Final Report

NASA Reduced Gravity Student Flight Opportunities Program 2003

GYRE: Evaluation of Visual Navigation Techniques for Autonomous Free-Flying Robots

in the field of
Free-Flying Robotic Navigation

submitted by

The GYRE Project
University of Washington
Seattle, Washington 98195

Contact:
David Bliss
+1 206 860 3514
dbliss@u.washington.edu

Faculty Supervisor:
Linda G. Bushnell
+1 206 221 6717
bushnell@ee.washington.edu

Team Members:

David Bliss, Flyer
Junior, Chemistry
dbliss@u.washington.edu

J. Lee Zeman, Ground Crew
Senior, Computer Science
leezee@cs.washington.edu

Paul Saitta, Flyer
Senior, Computer Science
pbs@cwru.edu

Matthew Dockrey, Flyer
Freshman, Undeclared
gfish@cyphertext.net

Adam Bliss, Alternate Flyer
Senior, Mathematics
abliss@hmc.edu

1. Table of Contents

I. Technical Portion

1. Experimental Abstract

We will build an autonomous free-flying robot capable of orienting itself using visual cues and navigating in a microgravity environment. The robot will determine its rotational and translational velocity and direction using images acquired by onboard cameras. Because the environments in which autonomous free-flying robots might be expected to be used (the KC-135 or a space station) are difficult to understand and model visually, we will be testing three different motion capture algorithms. Once the robot has determined its motion vector visually, it will attempt to navigate, first attempting to cancel its current motion vector, and then to execute a preplanned series of test maneuvers. Propulsion will be via high pressure compressed air reaction jets under electronic computer control.

2. Scientific Findings

Data Gathering

Our experiment was designed to compare the relative success of two vision algorithms by observing how quickly and accurately GYRE performed station-keeping based on their calculations. However, aboard the KC-135, a hardware failure prevented us from firing the thrusters and observing the merit of the algorithms directly in this fashion. Instead, team members aboard the KC-135 recorded live footage taken in microgravity by the robot's cameras to GYRE's hard drive. This video was used to perform 11 simulated parabolas on the ground in the weeks after our flight. Since the video was taken in-flight with the actual hardware, resolution, timing and speeds intended in our experiment, the simulated parabolas represent an opportunity to test the algorithms accurately.

Analysis

Each algorithm operates upon two frames taken 1/30th of a second apart. It selects features of interest from each frame, makes a list of all notable features, and compares the lists from both frames. If a feature appears in both frames, the algorithm compares its locations in each and records the distance its centroid has shifted between frames. The movements of all features in each frame are averaged into a vector representing the total estimated motion for that frame. Input from each camera is used to estimate motion in two dimensions.

We chose two criteria to grade each algorithm on: accuracy and precision. Both are important in a visual servoing application. Precision is a measure of how repeatable the algorithm's results are or how similar the results from two similar frames are. To determine precision, a human crew member reviewing the images selected fifteen series of frames that depicted smooth movement. Usually this occurred during times when the robot was drifting in microgravity without outside impetus from a crew member or collision. We then ran the frame series through each algorithm and looked for "spikes" in the resulting data. These one-time, large errors are most likely caused by an incorrect matching of features in each frame and are characterized by strong deviations in both x and y vector estimations. Each algorithm lost a point for each spike.

