CSC498 - Topics in CS: Robotics Perception

Assignment 1
Due: Feb 15, 2022, at 4 PM

1 Pen and Paper (40 Marks)

Submission: Please upload a PDF of solutions – either TeX or hand-written and scanned. Please ensure non-TeX solutions are legible or you may lose marks.

1.1 Homogeneous Coordinates (10 Marks)

1. You are given the following two lines:

$$l_1 = \left\{(x,y)^\top \in \mathbb{R}^3 \mid 2x + y + 1 = 0\right\}$$
$$l_2 = \left\{(x,y)^\top \in \mathbb{R}^3 \mid -x - 2y + 4 = 0\right\}$$

   First, find the intersection point of the two lines by solving the system of linear equations. Next write the lines using homogeneous coordinates and calculate the intersection point using the cross product. Do you obtain the same intersection point? **(5 Marks)**

2. Write down the line whose normal vector is pointing into the direction $(2,2)^\top$ and which has a distance of 3 from the origin. **(2 Marks)**

3. What distance from the origin and what (normalized) normal vector does the homogeneous line $\tilde{\mathbf{l}} = \left(1, \frac{5}{2}, \frac{\sqrt{29}}{4}\right)^\top$ have? **(3 Marks)**

1.2 Transformations (15 Marks)

1. Write down the 2×3 translation matrix which maps $(1,2)^\top$ onto $(0,4)^\top$. **(3 marks)**

2. Let's assume that you are given N2D correspondence pairs

$$(\mathbf{x}_i, \mathbf{y}_i) = \left(\begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}, \begin{pmatrix} y_1^i \\ y_2^i \end{pmatrix} \right)$$

   Find the 2×3 translation matrix mapping $\mathbf{x}_i$ onto $\mathbf{y}_i$ which is optimal in the least square sense.

   Hint: Define a cost function as

$$E(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{T}\mathbf{x}_i - \mathbf{y}_i\|_2^2$$

   and find the optimal $\mathbf{T}^\star$ which minimizes E :

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} E(\mathbf{T})$$

   Next, find $\mathbf{T}^\star$ by calculating the Jacobian $\mathbf{J}_E$ of E and setting it to $\mathbf{0}^\top$:

$$\mathbf{J}_E = \left[\frac{\partial E}{t_1}, \ldots, \frac{\partial E}{t_N}\right] \stackrel{!}{=} \mathbf{0}^\top$$

   Can you give an intuitive explanation for the equation you derive for $\mathbf{T}^*$? **(9 marks)**

3. You are given the following three correspondence pairs:

$$\begin{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -5 \end{pmatrix} \end{pmatrix} \\ \begin{pmatrix} \begin{pmatrix} 5 \\ 7 \end{pmatrix}, \begin{pmatrix} 7 \\ 6 \end{pmatrix} \end{pmatrix} \\ \begin{pmatrix} \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ -4 \end{pmatrix} \end{pmatrix}$$

Use the derived equation, calculate optimal 2×3 translation matrix $\mathbf{T}^\star$. **(3 marks)**

1.3 Camera Projections (15 Marks)

1. Calculate the full rank 4×4 projection matrix $\tilde{\mathbf{P}}$ for the following scenario **(4 Marks)**:
   - The camera pose consists of a 90° rotation around the x axis and translation of $(2, 0, 2)^\top$.
   - The focal lengths f_x, f_y are 100 .
   - The principal point $(c_x, c_y)^\top$ is $(25, 25)$.
2. For the previously defined projection, find the world point in inhomogeneous coordinates $\mathbf{x}_w$ which corresponds to the projected homogeneous point in screen space $\tilde{\mathbf{x}}_s = (250, 500, 10, 2.5)^\top$. **(3 Marks)**
3. Let's perform our first projection of a geometric shape. We define $\mathcal{C}_0$ as the cube centered at $\mathbf{c}_c = (0, 0, 25)^\top$ with equal side lengths $s = 20$.
   (a) Project the 8 corners of the cube $\mathcal{C}_0$ to the image plane for the pinhole camera with focal lengths $f_x = f_y = 5$ and the principal point $(c_x, c_y)^\top = (10, 10)^\top$. **(4 Marks)**
   (b) Project the 8 corners of the first cube $\mathcal{C}_0$ using an orthographic projection and add the principal point $\mathbf{c}_c = (10, 10)^\top$ onto the obtained pixel coordinates to be in the same coordinate system as before. **(4 Marks)**

2 Coding Exercises (60 Marks)

The following exercises will be coding exercises. They are all contained in the colab notebook [link]. The notebook itself is self-contained but you can also use this document as guidance. If you are stuck, you can find Hints in the notebook itself which are written upside-down.

Submission: Submit your code and results in an IPython notebook (.ipynb) with all the **images and video outputs embedded** in it. You will lose marks if your notebook works but doesn't contain outputs.

2.1 Perspective Projection (25 Marks)

1. Implement the function get_camera_intrinsics which takes as input the focal lengths f_x, f_y and the principal point $(c_x, c_y)^\top$ and returns a 3×3 camera intrinsics matrix $\mathbf{K}$.**(5 Marks)**
2. Implement the function get_perspective_projection which takes as input a 3D point in camera space $\mathbf{x}_c$ and the camera matrix $\mathbf{K}$ and it returns a two-dimensional point in screen space, hence the pixel coordinates $\mathbf{x}_s$ for the 3D point $\mathbf{x}_c$. **(7 Marks)**
3. Implement the function get_face_color which maps a normal vector $\mathbf{n}$ and a point light direction vector $\mathbf{r}$ to an RGB color value. We assume the light to be a point light source,i.e., the light source is infinitely small. You should follow the rendering equation discussed in the lecture where you can assume that the surface does not emit light, the BRDF term is always 1 , and the incoming light L_{in} is 1 for the point light source direction. **(3 Marks)**

4. Create a visualization where you change both focal lengths similar to the previously given example. **(5 Marks)**

5. Create a visualization where you translate the cube along the y-axis between $[-2, 2]$. **(5 Marks)**

2.2 Orthographic Projection (5 Marks)

1. Implement the function get_orthographic_projection which maps a 3D point in camera space $\mathbf{x}_c$ to 2D pixel coordinates $\mathbf{x}_s$ using a orthographic projection. **(3 Marks)**

2. To confirm your analysis, plot a projected cube with center $(0, 0, 150)^\top$ and focal lengths $f_x = f_y = 10000$. When does the perspective projection look most similar to the orthographic projection? **(2 Marks)**

2.3 Panorama Stitching using DLT (30 Marks)

1. Implement the function get_Ai which takes as input the two homogeneous points $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_i$ and returns the 2×9 matrix $\mathbf{A}_i$ discussed in the lecture. (Note: It's 2×9 and not 3×9 as you can drop the last row (see lecture).) **(10 Marks)**

2. Implement the function get_A which uses the get_Ai for all N correspondence pairs and concatenates them to return the $2N \times 9$ matrix $\mathbf{A}$. **(5 Marks)**

3. Implement the function get homography which maps the N point correspondences to the homography $\mathbf{H}$ using the Direct Linear Transform (DLT) algorithm. **(15 Marks)**