Name: Denny-Thomas-Varghese; SID: 55506653

CS4186 Computer Vision Assignment 2 Report

The following report details the final method used for construction of disparity maps between the pairs of images provided, namely in directories Art, Dolls and Reindeer.

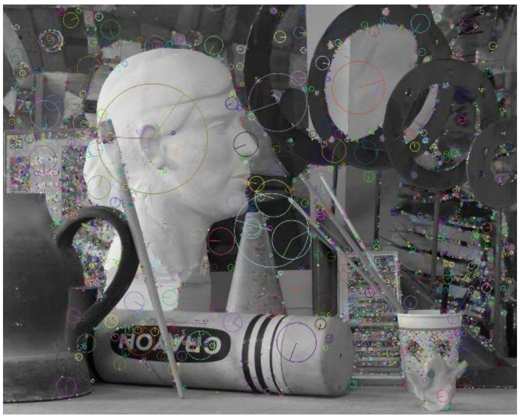
Major Steps in the Method:

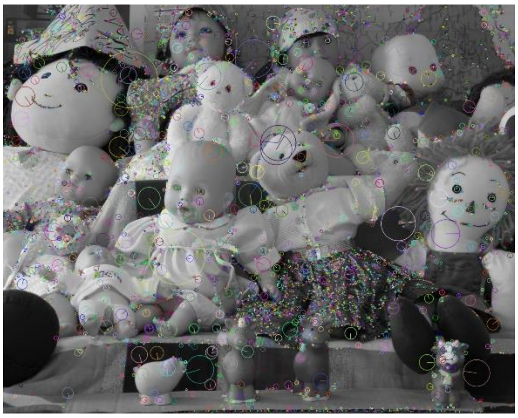
For each pair of images, the following steps were followed:

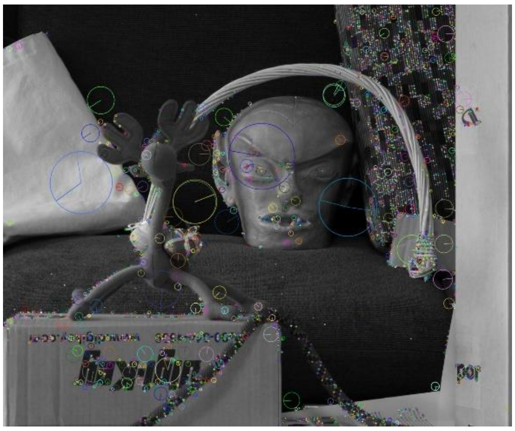
Step 1:

Use SIFT or xfeatures2d.SIFT_create() to detect features (using the sift method sift.detectAndCompute() i.e. get keypoints and descriptors, of the pair of images.

Here are the results from this operation:



