Vision Based Control of a Mobile Robot

Anthony Erlinger

Santa Barbara City College
Electrical Engineering

Mentor: James Riehl
Faculty Advisor: Professor João P. Hespanha

Center for Control Engineering and Computation (CCEC)

The Hardware:

ActivMedia's Pioneer 3-DX Mobile Robot

- Linux Platform
- 8 Sonar sensors at 15 degree offset
- Two independent wheels.
- Hitachi H8S Processor

Sony PTZ (Pan-Tilt-Zoom) Camera

- 95 degree Pan (Left-Right)
- 25 degree Tilt (Up-Down)
- 12x Zoom

To be used as a research platform for more sophisticated applications such as unmanned vehicles or other autonomous systems.

A Few Applications of this Technology:

Unmanned Vehicles

- Aerial Vehicles
- Extraterrestrial Vehicles
- Automobiles

Satellites

- Use image processing to search for and track various targets.

Autonomous Security Robots

Objective:

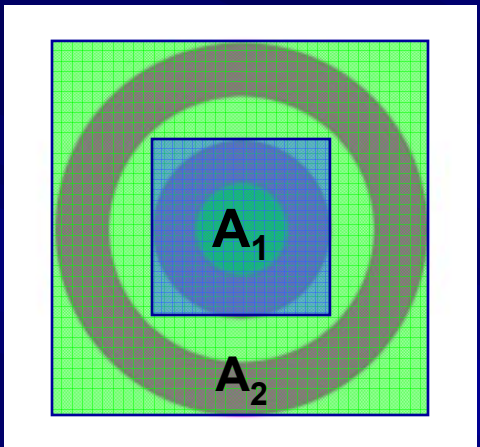
Develop a C++ interface for robust target detection and navigation of an autonomous robot to be used in research on advanced search algorithms and various control experiments.

Challenges:

1. Recognizing multiple targets with different colors and shapes.
2. Finding distances to detected targets.

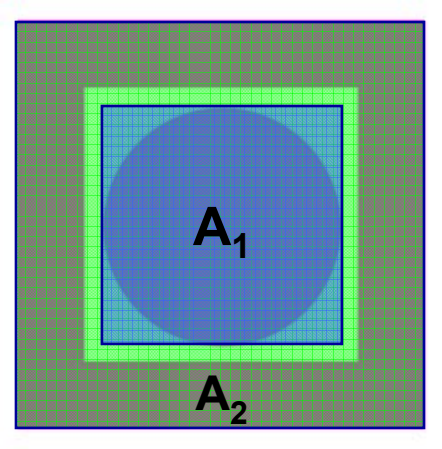
Challenge #1: Target Detection

In order to navigate we first must be able to detect various types of targets. Below are two different types of targets which our vision detection algorithm can allow the robot to distinguish between.



Type I Target

Area-Ratio: $A_1/A_2 \approx .21$



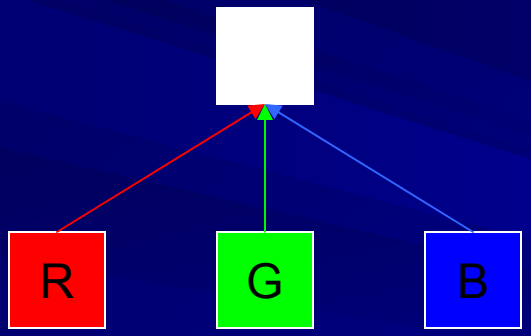
Type II Target

Area-Ratio: $A_1/A_2 \approx .30$

We distinguish between these two targets by computing the ratios of the areas of the region of each target. This allows us to distinguish between multiple sizes and colors of different targets.

What is an Image?

A digital image consists of several thousand pixels. A pixel (Short for picture element) is the smallest part of an image. Pixels in a color image are composed of three color components: Red, Green, and Blue.

