MIT CSAIL

6.8300/1 Advances in Computer Vision

Spring 2024

Problem Set 1

Posted: Tuesday, February 13, 2024   **Due:** Thursday 23:59, Feb 22, 2024

6.8300/1 students are expected to finish all problems unless otherwise indicated.

The relevant material for this Problem Set was covered in Lectures 1 and 2.

Attention: If you fail to include screenshots of your code in your report, we will not be able to give you points for your code, so please do not forget.

We provide a Python notebook with the code to be completed. You can run it locally or on Google Colab. To use Colab, upload it to Google Drive and double-click the notebook (or right-click and select Open with Google Colaboratory), 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 (1) a report named `kerberos-id.pdf` including your answers to all required questions with images and/or plots showing your derivations/results and screenshots of the code you wrote. (2) There will also be a secondary Gradescope submission assignment pset 1b to upload your Python notebook, with the cells run and the relevant source code. We cannot dive into your code for grading (only for honor code violations), so missing code screenshots in the `.pdf` mean missing work.

Submission links will be released at least two days before the deadline; in the meantime we are getting section information in order.

Late Submission Policy: If your pset is submitted within 7 days 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 in discrete jumps.

Problem 1 *Perspective and orthographic projections* (10 points)

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), as long as there are some straight edges in the picture.

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 the two images on slide 38 in the lecture 1 notes or in section 2.2 of the course notes (Figure 2.2)).

In some image editing tool (e.g. Google Slides or a pdf editor), find straight edges in the pictures and add red lines highlighting and extending those edges.

(1) Submit the two pictures and (2) add a sentence to describe the differences in their projection types with reference to your red lines.

Problem 2 *Orthographic projection equations* (10 points)

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 (see book, Fig. 2.3).

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* (10 points)

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* (20 points)

Fill in the missing lines (look for `TODO` comments) in the provided notebook `pset1.ipynb`, and include them in the report as screenshots. Also, include the generated plots in your report. First, find a way to classify edges as contact, 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). (*Hint:* It may be helpful to refer to Sobel operators and discrete Laplace operators to do this section).

Problem 5 *Run the code* (10 points)

Rerun the script with at least two more of the provided images (e.g., `img2` and `img3`) and for each image try at least two different view angles. Include the generated plots in your report.

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

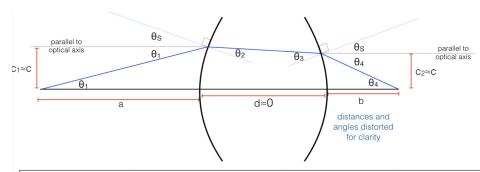**Problem 6** *Violating simple world assumptions* (10 points)

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

Problem 7 *Lenses* (20 points)

In class and the class notes, we derived the lensmaker's equation for the case of a symmetric thin lens, of radius of curvature R for the front and back surfaces of the lens. Suppose, instead, that we have a "plano-convex" lens, with the front surface shape with a radius of curvature R, as before, but with a flat back surface.

(a) (1 point) Write down the 5 equations relating the angles of rays passing through a plano-convex lens. The symmetric lens case is shown in Figure 1 (from Lecture 2 and Chapter 6 of the course notes). You may use the same variables and assumptions. A simplified diagram of the plano-convex lens is shown in Figure 2.

(b) (1 point) Using the equations from part (a) and the lensmaker's formula, find the expression for the lens focal length for the plano-convex lens described above, as shown in Fig. 2.

Problem 8 *Submit your `.ipynb` (fully run) at secondary Gradescope assignment.* (1 point)

Angle	Description	Relation	Reason
θ_1	initial angle from optical axis	$\theta_1 = \frac{c}{a}$	small angle approx.
θ_2	angle of refracted ray wrt front surface normal	$n\theta_2 = \theta_1 + \theta_S$	Snell's law, small angle approx.
θ_3	angle of refracted ray wrt back surface normal	$2\theta_S = \theta_2 + \theta_3$	symmetry of lens, thin lens approx.
$\theta_S + \theta_4$	angle of ray exiting lens wrt back surface normal	$n\theta_3 = \theta_S + \theta_4$	Snell's law, small angle approx.
θ_4	final angle from optical axis	$\theta_4 = \frac{c}{b}$	small angle approx.

Figure 1: Top: Diagram of a light ray passing through a symmetric thin lens. Bottom: Relations between angles of rays passing through the lens.
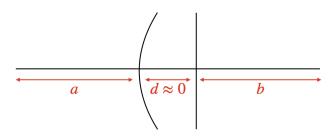

Figure 2: Partial diagram of a plano-convex lens.

Research problem *The real world* [Optional]
Note: This problem is optional for all students, both graduate and undergraduate students. No credit will be given for this problem. It is just for fun!

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 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.