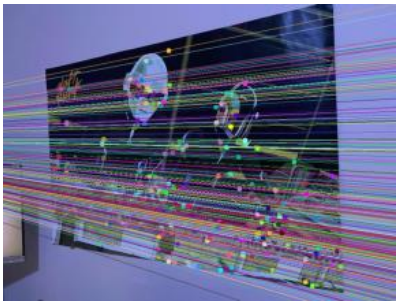
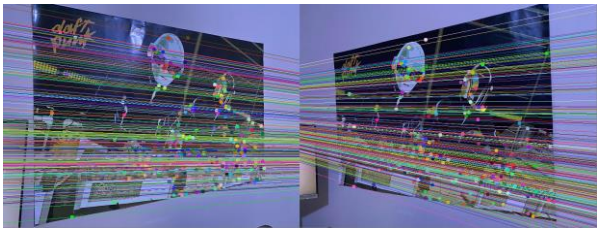




ELEC 474

Prelab 5 – Epipolar Geometry

Prelab 5 – Epipolar Geometry

Kevin Huang and Michael Greenspan

Contents

1. Epipolar Lines

For this prelab you will make use of feature detection to detect matches between two stereo images and then perform the calculation of epipolar lines between the two images. Depending on which feature detection you use you might have to import the ***opencv_contrib*** library.

You can install the ***opencv_contrib*** library by typing into the Anaconda Prompt:

```
pip install opencv-contrib-python
```

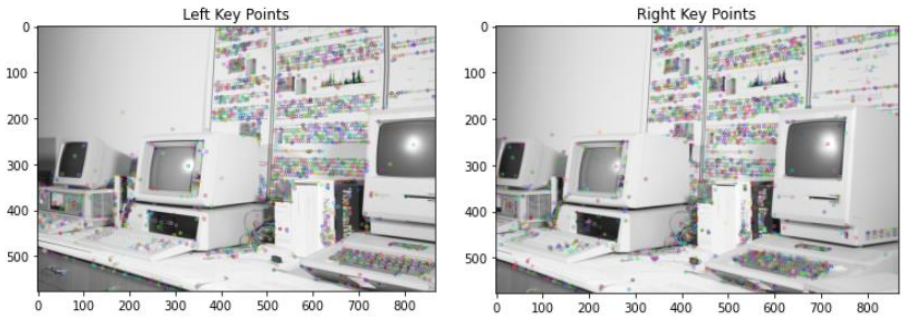
Once installed, you can now use the SIFT algorithm implementation in Python.

1.1 Feature Matching

For this portion of the prelab use any feature detection algorithm to find your matches. Follow these steps for your feature matching:

1) Load in two images for matching.
2) Initialize any feature detection module and detect and computer your **key points** and **descriptors**.
3) Using any **cv2** matcher and find matches with your **descriptors**. When computing, set your matcher output (**k=2**) to output **two** possible matches.
4) Filter your matches with **Lowe's ratio** but instead of saving the **match**, save both the **left and right points** of the **match**.

Left and Right key points for "computers_left.png" and "computers_right.png":



1.2 Epipolar Lines Calculation

To calculate the Epipolar Lines between each of your images you must find the **Fundamental Matrix** which is calculated using both of your **left and right** points from your **matches**. Follow these steps to find your **Fundamental Matrix** which will then be used to calculate your Epipolar Lines.

