Open[ELE00142M] Machine Vision and Human Machine Interaction

UNIVERSITY of York

Lab 3: Visual Servoing with MATLAB

Lecturer: Jihong Zhu <jihong.zhu@york.ac.uk>

Demonstrator: John Bateman <john.bateman@york.ac.uk>

Technician: Alejandro Pascual San Roman <alejandro.pascualsanroman@york.ac.uk>

Aims and Objectives

In this lab session you will be doing further reading on a tutorial on visual servoing and implementing a image-based visual servoing algorithm with MATLAB. You can should use Windows instead of Linux.

Why switch to MATLAB Instead? I am a bit concerned and only a portion of the students have robotics knowledge.

It is expected that you will complete all of these activities in two weeks of labs (Week 7 - Week 8). The P/T/410 and P/T/411 laboratories are open for your use outside of scheduled lab times in case you need extra time to finish. You can also work at home by using MATLAB on your own computer.

Learning outcomes

- Implementing successfully a visual servoing algorithm with MATLAB
- Showcase the the robot movement with MATLAB

Software required

- MATLAB

Tasks

1. Required package for IBVS

The lab requires installing the Robotics Toolbox:

https://petercorke.com/toolboxes/robotics-toolbox/

2. Reading

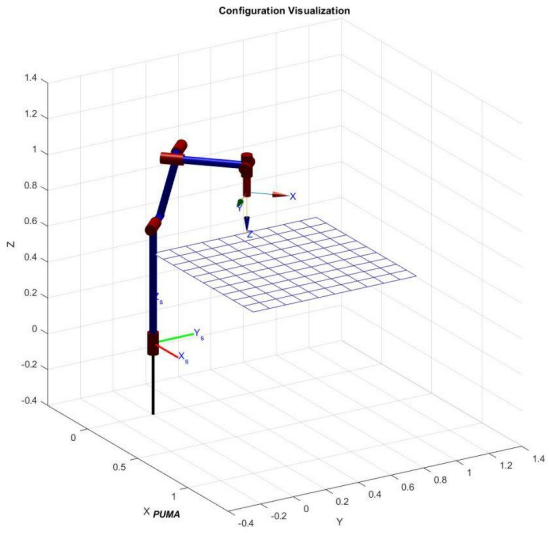
Before doing the lab assignment it is recommended to read through:

Visual servo control. I. Basic approaches:
https://ieeexplore.ieee.org/document/4015997

which give a more detailed description on visual servoing.

3. Your Tasks

Open main.m file and run the first and second section, you should see the following:



A 6 DoFs robot with a task plane in grid.

The intrinsic parameter of the camera and the task points is given (running

Camera calibration & Set Target point & 4 interest points section & Desired Parameter for robot control)

The start the VS loop, you need to fill in where I have put

```
%------ Your code here ---------------------------------------
```

```
%-----------------------------------------------------------
```

There are instructions and also hint on how to proceed

In Section Spatial Velocity Calculation, you will also need to fill `Calculate_Lx.m`

If everything is implemented correctly, you should be able to visualize a robot moving, plus a view of the image features converge to the target.