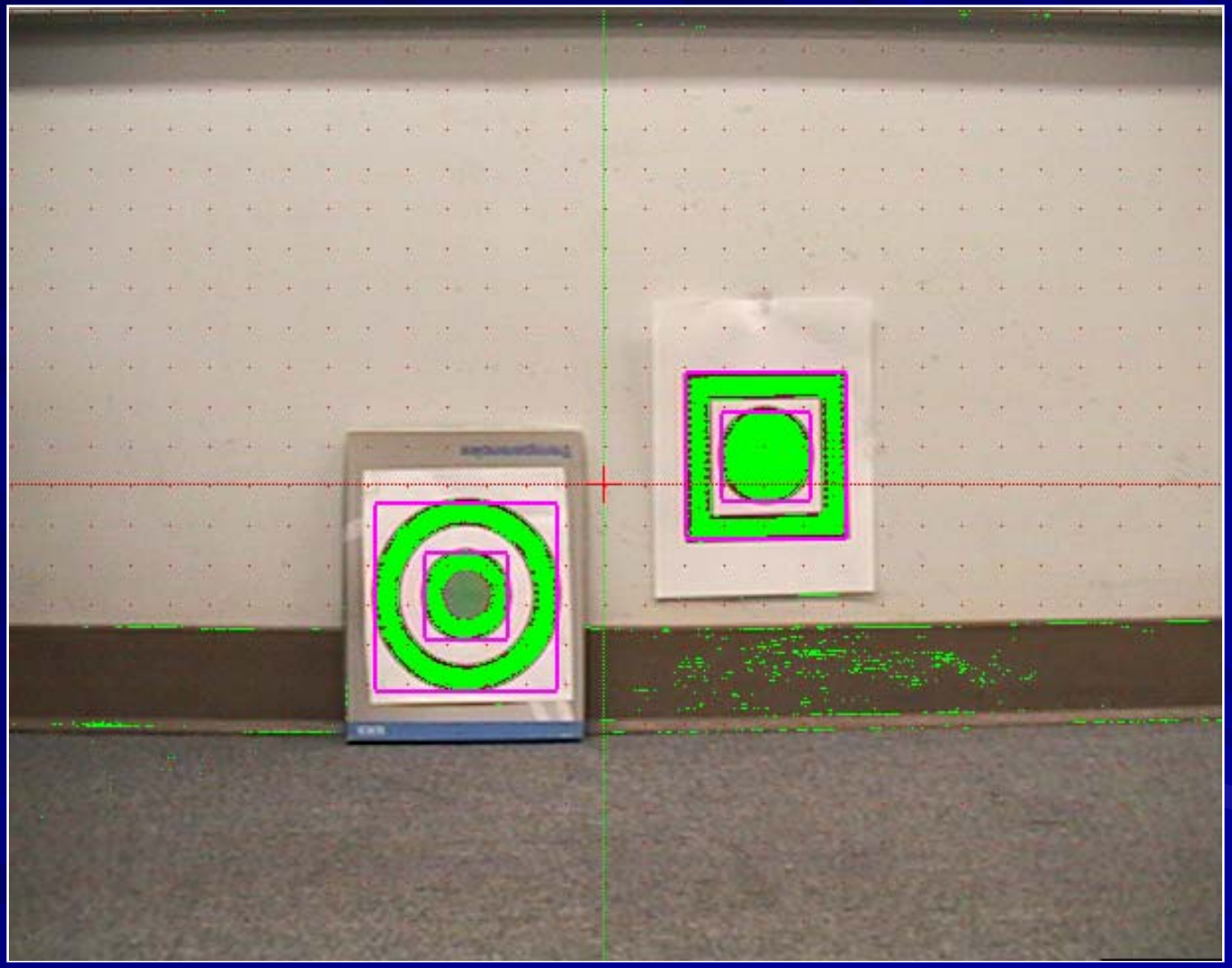


With 24 bit color we can store 8 bits for Red, Green, and Blue respectively. This allows for millions of colors.

2^{24} = 16,777,216 values of color!

In this algorithm we will be searching for pixels in this RGB range (The color of our targets):