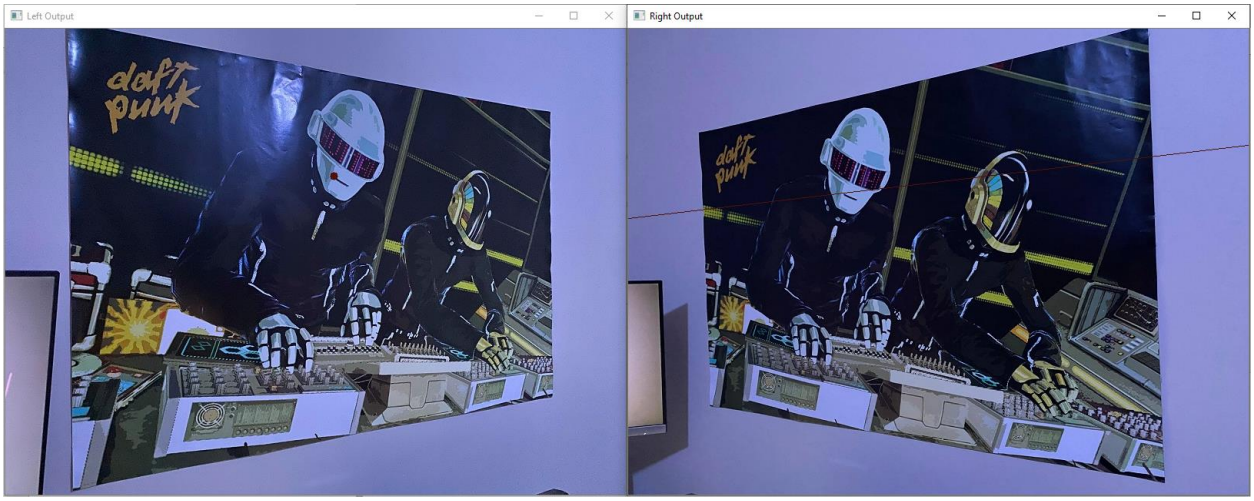
1) Convert your **left and right** points into **int32 format**.
2) Retrieve your **Fundamental Matrix** as well as your inliers mask by using **cv2.findFundamentalMatrix()**.
3) Mask your **left and right** points using the inliers mask and **.reshape(-1, 1, 2)** to point format.
4) Use **cv2.computerCorrespondEpilines()** and your **Fundamental Matrix** to calculate you Epipolar Lines. A **line equation** is returned which are three values **(A, B, C)** which correspond to the general form of a straight line: **Ax + By + C = 0**.
5) With your collection of lines create a **Python function:**
    - That passes an image, epipolar lines, left points, and right points. This function will draw all your **Epipolar lines for one image**. This function should use each **line equation** to calculate two **bounding** points. One point should correspond to the 0^{th} column of the image (Left Bound) and the other should be the column size of the image (Right Bound).
    - Once you have left and right bound points, use **cv2.line()** and **cv2.circle()** to draw your epipolar lines and point correspondences.
6) Perform steps 4) and 5) on both the **left and right** images and output the images in a **stacked cv2** window:

computer_left.png computer_right.png:

daft_punk_left.jpg and daft_punk_right.jpg:

1.3 Point/Conjugate Epipolar Line Selection

To convince yourself that this is actually working correctly, implement an interface that allows you to select a point in one image (e.g. with a mouse click), and display the conjugate epipolar line in the other image. For a single selected point in the left image (the red dot at the corner of the white-masked musician's mouth), the output would look like this:



Notice that the conjugate epipolar line in the right image intersects closely with the selected point from the left image, which is an indication that the Fundamental Matrix was calculated accurately.

2. Submission

The submission for this prelab should include a .zip of:

- .ipynb file that includes:
  - All of your code for Epipolar Lines Calculation and Visualization (Steps 1.1 and 1.2)
    - Test your code on stereo images (computers_left.png and computers_right.png) and left and right images (daft_punk_left.jpg and daft_punk_right.jpg)
    - Your code should display images of your key points, matching points, and epipolar lines.
  - All of your code for Point/Conjugate Epipolar Line Selection (Step 1.3)
    - Your code should execute on either of the two provided stereoimage pairs.

Your code will be run in Jupyter Lab to test for functionality. The marking rubric is as follows:

Section	mark
1.2 Epipolar Lines Calculation	1.5
1.3 Point/Conjugate Epipolar Line Selection	1.0
Correct submission format	0.5
Total:	**3**