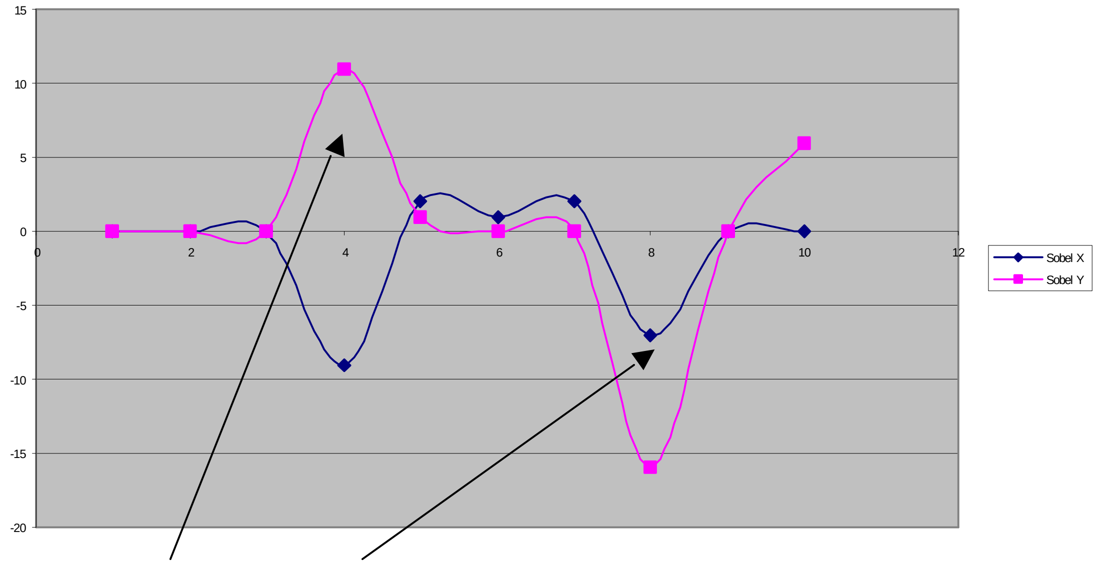
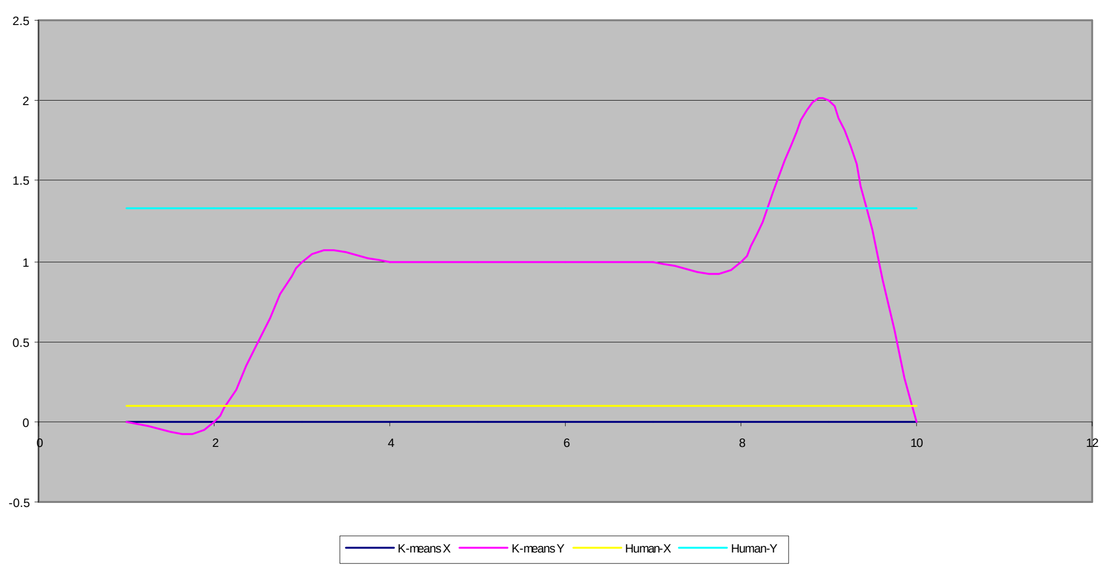


Figure 1: Precision: Two “spikes” are visible in the Sobel-derived motion estimation data shown here.

In visual servoing, a “spike” in motion estimation data would cause the robot to overreact and possibly spin itself out of control in an attempt to correct what it sees as a sudden large error. Such a mistake



may well undo several rounds of careful positioning.

