The Foundations of Computer Vision

ECE 188, Spring 2023
Project: Vision-Based 3D Reconstruction and Recognition System

Stage 2:

In an effort to avoid confusion with the KITTI dataset camera parameters, we have simplified the second stage of the project, and split it up into two separate parts: rectification and stereo matching. A short description of the two parts will be given below, as well as benchmarks. In the folder provided for this stage, you should find two subfolders: **stage2_rectification** and **stage2_stereo**, giving you the data needed for rectification and stereo matching, respectively.

Rectification:

Frequently in computer vision, when computing depth from stereo images, we first need to rectify the images such that their epipolar lines coincide and are parallel to the x-axis of the image. This typically requires use of the camera matrix, and therefore, the camera intrinsics. To simplify the usage of the KITTI dataset, we will instead be giving you an input image, and an image which has been warped using a randomly generated homography. It will be your job to warp this image back to its original perspective. Input images can be found in **input_imgs**, and warped images are found in **warped_imgs**.

Similar to the first stage, we will be using PSNR and SSIM between your rectified images and the original input images as metrics. The benchmark for this section will be an average of **30.0 PSNR** and **0.90 SSIM** between all 11 images. Keep in mind that these are soft benchmarks, and you do not have to pass them to get an A on this project. We will be grading you based on the methods used and the code given. We also provided in the folder a **CalculateMetric.ipynb** to use to calculate the PSNR and SSIM of an image pair, since warped images will have black borders which make it hard to calculate metrics normally.

Stereo:

In computer vision, one of the common methods for generating a 3d reconstruction of the scene is through stereo, which uses two cameras placed at some known distance apart. By looking at the same features in the two images, and calculating the pixel disparity between them, we can use triangulation to calculate the depth (therefore 3d reconstruction). In this part, you will be taking in the left and right camera image pairs and calculating a disparity map. The left and right images will be in **left_imgs** and **right_imgs**, respectively. The ground truth disparities will be in **disparities**, and will be in the left image reference frame.

The metric used for this stage will be root mean squared error (RMSE). The benchmark for this section will be **3.5 pixels** RMSE. Keep in mind that the disparity map generated by you and the ground truth disparity map will be sparse, meaning some pixels will be zero by default. This RMSE is calculated by masking out all pixels that do not both have a nonzero value in the ground truth and calculated disparity map. Also keep in mind that the units have to be in **pixels** for this RMSE benchmark to make sense. Check the Week 6 discussion for more help on this.