are planning.

4.1 Orienting a Mobile Robot

The vision system has been integrated onto *Lore*, a mobile research robot at MITRE. Lore is a B21 *Real World Interface* robot base augmented with a *Sony XC999* color camera and the Cognachrome vision system. The camera is mounted on the top of the robot and pointed straight up.

On the ceiling of the test area is a target of the type described in section 2. The robot executes a search pattern until the target comes within the field of view. Once this happens, the robot maneuvers until it is directly under the target, and then corrects its rotation.

This experiment is by no means a complete test of the tracking system. The robot's rotation and translation degrees of freedom are interlinked. Additionally, the pitch and yaw are fixed. As a result, it is possible to navigate to directly under the target simply by trying to center the target in the image frame. However, to make the system more realistic, the algorithm used to direct the robot relies on the relative positions of the three colored spheres — the same feedback that a free-flying robot would get.

4.2 On the Deck with MPOD

The next set of experiments we plan to undertake is to transition the tracking system into the control stream of MPOD. MPOD is an underwater test vehicle at the University of Maryland. MPOD has six degrees of freedom using paired sets of thrusters. The system returns video and telemetry and takes commands through a cable leading out of the test tank into a bank of workstations on the deck.

Our plan is to insert the video and tracking hardware in line with the operator control station. Unless the operator overrides, the tracking system will issue commands that from the robot's perspective will appears as ordinary operator commands. The video from the robot will feed directly into the tracking system. A target will be mounted on a support structure near one wall of the test tank. Initially, the robot will be pointed at the target, and each colored ball will be manually identified. The robot will then be spun in a random orientation and instructed to dock. The robot will begin a search rotation until the target is sited. MPOD will then close in on the target, taking a course halfway between straight to the target and perpendicular to the target until the robot is near the Z axis.

In the underwater tests, it is important that the robot not move a significant distance towards or away from the target unless the target is in the field of view of the camera. Since the tracking system relies on colors, and water absorbs different colors differently, there is a significant color shift in the apparent color of each target ball as the distance between the target and robot changes. This color shift can automatically be compensated for by the tracking software as long as the target remains in constant view. Tracking calculations are interleaved with updates to the color table to keep the target colors close to the colors currently being seen by the camera. In this sense, tracking in space will be easier then tracking in the test tank.

4.3 Tracking for Wet-Ranger

Wet-Ranger is a functionally equivalent vehicle to the flight Ranger Scheduled to be launched in late 1996. The electronics and manipulator sections of the two robots are almost identical. The propulsive modules are of course completely different. Wet-Ranger uses a thruster set similar in design to MPOD.

Our docking tests with Wet-Ranger will be similar to those to be done on MPOD. The primary difference is that on Wet-Ranger, all the electronics will be onboard. The electronics package we are using fits in a $7 \times 2.5 \times 2$" box and masses about eight ounces. There is ample room in the current design of the electronics compartment on Wet-Ranger to house this extra hardware. Once again, the tracking system will issue commands similar to those provided by the operator, The operator can switch the tracking system on and off line.

4.4 Current Status

The basic tracking hardware and software have been constructed, and the system has been installed on Lore. Lore has been tested with monochrome targets, and can easily follow them even when the target is moved at fairly high speed. By keeping track of where the target was last seen in the frame, the robot can track the target smoothly even if the target moves out of view.

We have also tested the tracking system using the tank cameras at the University of Maryland's Space System Laboratory neutral buoyancy tank. During these tests, a diver carried a colored marker towards and away from the camera. The purpose of these tests was to verify that the tracking software could compensate for the color shift due to the frequency absorption of the water. The tests were successful and three colors were identified that are distinct from one another, and provide robust tracking over a large portion of the tank.

We have also tested software which tracks the two end colors of the target and returns orientation information to the control stream. This has proven robust and stable even when the camera is close to the target and the colored markers have a large extent on the image plane.

We are currently debugging the full target tracking system on Lore, and expect to have that operational in late '94. Tests with MPOD and Ranger will start in early '95.

5 Conclusions

Using color tracking for target docking has certain advantages over more traditional space tracking systems. Since the system is passive, it is much lower power then a ranging system. Premounted targets on the target vehicle are not necessary, though desirable. The targets are simple and inexpensive. The tracking system is also inexpensive and has a minimum of impact on the other spacecraft systems. Ranger is a particularly good fit for this kind of system. The teleoperation interface provides a natural place for the automated docking system to tie in to the spacecraft's control system.

While not yet ported to Ranger, early tests on our mobile platform Lore indicate that the tracking system can provide the necessary guidance to successfully move the spacecraft in the desired way. Our tests in the neutral buoyancy tank have shown that the system is adaptive enough to handle color shift due to absorption by water. This color shift is much greater then would be encountered by Earthshine or other likely color changes that would be found in orbit. Color tracking appears to be a low-cost low-impact method of adding automation to the Ranger system

Acknowledgments

The authors wish to thank Randy Sargent, Marc Slack, Dave Akin and the crew of the SSL for their help in this research. This work was supported by the MITRE Corporation internal research funds.

References

[1] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[2] D. P. Miller. Rover navigation through behavior modification. In *Proceedings of the NASA Conference on Spacecraft Operations Automation and Robotics*, July 1990.

[3] A. Wright. A High-speed Low-latency Portable Visual Sensing System. *Proceedings of the SPIE symposium on optical tools for manufacturing and advanced automation*, September 1993.