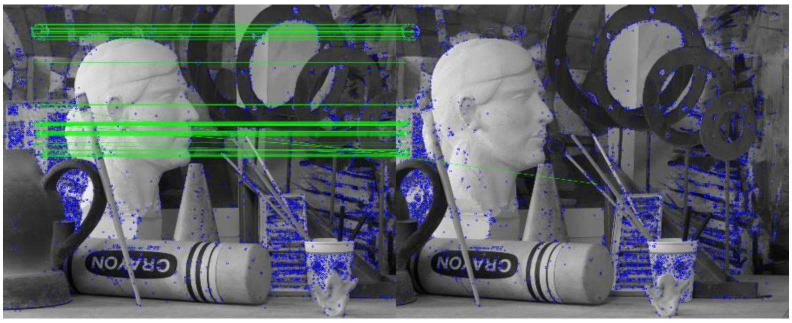


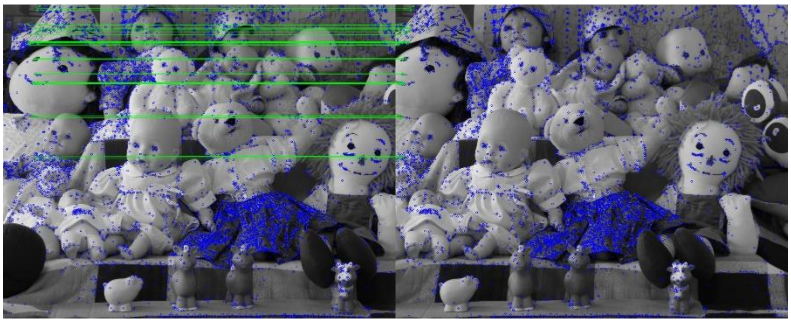


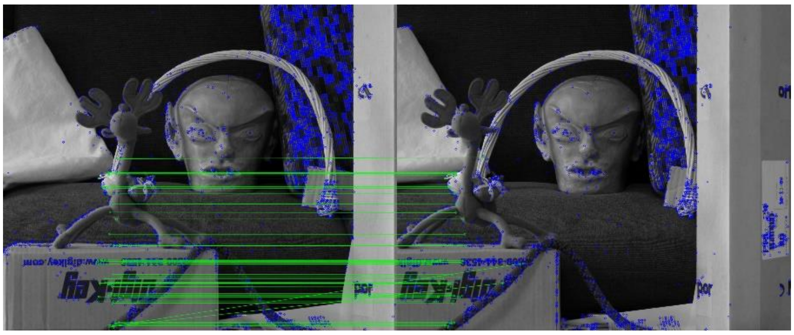
Step 2:

Use the FLANN matcher to match the keypoints of both the images. The FLANN matcher's index parameters here use the KDTREE algorithm and 5 trees. Then obtain the matches with knnMatch() with k=2. To obtain and keep only the distinctive or "good" feature matches, the matches are enumerated through and the ones with distance less than the threshold (here, 0.7) are kept.

Here are the results from this operation:







Step 3:

Use cv.findFundamentalMat() using the keypoints of the first and second image, and cv.FM_RANSAC to obtain the fundamental matrix. Select only the inlier points in both the keypoints arrays by using inliers.ravel()==1.

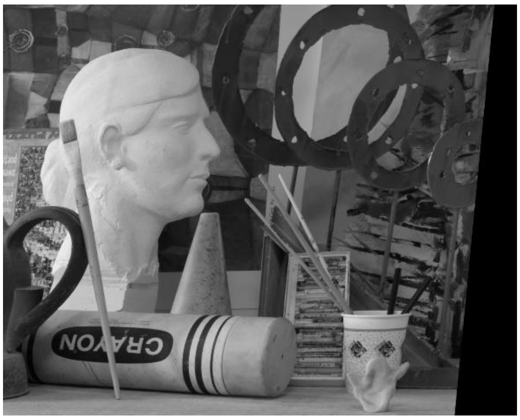
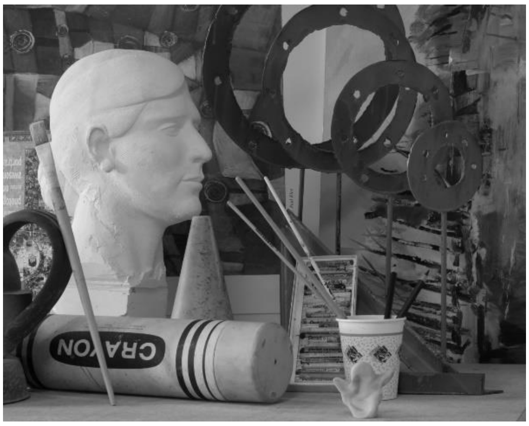
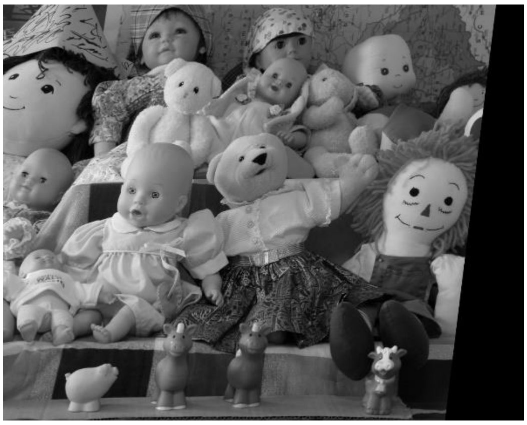
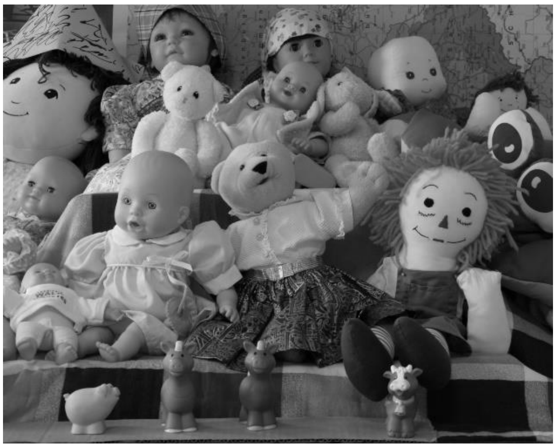
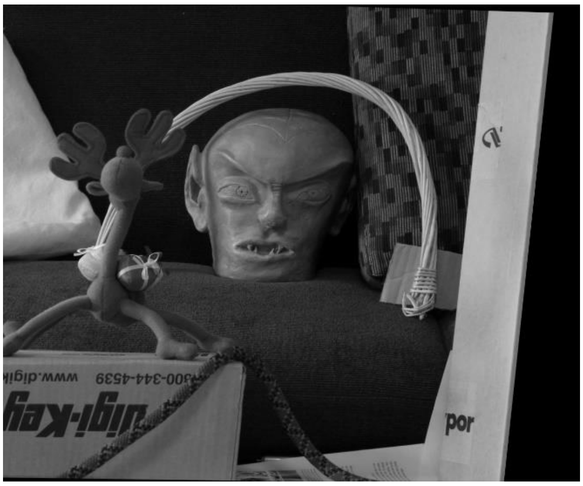
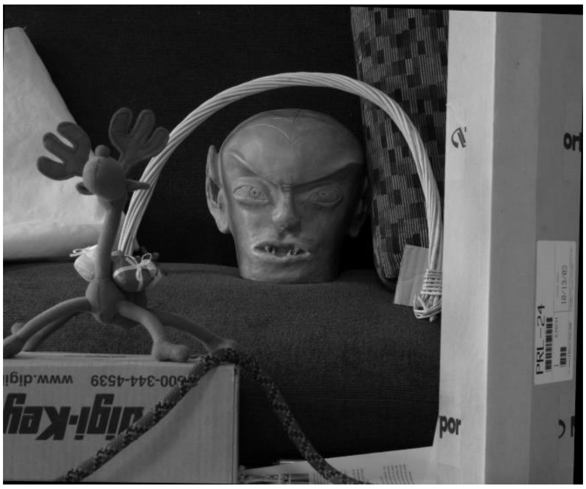
Step 4:

