

415.773 Assignment 3

Due Date: Fri, 25th October 2002

Assignment 3 will basically involve taking all that you have done, polishing everything and setting up the final demos.

Demos

You will have 2 different demos to perform:

1. The first will relate to the displacement of the car from a starting position (bottom left of the field) to a goal position (top right of the field). You will have to pass the bridge and go through the tunnel while avoiding a few obstacles (same as in assignment 2). All the tasks you have been involved in will contribute i.e camera calibration, patch detection, motion planning and car control. You will be given 3 trials (one after another) to complete this task. I will move the tunnel, the obstacles and the bridge, therefore you should rely on patches or/and lines detection to find the position of the tunnel, bridge and obstacles. You should also be able to compute a feasible path (related to the car limitations regarding turning radius, control, etc...) and display it. At any given time during the car motion we **MUST** be able to see the relative position of the car in respect to the planned path. Basically you need to study:
 1. The control of the car through any given straight line and how to redirect the car whenever it goes far enough from the line.
 2. How to make the car turn from a given point to another one. You should study the geometry of adjacent circles of the same radius, equal to the turning radius of the car. This should give you an idea of the potentially feasible turns the car is able to perform.
2. The second demo will involve the car control through hand posture recognition. The bridge and the tunnel will be removed from the field. A few patches (as much as different signs your alphabet includes but a least five) will be randomly placed on the field. The car will once again start from the same position as of demo 1 and will have to reach any stipulated patches. The patch to go will be identified by a sign. First you will make a sign, the sign will correspond to a number. After retrieving the patch co-ordinates, the path planning and motion planning from the given position of the car to the next patch should be computed and followed by the car. Your interface **MUST** show the path computed. It should go relatively quickly (only a few seconds between signing and starting the car motion). There is three levels of planning in this demo:
 1. Second class: You can plan the path only from the start position to any given patch (the car needs to be relocated to the start position manually when it completes its motion to the patch). This simplify the path planning and motion planning...
 2. First class: You can plan the path from any given patch to any other one. It is more complicated as patches will have random positions on the field but I remind you that there is no more obstacles between the 2 patches...
 3. God (of Motion planning at least): you keep the bridge and the tunnel and go from one patch to the next one (depending on signs) while planning the shortest path and use the bridge and tunnel when necessary.

Hints:

A good way to set up your hand gesture recognition system is to go for interaction. This will allow you to decide when a sign should be acquired, for how long. You should indicate the results of recognition. The next sign should be acquired only on request and only when the car is not moving. Here is how I would do it (items in quote signify interface instructions to the user):

Interface "Show a sign and hold it for the next 5 seconds"

The user is showing only one sign for this time. As soon as the sign is acquired the interface should ask the user to stop signing.

Interface "sign acquired, stop signing"

After the sign is recognized, the interface should provide the results

Interface "the sign is 1" (could be 2 or 3, it's just an example)

If the sign corresponds to a patch number, the path planning is performed, the car moves to (and reaches) the patch indicated by the sign.

After the motion is completed, the interface should ask the user to perform another sign.

Interface "Do a sign and keep it for the next 5 seconds"

Stand alone hand posture recognition system

You MUST set-up a version of your interface which recognizes signs one after another without interaction with car control.

Example:

Interface: "Show a sign and hold it for the next 5 seconds"

The User is doing ONE sign for the requested time.

Interface "sign acquired, stop signing"

Interface "the sign is 1" (could be 2 or 3, it's just an example)

Interface: "Show a sign and hold it for the next 5 seconds"

The interface should also state when a sign is not recognized. This will help me test your hand gesture ability.

Note: You need not use an interface such as explained above if your system is fast enough to deal with video stream (several frames per second). In this case, the user should perform a series of signs and the program display the outcomes as fast as possible. You may explore the performance of your system and provide corresponding results in your report. I will extensively try your system :-)

Code submission

Each group will have to submit its code (just before the demo) for each part of the project to me. Each part should include a README file and a HOWTO file. The README file will describe the different components of your code (camera calibration, line detection, etc.). The HOWTO file gives the commands I need to use to run each section of your code and what I should expect in return (an image, a text file). You can create an HOWTO file by portion (Calibration, line detection, path planning, motion planning). A DEMO file should list the instructions (a list of what you type each time you launch something) needed to perform your demo. You have to submit the stand alone hand posture recognition system with a HOWTO and README file. Last but not least, your code should be well commented. After the demos I will check and run your code with the instructions provided and will take that into account for the overall practical part (Assignment 1,2,3, assessment, demo) final mark. I shall be able to run your code without any understanding of what it is doing as all the instructions should be listed in the HOWTO file and (concise) descriptions of the code in the README file.

Sharing

For the groups which are ahead, you MUST provide the code to any other groups which may request it. However you need to keep track of it (which group is using your code) and I will take that into account for assignment 3 marking. Groups requesting codes from others should be clear about it. Don't ask for code if you don't intend to use it...

Demo

Demos will take place either on the 25th of October or the following week (no lecture) for example between 12.30 pm and 2.30 pm on the 30th of October. We want all the keys and swipe cards to be returned by the 1st of November. We will not mark your group exams if you are the key holder and the keys haven't been returned. Even if you intend to do a project with us afterwards I would like you to return the keys and cards first and apply again if necessary. this is to protect the lab as we had a burglary last year and would like to avoid it this summer.

Submission

Your report MUST include both individual sections (one per student) and a group section. The group report may deal with introduction, description of your overall achievements in regard with the proposed work and conclusion. Each group member should write an individual report regarding his/her contribution to the project. Your report should be a scientific contribution rather than a daily diary account of your 773 journey. Front page should clearly indicate group numbers, group members and who did what. If you have improved previous parts, mention briefly the related theory and show improved results.

The new parts developed with respect to the work previously done in the previous exercise and assignment

Description of algorithms used for the different parts.

Description of your programs.

Description of your experiments.

Conclusions and/or comments.