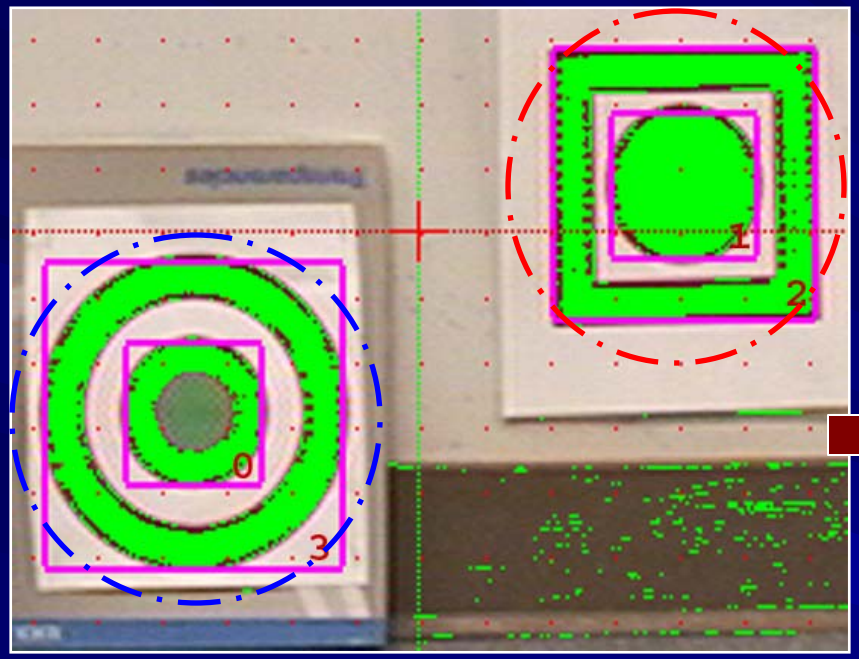
How the Robot Sees the Target:



Color Threshold:

Pixels within color threshold

A Closer Look:

