ECES 681Computer Vision Homework 2

Somayeh Keshavarz

This assignment is due on **Jul 30, at 11:59 PM EST**.

Submission:

You need to submit:

- A PDF (hw2_writeup.pdf) containing answers to all questions, along with plots and visualization.
- Python Code (hw2_code.ipynb) including all functions.
- Ensure that the **Jupyter Notebook** is self-contained, functional, and includes all outputs.

Homework: Creating a 3D Stereo Image from Two Images Without Camera Intrinsic Parameters

Objective

In this assignment, you will generate a **stereo 3D image** using two images taken from different viewpoints. Since camera intrinsic parameters are unknown, you will estimate depth using **uncalibrated stereo rectification** and **disparity computation**.

Instructions:

Step 1: Detect and Match Feature Points

- Use **feature detectors** such as **SIFT, ORB, or Harris Corner Detector**.
- Find **corresponding points** between the two images using descriptor matching.

<u>Visualize the location and scale of the thirty strongest features in image1 and image 2.</u>

Step 2: Compute the Fundamental Matrix FF

- Estimate the **Fundamental Matrix FF** using **RANSAC** to remove outliers.

Step 3: Draw Epipolar lines

- Then sample 3 pixels on image 1 and draw their epipolar lines on image 2 using the computed fundamental matrix. Visualize the result.

Step 4: Rectify the Images

Step 4.1. Use **cv2.stereoRectifyUncalibrated()** to compute **rectification homographies**.
cv2.stereoRectifyUncalibrated() is an **OpenCV function** used to **rectify stereo images when camera intrinsic parameters are unknown**. It estimates the transformation required to align both images so that **epipolar lines become horizontal**, making stereo matching easier.

What is Stereo Rectification?

- **Stereo rectification** warps both images so that corresponding points **lie on the same row**.
- This simplifies **disparity calculation**, which is essential for **depth estimation**.

Without Rectification

- Corresponding points appear **at different heights**.
- Stereo matching is **difficult and computationally expensive**.

After Rectification

- Corresponding points **align on the same row**.
- Disparity is measured **along the x-axis only**.

Returns

- **h1** → Homography matrix for the **left image**.
- **h2** → Homography matrix for the **right image**.

Homographies H1H1 and H2H2 are computed so that the new epipolar lines become **parallel and horizontal**. These homographies **transform both images** to a common rectified plane.

Step 4.2 : Apply the Homographies

- The images are **warped** using H1H1 and H2H2, ensuring that **corresponding points align on the same row**.

Deliverable: Display the **rectified images** and explain how rectification helps in stereo matching.
Rectify the stereo images and display them as a stereo anaglyph. You can use red-cyan stereo glasses to see the 3D effect. Use AxesAnaglyph() from mpl_stereo library.

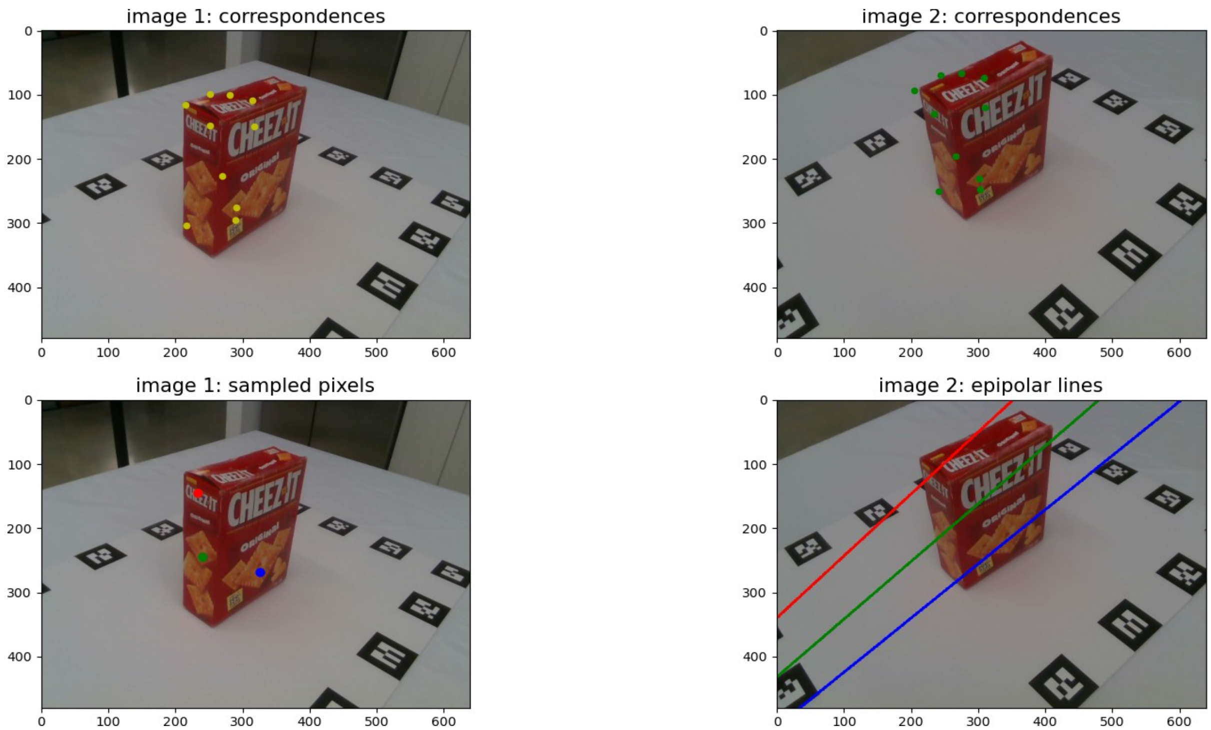
image 1: correspondences
image 2: correspondences
image 1: sampled pixels
image 2: epipolar lines
0
100
200
300
400
500
600