We also graded each algorithm on accuracy. A team member selected fifteen series of ten frames that depicted movement along a single vector without noticeable acceleration, deceleration, or speed change. The human team member estimated motion from beginning to end of the sequence, and this estimation was averaged across frames. Both algorithms were run on the frames and the algorithm with less error (as calculated by the least squares method) won a point. Many of the accuracy test frames feature motion in only one axis, which is easiest to verify with human judgement.

Algorithm I: K-means color clustering

The color clustering algorithm tracks the centers of pixel areas of distinct color. It begins by grouping each pixel in an image into one of several color groups, based on RGB Euclidean distance from one of several seed colors. It then discards any colored areas that take up less

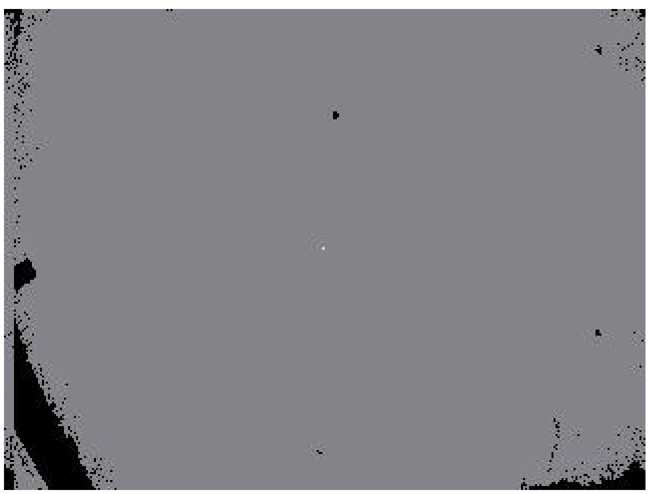
t color.

than one percent of the frame as being too small to be accurate. Finally, it records the center, area, and boundaries of each area and tries to match them up with similar areas in the next frame.

This image shows lines connecting areas the algorithm decided were similar. The upper arm and hand are moving at a different speed than the rest of the image - the crew member is straightening his arm.