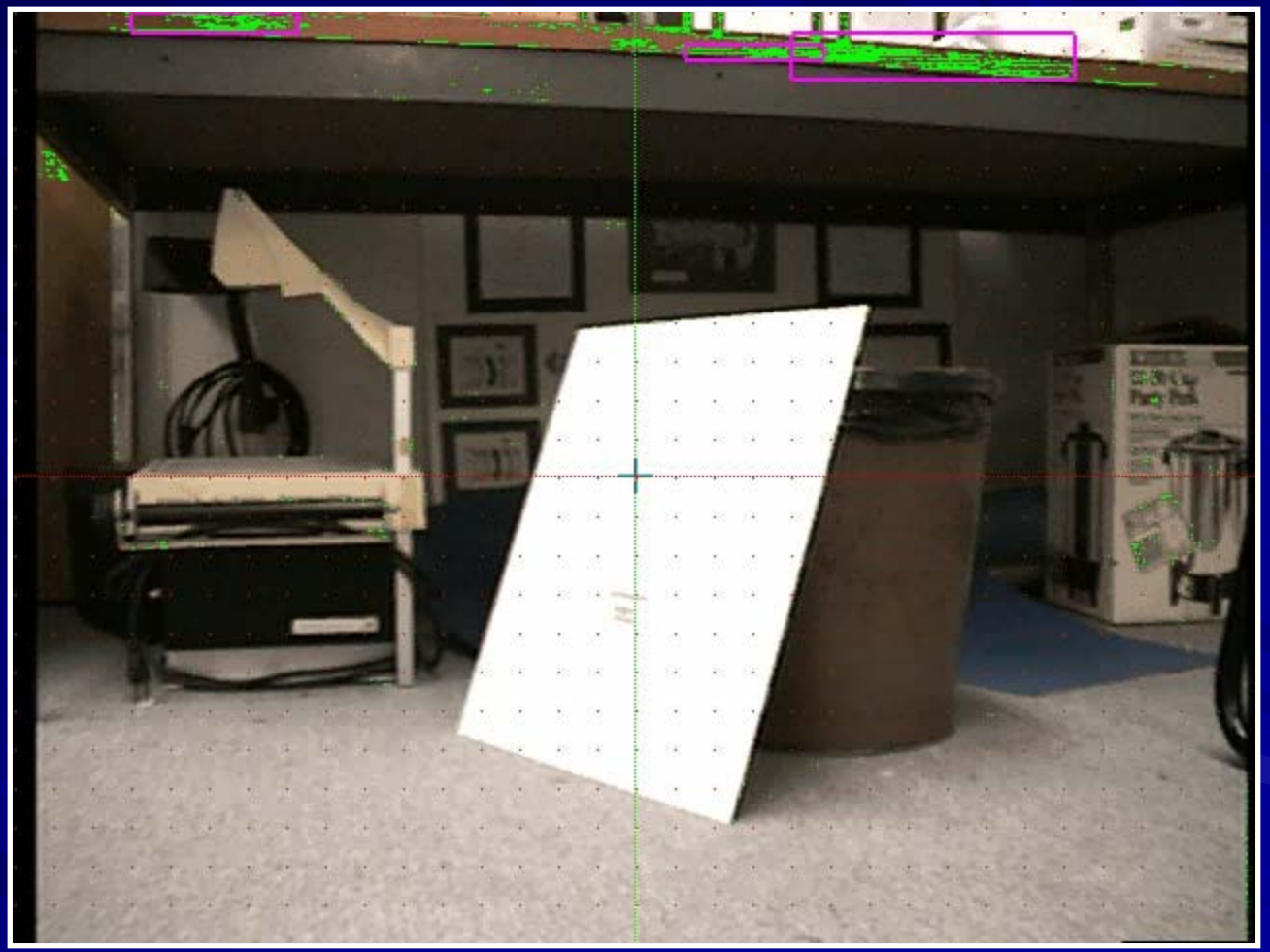


Console Output:

```
Number of pixels in Target 0:  969     (x, y):  250,  296
  Left-Right edge of Target 0: 229      272.
  Top-Bottom edge of Target 0: 275      318.
        Area: 1849
Number of pixels in Target 1:  1609    (x, y):  402,  225
  Left-Right edge of Target 1: 379      425.
  Top-Bottom edge of Target 1: 204      247.
        Area: 1978
Number of pixels in Target 2:  2717    (x, y):  403,  225
  Left-Right edge of Target 2: 362      443.
  Top-Bottom edge of Target 2: 184      268.
        Area: 6804
Number of pixels in Target 3:  2871    (x, y):  250,  296
  Left-Right edge of Target 3: 204      297.
  Top-Bottom edge of Target 3: 249      344.
        Area: 8835
Number of filtered targets:  4
```

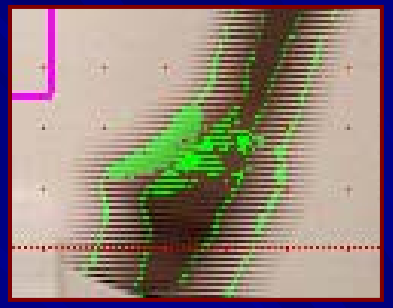
```
Largest target is:  3, with size of 3044.
Area:  1849, 8835. Area-ratio 0.209281
Target type I detected!
        (X,Y) Position of target:    (250, 296)

Area:  2024, 6804. Area-ratio 0.297472
Target type II detected!
        (X,Y) Position of target:    (402, 226)
```

Video Demonstration:

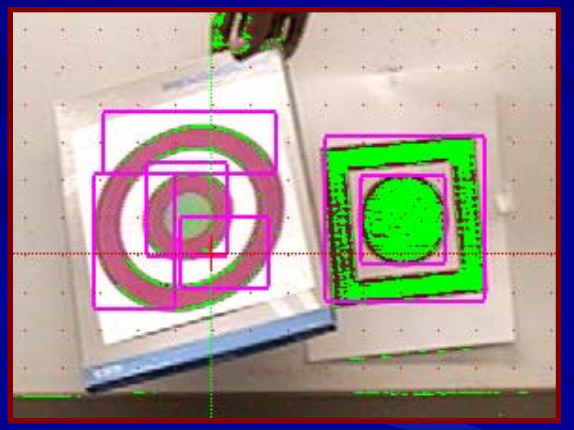


Common Difficulties with Image Processing Algorithms

- Camera Blur
  - There is some time required for the camera to capture an image which depends on the camera's shutter speed. Large changes in movement will result in a blurred image.