Once the fundamental matrix and the inlier keypoints are available, obtain the epilines on the first image by using cv.computeCorrespondEpilines() along with the previously obtained keypoints array of the other image and the fundamental matrix, and vice versa. This way, epilines on both of the images can be obtained.

Step 5:

Use cv.stereoRectifyUncalibrated() along with the 2 keypoint arrays, fundamental matrix and the image-size (width, height). This returns the homographies for both the images. Then, use cv.warpPerspective() with the image, its homography and its image-size to obtain the rectified image. Do this for both the images.

Here are the results from this operation:

Step 6:

Use cv.StereoSGBM_create() with parameters minDisparity, numDisparities, blockSize, uniquenessRatio, speckleWindowSize, speckleRange, disp12MaxDiff, P1 and P2. The values used for these parameters are based on empirical evidence and suggestions from online forums. For example, for the parameter uniquenessRatio, it was found that a value >5 would result in inferior performance.

Use the compute() method of the StereoSGBM variable created above, along with the two rectified images to obtain the disparity map of the image passed as the first parameter of the compute() method. This understanding will help in the next and final step.

Step 7:

By taking a look at the previous result (disparity map), it is easy to deduce that some form of "smoothening" could make it look closer to what the ground truth disparity maps look like. Create a matcher for the second image using cv.ximgproc.createDisparityWLSFilter(), passing the matcher for the first matcher as parameter. Then use that matcher to obtain the disparity map of the image passed as the first parameter (here, the second image). The parameters lambda and sigma are set to 35000 and 1.5 upon suggestions from online forums. The first image is the second parameter here. A filtered disparity map is then created and normalized.

This disparity map is the final one used to measure performance.

