MIT CSAIL

6.819/6.869 Advances in Computer Vision

Spring 2021

Problem Set 1

Posted: Tuesday, February 8, 2022   **Due:** Tuesday 23:59, Feb 15, 2022

6.869 and 6.819 students are expected to finish all problems unless otherwise indicated.

We provide a python notebook with the code to be completed. You can run it locally or in Colab. To use Colab, upload it to Google Drive and select 'open in colab', which will allow you to complete the problems without setting up your own environment. Once you have finished, copy the code sections that you have completed as screenshots to the report.

Please submit a .zip file named ⟨your_kerberos⟩.zip containing 1) a report named `report.pdf` including your answers to all required questions with images and/or plots showing your results, and 2) the python notebook provided, with the cells run and the relevant source code. **Only the report will be considered for grading**, but the code may be used in case of code of honor violations.

Late Submission Policy: If your pset is submitted within 7 days (rounding up) of the original deadline, you will receive partial credit. Such submissions will be penalized by a multiplicative coefficient that linearly decreases from 1 to 0.5

Problem 1 *Perspective and orthographic projections* (1 point)

The goal of this first exercise is to take images with different settings of a camera to create pictures with perspective projection and with orthographic projection. Both pictures should cover the same piece of the scene. You can take pictures of real places or objects (e.g. your furniture).

To create pictures with orthographic projection you can do two things: 1) use the zoom of the camera, 2) crop the central part of a picture. You will have to play with the distance between the camera and the scene, and with the zoom or cropping so that both images look as similar as possible, only differing in the type of projection (similar to figure 2.2 in the lecture 1 notes - section 1.2 of the book i.e. `simple_vision_system.pdf`).

Submit the two pictures and clearly label parts of the images that reveal their projection types.

Problem 2 *Orthographic projection equations* (1 point)

Recall the parallel projection equations stated in class:

$$x = \alpha X + x_0 \tag{1}$$
$$y = \alpha(\cos(\theta)Y - \sin(\theta)Z) + y_0 \tag{2}$$

which relate the coordinates of a point in the 3D world to the image coordinates of an orthogonal camera rotated by θ over the X-axis.

Show that the equations emerge naturally from a series of transformations applied to the 3D world coordinates (X, Y, Z), of the form:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \cdot P \cdot R_x(\theta) \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{3}$$

Where $R_x(\theta)$ is a 3×3 matrix corresponding to a rotation over the X axis, P is a 2×3 matrix corresponding to the orthogonal projection and α is a scaling factor to account for the size of the camera sensor, which is a single scalar when the pixels are square (assumed in this case).

Then, find α, x_0 and y_0 when the world point $(0, 0, 0)$ projects onto $(0, 0)$ (which corresponds to the center of the image) and the point $(1, 0, 0)$ projects onto $(3, 0)$.

Problem 3 *Edge and surface constraints* (1 point)

In the lecture 1 slides, we have written down the constraints for $Y(x, y)$. Briefly derive the constraints for $Z(x, y)$ along vertical edges, horizontal edges, and flat surfaces.

Problem 4 *Complete the code* (2 points)

Fill in the missing lines in the notebook: `Pset1.ipynb`, and include them in the report as screenshots. First, find a way to classify edges as vertical or horizontal edges. Next, fill in the rest of the conditions of the constraint matrix. The constraints for when the pixel is on the ground have already been done for you as an example. Put the kernel in `Aij` and the value you expect in `b` (the conversion to a linear system is done for you later so you don't need to worry about that part).

You only need to modify the locations marked with a `TODO` comment.

Please make sure to also include your answers for `vertical_edges`, `horizontal_edges`, and your formulations for `Aij` and `b` for the different constraints in the report.

Problem 5 *Run the code* (1 point)

Select some of the images included with the code and show some new viewpoints for them.

Optional: You can also try with new images taken by you if you decide to create your own simple world.

Problem 6 *Violating simple world assumptions* (1 point)

Find one example from the four images provided with the problem set (`img1.jpg`, ..., `img4.jpg`) when the recovery of 3D information fails. Include the image and the reconstruction in your writeup, and explain why it fails.

Research problem *The real world* [optional]

A research problem is a question for which we do not know the answer. In fact, there might not even be an answer. This question is optional and could be extended into a larger course project.

The goal of this problem is to test the 3D reconstruction code with real images. A number of the assumptions we have made will not work when the input is real images of a more complex scene. For instance, the simple strategy of differentiating between foreground and background segmentation will not work with other scenes.

Try taking pictures of real-world scenes and propose modifications to the scheme proposed in this lecture so that you can get some better 3D reconstructions. The goal is not to build a general system, but to be able to handle a few more situations.