Where it goes wrong

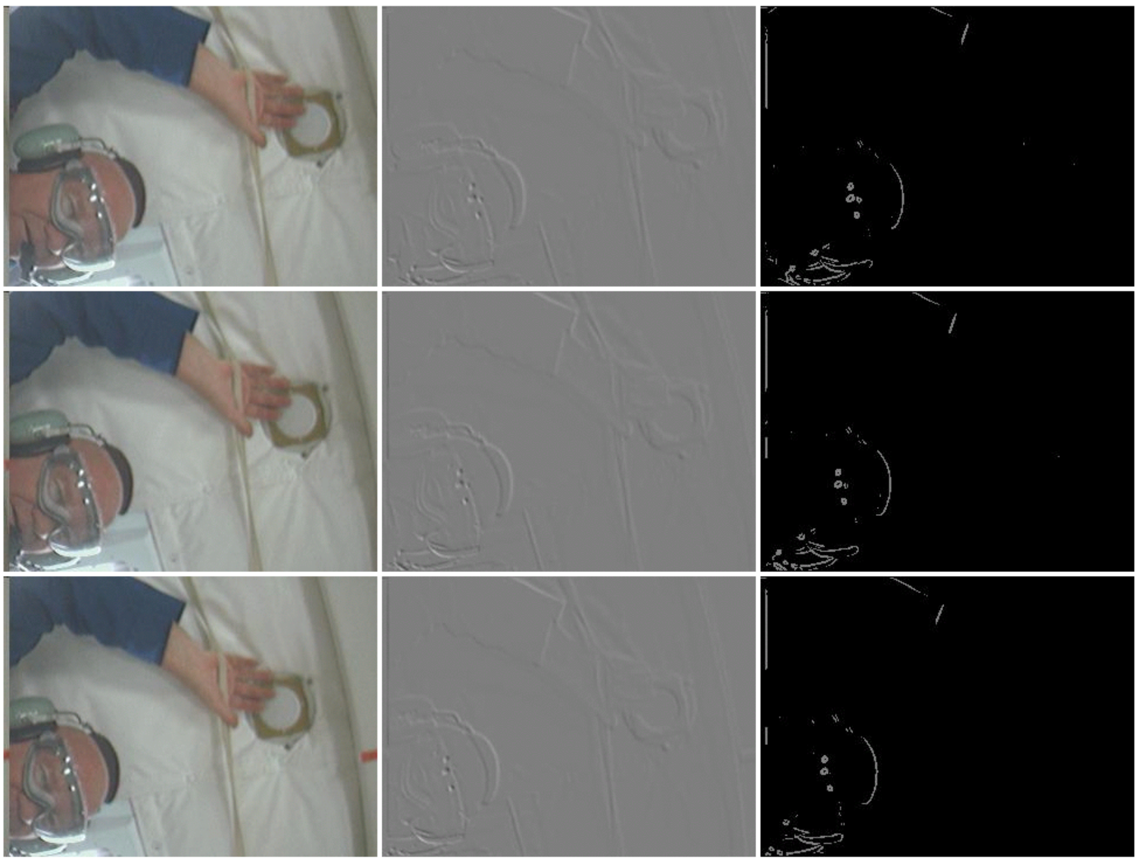
The K-means color sorter works well when an image has about ten medium-sized colored areas. Too many areas cause it to have more difficulty correctly matching images; too few and there is no data to extract a vector from.



One thing the simulated runs did not include was execution of the automatic configuration routine, designed to select a good set of seed colors for a particular environment. We feel this would increase the accuracy of the images, but the configuration routine is designed to utilize exact camera angles and those shots were not available to us.

Algorithm II: Sobel/Burns edge detection

The Sobel/Burns feature extractor uses a Sobel matrix to select pixels with a high regional gradient from a grayscale image. These pixels are grouped into line segments by angle and intersection using the Burns accumulator. The algorithm then tries to match line segments that appear in both images.