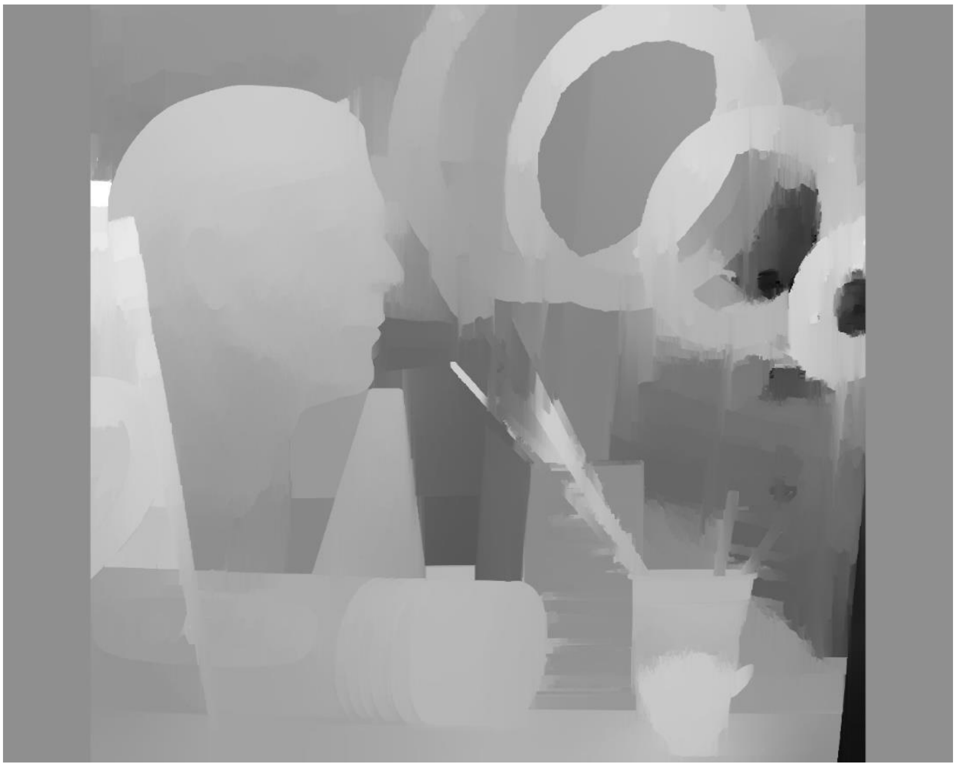
Running the program:

Please note that in order to run the program successfully, the folder structure for the Drive images and the result images must be followed. The following folder structure is used to run this program and save results. At the same directory level as the python program, the following should be present:

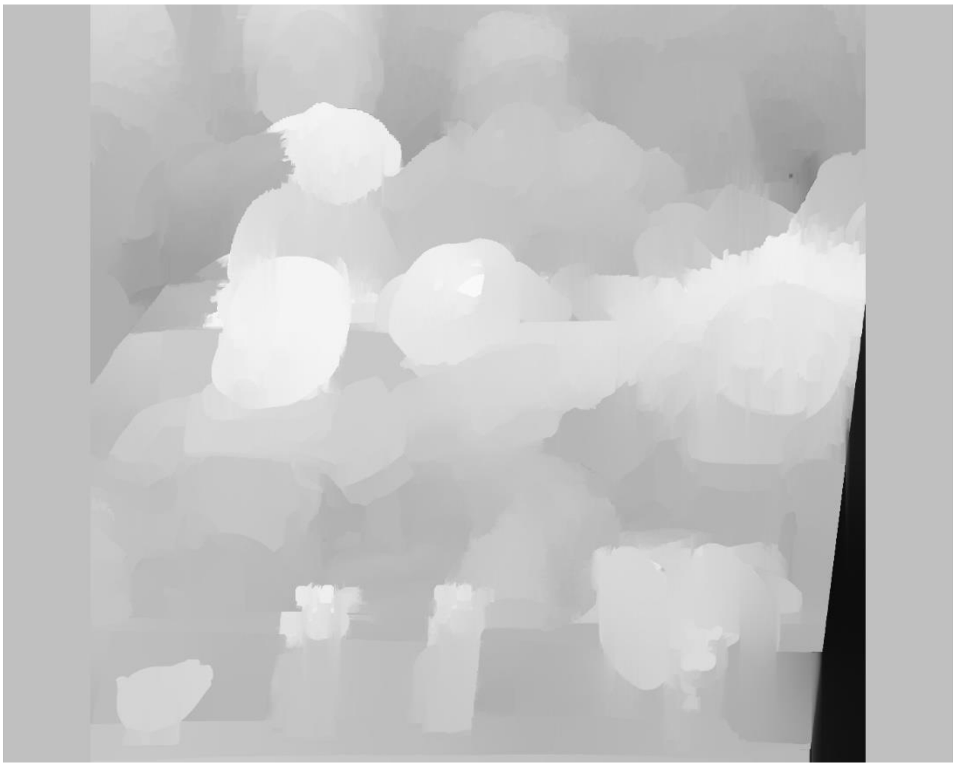
- A folder named "StereoMatchingTestings" containing three subfolders, namely "Art", "Dolls" and "Reindeer" each containing the 2 views or images.
- A folder named "allResults" containing folders "pred" and "others". "pred" will contain the final disparity maps in folders "Art", "Dolls" and "Reindeer". It is designed in this manner to make it easier to test with the PSNR python script provided.