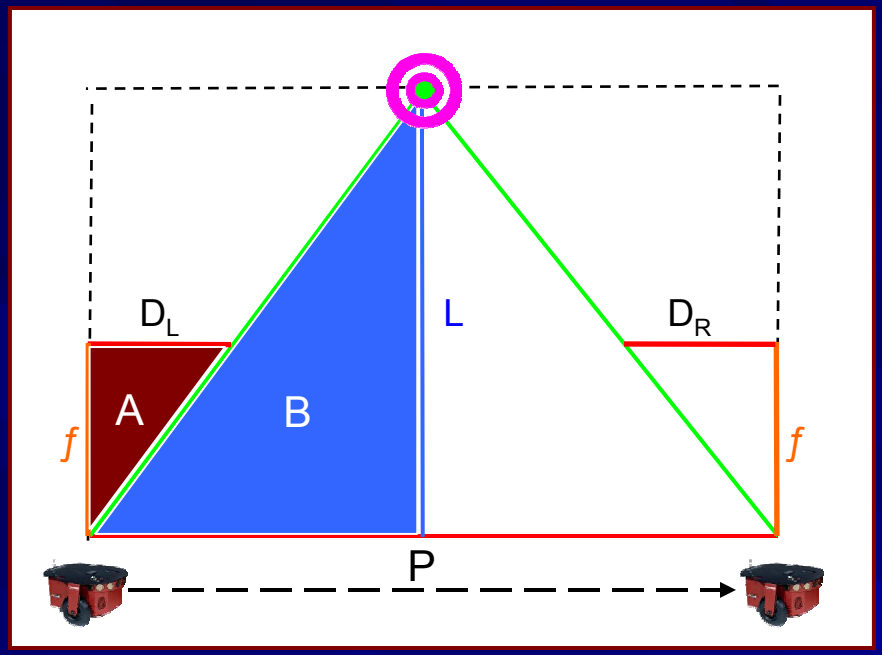


- Color Distortion
  - Colors appear differently to the camera at different angles, or at different values of light. This can cause errors in target detection or unexpected results.

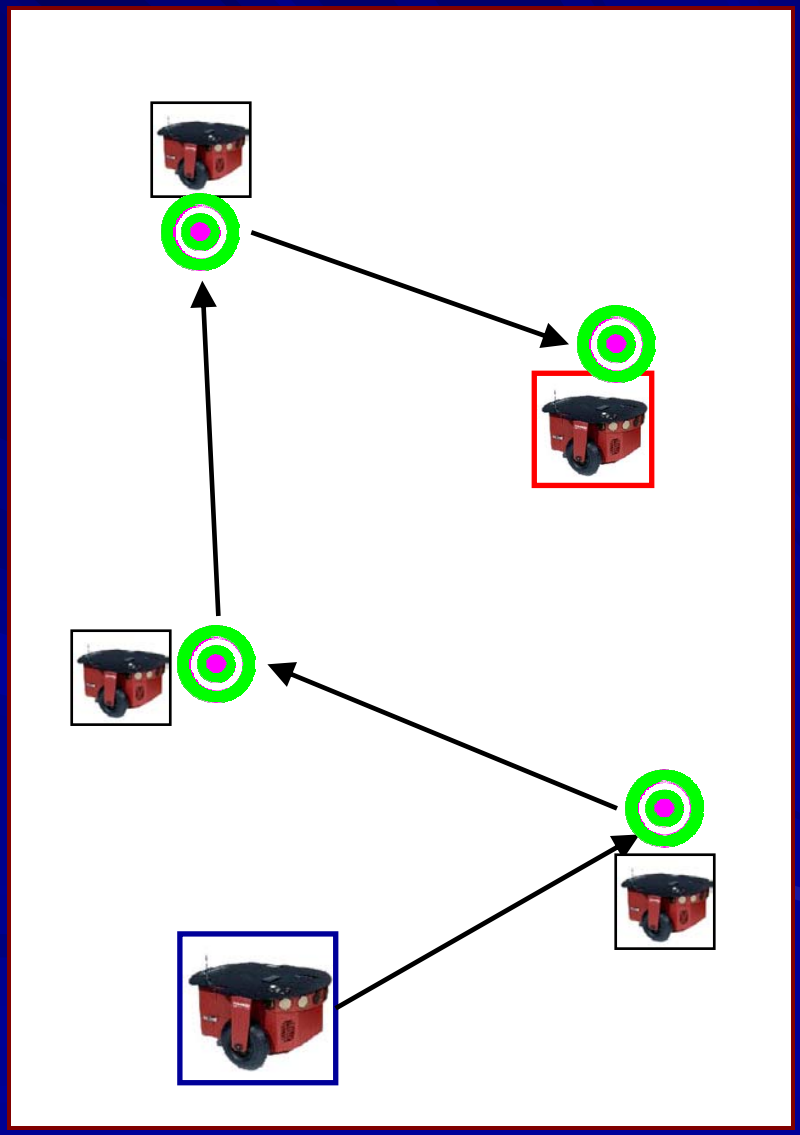


- Time Lag
  - There are 307,200 pixels in a 640x480 image. Each of these pixels must be processed *many* times each frame.
    - 640x480 = 307,200 Pixels → Several million calculations… All done in 250 milliseconds.

Challenge #2: Computing Distance from a Target



Using geometrical techniques we can approximate a robot's distance from a specific target given multiple positions of the robot and the percieved position of the target in the image captured by the camera.



Acknowledgements:

- **Mentor** James Riehl
- **Faculty Advisor** Professor João Hespanha
- Ricardo Sanfelice
- INSET Program
- National Science Foundation for funding and sponsorship of INSET
- Family and others who have always been for me along the way