Coding challenge

1 Task

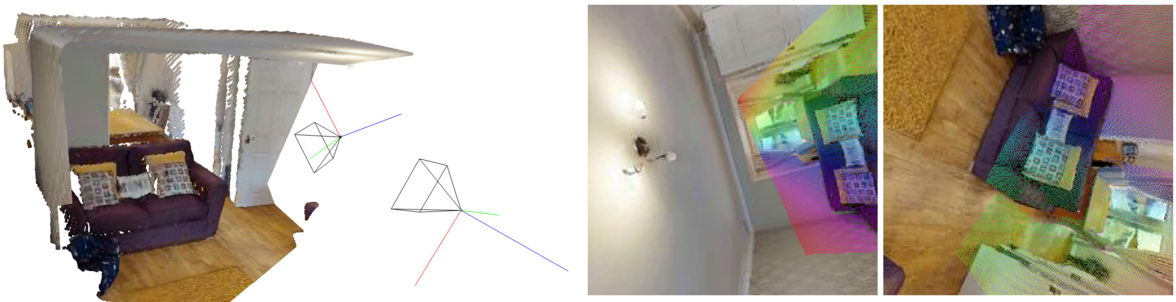
You are provided with an unprocessed dataset consisting of rectified RGB images, associated depth maps and camera intrinsics (field of view) and extrinsics (6D pose consisting of 3D position and rotation quaternion). You are given a basic data loading script that loads a pair of images with their associated metadata. The task is to implement a function that computes for each pixel in image A whether that pixel is visible in image B, and if so, to which pixel it maps to in image B.

2 Requirements

- Implement the code in Python 3, C++, or Rust. You may use any library you see fit.
- Your code should be reasonably performant (it should take less than 10s per image).
- Visualize the matched pixels color-coded for each of the five evaluation image pairs similarly to this sample image. Export them as `result_0.png` to `result_4.png`. In our example in Figure 1b, we choose a hue-color coding based on the polar coordinates from the origin of image A.

3 Notes

- For batched processing, the use of PyTorch may be helpful and is encouraged.
- For dealing with pose transformations, roma may be helpful.
- For visualizing 3D pointclouds and the camera coordinates, open3d may be helpful. We provide a utility function to render a camera gizmo `create_camera_gizmo` in the appropriate camera coordinate system, which may be useful for debugging purposes. It may be helpful to you to first generate two pointclouds in camera coordinates based on the depth image and the camera parameters, and visualize them in open3d to make sure your transformations are correct.
- We furthermore provide basic data loading functions that will help you to get started, but you do not have to use them.
- The camera coordinate system is right-handed, y-up, with the camera facing towards negative z (right = red = x+, up = green = y+, forward (into the scene) = z- (negation of blue z+ vector)). We visualize this in Figure 1a.
- If you have any questions, don't hesitate to ask us.



(a) The camera coordinate system used in this assignment.

(b) Result sample for the scene shown left. Left: image A, right: image B. Corresponding pixels between A and B are colored in similar colors.

Figure 1: Reference figures.