**README**

* Run the program.
* You can choose the question you wanted. (1 or 2)
* For the 1. Question, program shows left and right images, raw disparity map and a filtered color scaled depth map.
* Press any key to continue to next image.
* For the 2. Question, program shows vector maps for both cameras.
* Feature points will be shown in command line for the current left frame.
* Press any key to continue and it will show epipolar lines with their corresponding feature points on the both images.
* The mismatch error of lines with the real points will be shown in command line for the current images.
* Press any key to repeat the same process with the next image.

**EE 576 HW10**

**Methodology:**

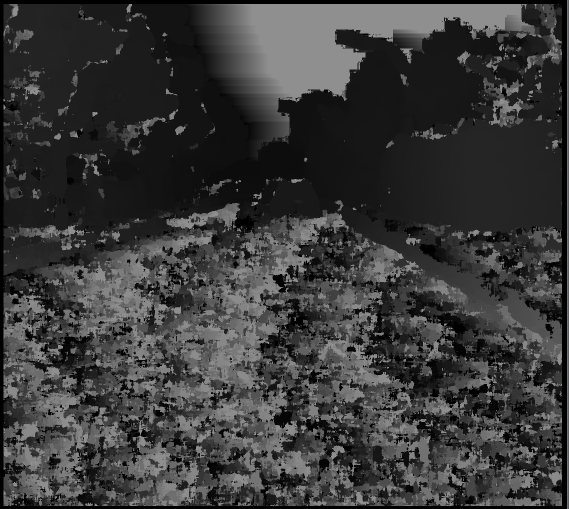
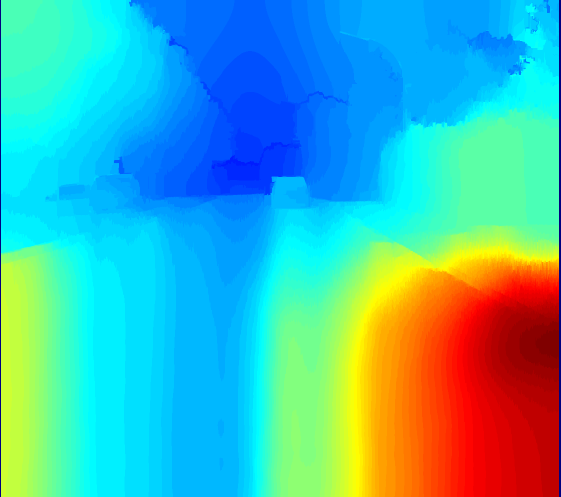
For this assignment, I used “StereoBM” OpenCV function to calculate disparities and “getDisparityVis” function to visualize it. For better results, I filtered the disparity map using “createDisparityWLSFilter” function, normalized and color scaled them using “COLORMAP\_JET”. For the second part, I used sparse optical flow code I wrote for HW9 and after finding the Harris features, I used “findFundamentalMat” function to find fundamental matrix and “computeCorrespondEpilines” function to find epipolar lines.

**Dense Stereo**

*Left Camera Right Camera*

Algorithm was successful for recognizing the depth of the images. Both in raw map and the color scaled map we can recognize the distance differences of the trees and other objects to camera along the way. For the raw map, it shows the distances as the shade of the grey goes darker but fails to show the true distance of the air (which is understandable) and the filtered version over emphasize the closeness of the corners probably because of the reflection on the bottom corner of the images but gives the depth feeling successfully and smoothly.

*Raw Disparity Map Filtered Color Scaled Depth Map*

**Epipolar Lines and Feature Points**



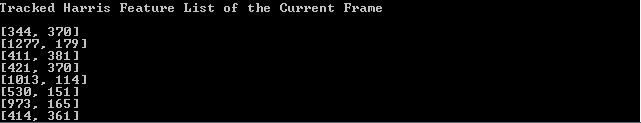
*Feature points and their disparity vectors on left image*

At the first two image, we can see the disparity vectors of both images. Left image points have vectors which point towards left and for right image, its vice a versa. This is because, when you switch to right camera from left camera, view is similar to a camera moved towards right. Image goes reverse way.

The next images show epipolar lines with the feature points. I preferred showing only the first 8 feature point (because we need minimum 8 points to find F matrix) and their epipolar lines in those images because this way, seeing whether matched features lie on the epipolar line or not is easier. For the first couple of images matching is nearly perfect but later errors got bigger (as can be seen in example image). This is probably proportional to the quality of the image and the epipolar line number because also the detected feature point numbers are getting lower for next images. Therefore, I was forced to use of less quality feature points for the last images. Otherwise there wasn’t enough points to construct fundamental matrix. For the images we can see the epipole of that image inside the image as a crossing of all epipolar lines.



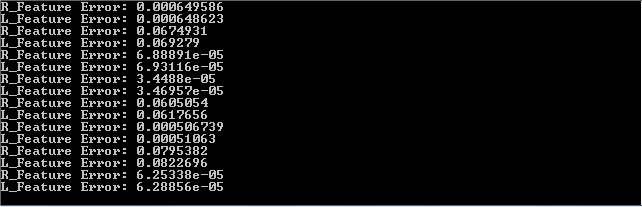
*Feature points and their disparity vectors on right image*

**

*First 8 feature point coordinates for the left image*



*Epipolar lines drawn from right image and feature points of the left image*

**

*The mismatch error of the first 8 points on both images with their corresponding epipolar lines.*

**

*Epipolar lines drawn from left image and feature points of the right image*

**

*An example of with all epipolar lines drawn*

**

*An example of high misalignment of epipolar lines with feature points*

**References**

1. EE576 Homework 9
2. <https://docs.opencv.org/master/d3/d14/tutorial_ximgproc_disparity_filtering.html>
3. <http://ros-developer.com/2018/12/21/computing-fundamental-matrix-and-drawing-epipolar-lines-for-stereo-vision-cameras-in-opencv/>

**Fehmi Ayberk Uçkun**

**2015401009**