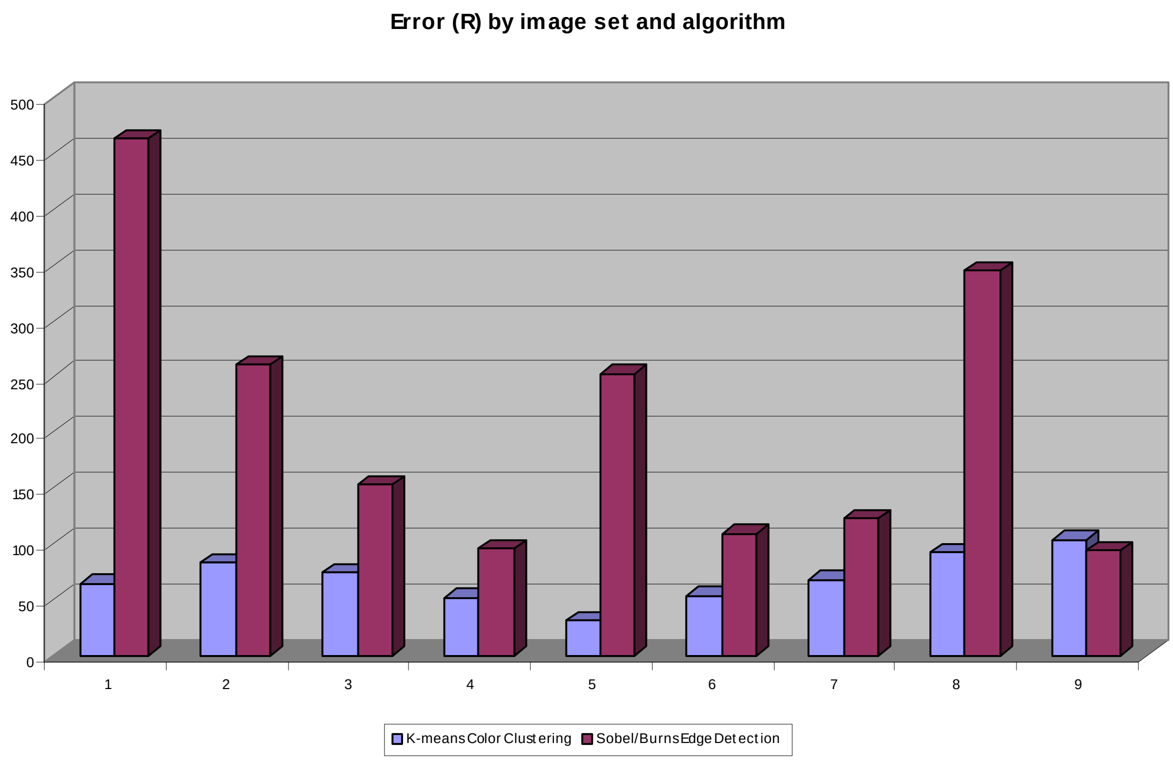


Where it goes wrong:

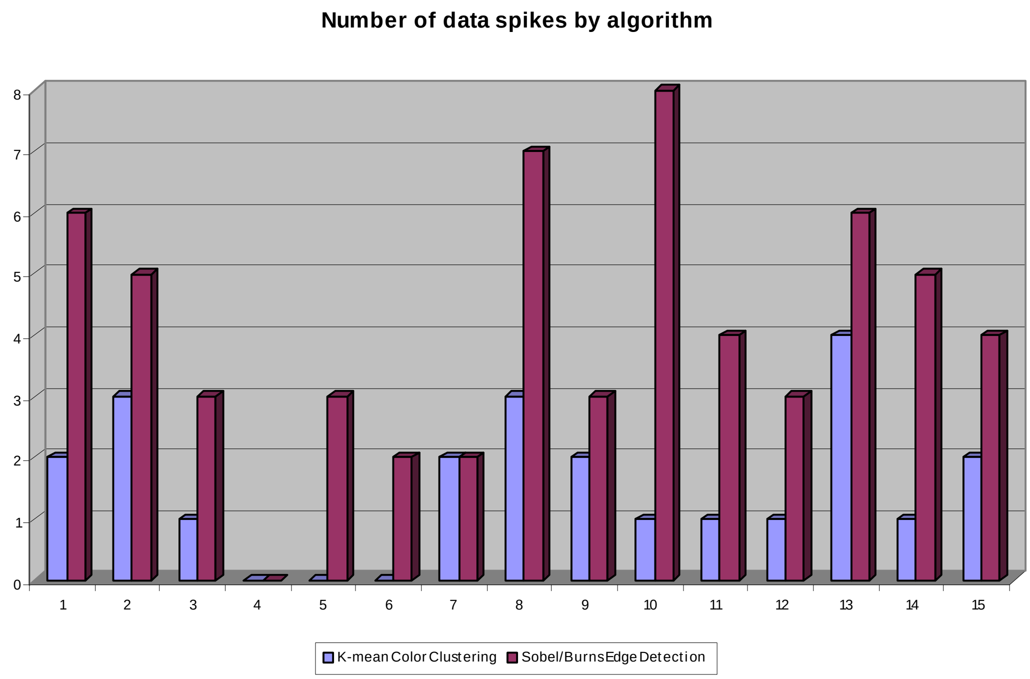
The edge detection algorithm works well when there are a few obvious stright lines to correlate. Very complex curved shapes contain no line segments, and fine textures contain many lines too small to group.

