Project2: Stereocritter
CS3630 Spring 2007

Due Begining of Class Feb 20

11th February 2007

1 Introduction

Widely used as a sensor for robot navigation, *stereo vision* uses two cameras to recover the depth of the scene. In this project, you will implement a "stereo-critter" in Player/Stage that uses stereo for perception, and behavior-based control for navigation, in particular, motor schema.

2 Part One (Due Feb 6)

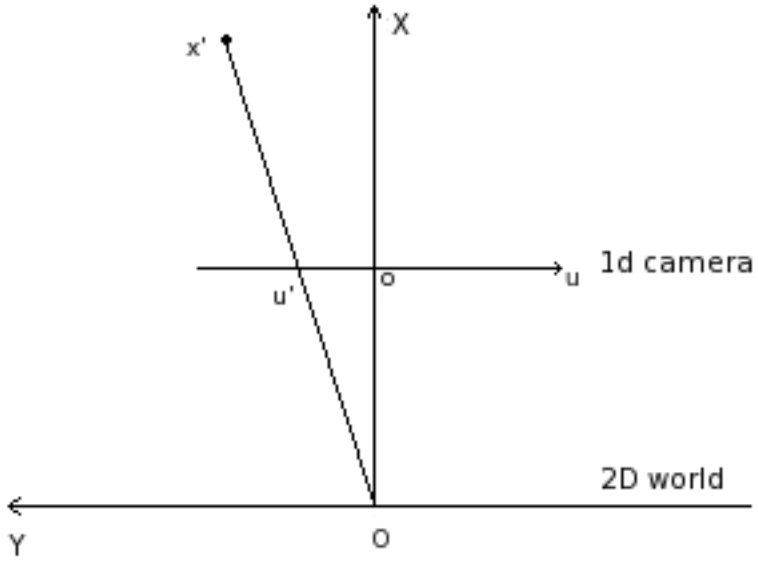
Install Player/Stage on your machine and try to get familiar with this environment. The installation packages have already been put on our course webpage (http://borg.cc.gatech.edu/ipr/html/resources.html). Submit a screen shot of your installed Player/Stage before the class on Feb 6, 2007 (Tuesday), so that I can make sure all of you have been ready for the part two of this project which will be given out soon. If you have trouble to install Player/Stage, please send email to the TA (houdan@cc.gatech.edu) before Feb 1 (Thursday). We can hold a tutorial session on Feb 2 (Friday) in some CoC lab depending on how many of you need help.

3 Part Two (Due Feb 20)

Once Player/Stage is running satisfactorily, part two consists of implementing a "stereo-critter" by completing the following 5 assignments:

3.1 1D Stereo

We simplify the stereo problem by assuming *known correspondences and 1D camera*s, which means we provide the image coordinates of the same obstacle on two 1D cameras. Use the image coordinates provided to compute the depth of all detected obstacles. Also, you could use the depth to recover the range and bearing of the obstacle relative to your stereocritter. We have provided the ground truth of range and bearing which you can use to validate your result. Note that when you compute bearing and range, you should be aware of the tranform of different coordinate system, which is illustrated in the figure below:



3.2 Motor Schema

Implement motor schema to solve the following three navigation problems:

1. No obstacles, very easy. You just need to navigate your stereoscritter directly to the goal. Submit one figure with the trail of your stereoscritter. Note that there is an option of 'show trails' in the menu of stage window.
2. Several obstacles scattered in the world, a little harder. You should navigate your stereocritter to avoid these obstacles in order to reach the goal. Submit one figure with the trail of your stereoscritter.
3. Obstacles arranged in a sink shape, you might get trapped in this case. Submit one figure with the trail of your stereoscritter.

3.3 Avoid-Past

Implement the avoid-past schema to solve the problem (3) above. Use motor schema with avoid-past to rescue your stereocritter out of the sink. Submit one figure with the trail of your stereoscritter. Here are some useful resources for you to implement motor schema and avoid-past schema:

- Goodrich's Potential Fields tutorial: http://borg.cc.gatech.edu/ipr/files/goodrich_potential_fields.pdf
- Tucker's paper on Avoid-Past schema: http://borg.cc.gatech.edu/ipr/files/Balch93tra.pdf

4 Guidelines

Get the code distribution http://borg.cc.gatech.edu/ipr/files/ipr-proj2.zip. Unzip it with the command 'unzip ipr-proj2.zip'. You should now have a directory called ipr-proj2. Under that you will find a README file

which describes the detail about how to run the code. To test the program, you need to type 'make all' and then run './run_test1'. This will give you the first navigation enviroment (no obstacle) and a robot. The other two navigation enviroments are contained separately in run_test2 and run_test3. You should edit ipr-proj2.cc in the same directory to your satisfaction, then recompile with 'make all'. Please do follow the README to write and run your code.

You should email your well commented code to Dan (houdan@cc.gatech.edu). You can write your code on any platform you like, but it must compile and run on helsinki.cc.gatech.edu (a standard CoC linux box).

Additionally, at the beginning of class, you should submit a one to two page report that describes the algorithm you used. The description should be of sufficient detail that someone could reproduce your work. You should characterize the strengths and weaknesses of your approach.

Your project will be evaluated on these criteria:

- Does your algorithm solve all navigation problem it should?
- Writeup: Do you describe your algorithm sufficiently well that someone else could implement it? Do you effectively analyze your algorithm, explaining its strengths and weaknesses?
- How elegant is your solution? Extra credit will be awarded for elegant solutions.