ECSE 4480/6470, MANE 4560/6120, CSCI 4480 November 1, 2021

Robotics I

Mini-Project 5: Vision Guided Motion

Assigned: November 1, 2021
Due: November 16, 2021

Check back here regularly for the latest update
Submit your project report to Gradescope and your code through your private GitHub repository. Discussion with your peers is encouraged (particularly on WebEx Team and Piazza), but you must hand in your own work and be able to explain what you have done in the project. **Verbatim copying (whether you copy from someone or let someone copy your work) of derivation, writing, software code, etc., is considered cheating and will result in zero for the project grade and notification to the Class Dean and Dean of Students. Multiple instances of cheating will result in failing of the course.**

Things to Do

- Post weekly update of your project progress on your team's WebEx space.

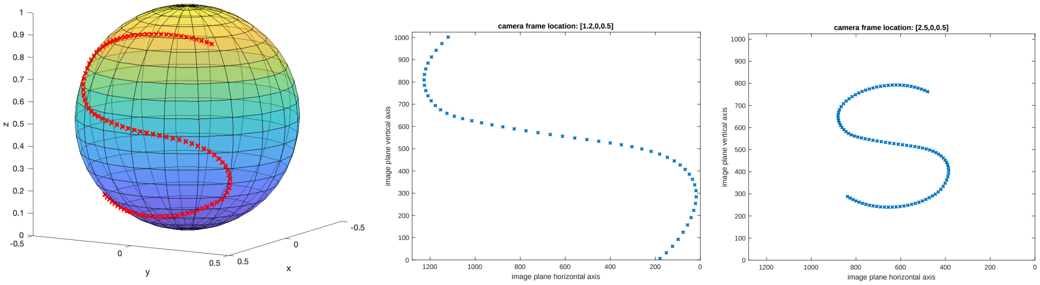
Task Description

1. **Camera Model and Projective Geometry** (20 points)

   Use the provided MATLAB function `cam_image.m` for a sample pinhole camera with the following intrinsic parameters:

```
W = 1280; % width in pixels (1280)
H = 1024; % height in pixels (1024)
rhow = 1e-5; % width per pixel (10um)
rhoh = 1e-5; % height per pixel (10um)
f = .015; % focal length (0.015m)
u0=W/2; %center of image plane in pixel coordinate
v0=H/2; %center of image plane in pixel coordinate
```

   Suppose the camera is pointing directly at the S-shape curve and the sphere. This means that if the reference frame (i.e., the frame to describe the S-shape) is $[\vec{x}_0, \vec{y}_0, \vec{z}_0]$, then the camera z-axis is $-\vec{x}_0$, camera x-axis is $-\vec{y}_0$, and camera y-axis is $\vec{z}_0$. Position the camera frame origin in the reference frame at $\begin{bmatrix} x_c \\ 0 \\ 0.5 \end{bmatrix}$ for $x_c = 1.2$ and 2.5. Your images should appear like below:



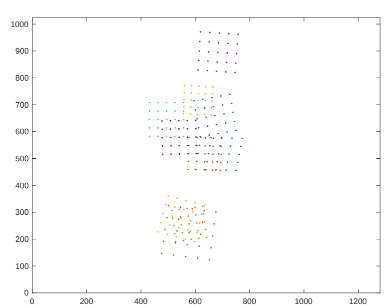
2. **Intrinsic Camera Parameter Calibration** (40 points)

   Use the following provided MATLAB code:
   `camcalib.m`: camera calibration script (it calls the following routines)
   `cam_def.m`: pin hold camera object definition
   `target_def.m`: a pattern of planar dots as the calibration target
   `homographymat.m`: solution of the Homography matrix
   `cam_image.m`: generation of of a camera image from a pattern of 3D points.

   (a) (10 points) Explain the derivation of the camera calibration process.

   (b) (10 points) Comment on the effect of the number of target points and number of views on the estimated camera intrinsic parameters (e.g., when there is rounding error and noise in the camera sensor).

   (c) (20 points) Use `cam_image.m` to generate the camera images of the target pattern and add a random image noise with zero mean Gaussian distribution and 0.1 pixel standard deviation. Find a combination of target pattern points and number of distinct view so that the error of estimated $(u_0, v_0, \lambda_x, \lambda_y)$ is below 1% and the worst case reprojection error less than 1 pixels.



3. **Camera/Robot Calibration** (20 points)

   Suppose the camera is mounted on the end effector of an IRB 1200 robot as used in Mini-Projects 3 and 4, but the location (position and orientation) of the camera with respect to the end effector is not exactly known. Furthermore, the location of the S-shaped curve relative to the robot frame is also unknown. Suppose a set of images and the corresponding robot joint angles are obtained. Propose an algorithm that will estimate the camera location with respect to the robot end effector and the robot location with respect to the S-shaped curve. Assume the S-shaped curve is known in the reference frame. Test your algorithm by putting the robot base at $p_{0B} = \begin{bmatrix} 2.5 \\ 0 \\ 0 \end{bmatrix}$ m and camera mounted at $p_{TC} = \begin{bmatrix} 0 \\ 0 \\ 0.05 \end{bmatrix}$ m on the end effector (in the end effector frame) with the camera optical axis pointing in the end effector x-direction.

4. **Visual Servoing** (Bonus Part: 40 points)

   (a) (20 points) Assume the camera and robot are both calibrated as in previous parts. For the robot and camera as located in Part 3, use position-based visual servoing (PBVS) to drive the robot end effector to $\begin{bmatrix} 1.8 \\ 0 \\ 0.5 \end{bmatrix}$ m and directly facing the sphere (so $\vec{x}_T = -\vec{x}_0$, $\vec{y}_T = -\vec{y}_0$, $\vec{z}_T = \vec{z}_0$). Start with an initial angle $q_0 = \begin{bmatrix} 0.1 \\ 1.6 \\ 3.5 \\ 3 \\ 2 \\ 0 \end{bmatrix}$ rad Acquire image of the S-shape curve, determine the position and orientation error, use kinematic control (i.e., a gradient step) to move closer to the target, acquire a new image, and repeat.

   (b) (20 points:) Describe how you can do this directly in the image plane (knowing what the S-shape curve should look like in the image plane at the target configuration). This is known as the image-based visual servoing (IBVS).

Deliverable

1. Your project report should be structured as follows:

   (a) Cover page with your name, course number, date, project title, and a statement on academic integrity (stating that you did this project by yourself):
   I, [name], certify that the following work is my own and completed in accordance with the academic integrity policy as described in the Robotics I course syllabus.

   (b) Summary: what did you try to do and accomplished.

   (c) Technical Content: containing the following for each section:

      i. Description of the problem
      ii. Derivation of the solution
      iii. Results based on your simulation

      The key is to **show that you understand what you are doing**, not just tweaking the code.

   (d) Conclusion: what you learned and what can be improved.

2. Put your code in your private GitHub repository (shared with the Instructor and TA).

3. If you have any video of your results (e.g., movies of the robot motion), provide a link (e.g., YouTube), or put them in your GitHub repository (in a separate folder under the miniproject1 folder).