3. Analysis, discussion, and conclusion

Our results were calculated from a number of simulated short parabolas. We measured the precision of each algorithm by comparing the number of data spikes each produced on common input, and the accuracy by comparing the square root of R^2, using human-generated estimation as our baseline. We feel these are an accurate reflection of the performance of the algorithm since each algorithm received images actually generated under working conditions, and each algorithm received the same set of images.



The Sobel/Burns edge detection consistently has a higher error than the K-means color clustering.



In fact the Sobel edge detection algorithm is so noisy and imprecise it was hard to evaluate its accuracy separately, even considering the fact that only short runs of the smoothest footage were chosen. For some particularly bad frame sequences, nearly 20% of the Sobel data consisted of spikes. It is quite apparent that the K-means color clustering is a better algorithm for this application and our subsequent experiments will use it.

Where to go from here:

While the K-means algorithm is clearly better than the Sobel/Burns transformation, there is room for improvement. We plan to rewrite the code to automatically reconfigure colors every hundred frames or so to get more meaningful colored areas and help deal with all-white areas of the vomit comet by selecting light gray clusters.

Even so, The K-means algorithm is still sometimes inaccurate or noisy. The next portion of our experiment will involve desogning and testing a feedback system, such as lead/lag or proportional-integral-derivative, that can compensate for occasionally incorrect input.

II. Outreach Activities

1. Outreach Objective

GYRE's objective in outreach is to show children and adults how robotics can be used to enhance our lives, both in space and on Earth, as well as to promote knowledge of basic scientific principles and how they can be used in robotics design and space flight.

2. Website

An extensive website for the project is located at `http://depts.washington.edu/gyre/`. It currently contains photo galleries of robot construction, outreach activities (visits to schools and the Pacific Science Center), information about team members and the proposal, and technical information. Links are provided to other relevant sites, including NASA's. This site contains over 1,000 files totalling over 200 megabytes. No access statistics are available.

3. Powerpoint and Poster Presentations

We demonstrated the robot at the Pacific Science Center, a large science museum in Seattle, as part of a special event commemorating the International Space Station and release of the IMAX movie on May 4 and 5, 2002. We erected the robot's testing framework at the Science Center, demonstrated and explained propulsion and camera systems to visitors, exhibited posters and and discussed special requirements for robotics in microgravity. Science Center staff have also expressed interest in this project as a very unusual addition to the annual robotics week in May. They suggested a half-hour lecture/demonstration including a talk given by a team member, a demonstration involving the actual robot, and a video. Details will be arranged after the flight.

We visited Lawton Elementary School in Seattle in Spring 2002 and demonstrated the robot to fourth and fifth graders as part of a unit on programming and robotics. To each of three participating classes, we gave a slide show and demonstration on principles of robotics and then discussion and hands-on demonstration of our robot. School staff expressed interest in having us return for a followup or to repeat the demonstration for new students, but exact details have not yet been arranged.

We traveled to Western Washington University in Bellingham to demonstrate the robot for a Robotics and Art unit for elementary school children who have qualified for the Highly Capable program in the Bellingham school district. We discussed the uses of robotics in modern society, especially scientific / high technology environment uses.

We presented at the University of Washington Undergraduate Research Symposium, a public event which gathers interest mostly from college freshmen, their families, and University departments. These are the people most likely to be working on RGSFOP projects in the near future and thus have an immediate and compelling interest.

We have presented our work to an undergraduate robotics course (EE400) at the University of Washington.

4. Photographs Of The Team Addressing Community Groups

n extensive photograph collection is available on our website, `http://depts.washington.edu/gyre`.

5. Copies Of Press Coverage Of Your Team

An article on our project was written for and published in the University of Washington’s EE department newsletter, the "Electrical Engineering Kaleidoscope", in January of 2003.

Washington Space Grant published an article about our project in their newsletter “Expanding Frontiers”, available at `http://waspacegrant.org/effallwin03.html`.