Results:

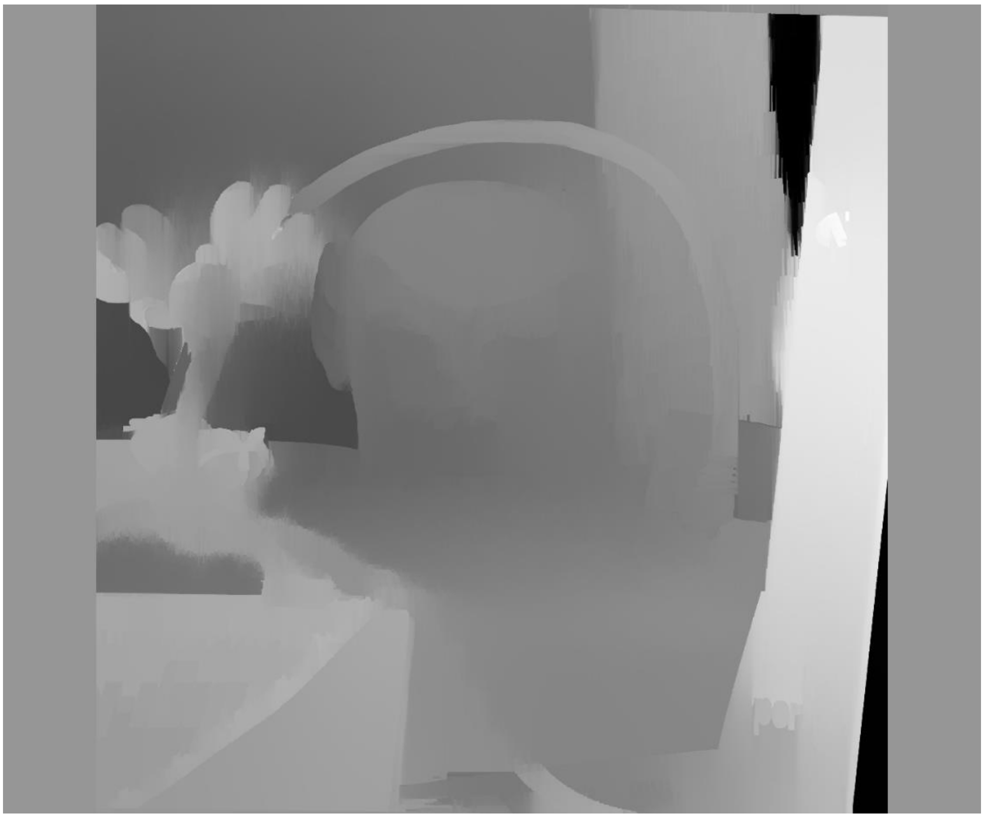
The Disparity Maps obtained for Art, Dolls and Reindeer pairs of images / view are as follows:



From Art

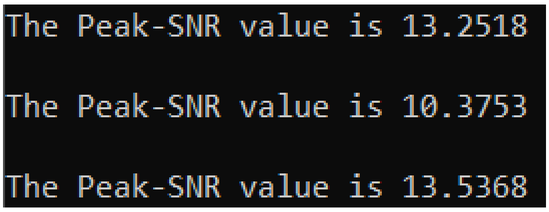


From Dolls



From Reindeer

The PSNR Results obtained for Art, Dolls and Reindeer (in that order) by running the PSNR script provided is as follows:



```
The Peak-SNR value is 13.2518

The Peak-SNR value is 10.3753

The Peak-SNR value is 13.5368
```