Desription of solution

Nadezhda Alsahanova

GitHub repository with solution

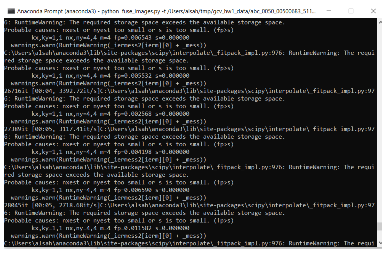
1. In function *get_view* the new CameraPose class object (pose_i) with extrinsics matrix for image_i was created. Also, the new RaycastingImaging class object (imaging_i) with image resolution and resolution_3d from intrinsics dictionary for image_i was created. To get point cloud from depth image function image_to_points from imaging_i was used. To transform the image to points in world frame camera_to_world function from pose_i was used. So, points_i from depth image_i in world frame was constructed.

2. In function *pairwise_interpolate_predictions* to be able to interpolate in view_i points from view_j were reprojected to view_i. For reprojection function world_to_camera from camera pose_i was used. So reprojected_j was got.

3. Then we found k (nn_set_size) nearest points for each of the points from reprojected_j. This point was from view_i. So indexes of these points(nn_indexes_in_i) were got.

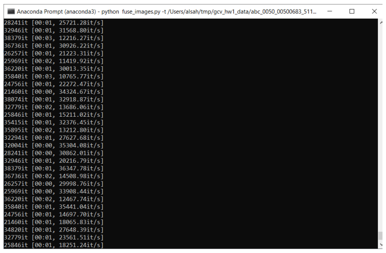
4. Before building interpolation, we need to check the possibility of it. Firstly, for points_i in the pixel grid of view_i with indexes (point_nn_indexes) that was got from nn_indexes_in_i we created an array. This array was consists of X and Y coordinates of this point and Z coordinate that was from image_i. So we got point_from_j_nns. Secondly, we found a distance between this point and point_from_j. If this distance less than a particular threshold the interpolation is possible.

5. If interpolation is possible, we made a bilinear interpolator from distances predicted in view_i (distances_i) into the point in view_j. For this, we can use interpolate.interp2d function from scipy. But it is not


Anaconda Prompt (anaconda3) - python fuse_images.py -t /Users/alsah/tmp/gcv_hw1_data/abc_0050_00500683_511...

```
6: RuntimeWarning: The required storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.006543 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:976: RuntimeWarning: The requi
red storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.005532 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
26716it [00:04, 3392.72it/s]C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:97
6: RuntimeWarning: The required storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.002568 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
27389it [00:05, 3117.41it/s]C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:97
6: RuntimeWarning: The required storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.004198 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:976: RuntimeWarning: The requi
red storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.006590 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
28045it [00:05, 2718.68it/s]C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:97
6: RuntimeWarning: The required storage space exceeds the available storage space.
Probable causes: nxest or nyest too small or s is too small. (fp>s)
        kx,ky=1,1 nx,ny=4,4 m=4 fp=0.011582 s=0.000000
  warnings.warn(RuntimeWarning(_iermess2[ierm][0] + _mess))
C:\Users\alsah\anaconda3\lib\site-packages\scipy\interpolate\_fitpack_impl.py:976: RuntimeWarning: The requi
```

Figure 1: interp2d


Anaconda Prompt (anaconda3) - python fuse_images.py -t /Users/alsah/tmp/gcv_hw1_data/abc_0050_00500683_511...

```
28241it [00:01, 25721.28it/s]
32946it [00:01, 31568.80it/s]
38379it [00:03, 12216.27it/s]
36736it [00:01, 30926.22it/s]
26257it [00:01, 21223.31it/s]
25969it [00:02, 11419.92it/s]
36220it [00:01, 30013.35it/s]
35840it [00:03, 10765.77it/s]
24756it [00:01, 22272.47it/s]
21460it [00:00, 34324.67it/s]
38074it [00:01, 32918.87it/s]
32779it [00:02, 13686.06it/s]
25846it [00:01, 15211.02it/s]
35415it [00:01, 32376.45it/s]
35895it [00:02, 13212.80it/s]
32294it [00:01, 27627.68it/s]
32004it [00:00, 35304.08it/s]
28241it [00:00, 30862.01it/s]
32946it [00:01, 20216.79it/s]
38379it [00:01, 36347.78it/s]
36736it [00:02, 14508.98it/s]
26257it [00:00, 29998.76it/s]
25969it [00:00, 33908.44it/s]
36220it [00:02, 12467.74it/s]
35840it [00:01, 35441.04it/s]
24756it [00:01, 14697.70it/s]
21460it [00:01, 18065.83it/s]
34820it [00:01, 27648.39it/s]
32779it [00:01, 23561.51it/s]
25846it [00:01, 18251.24it/s]
```

Figure 2: bisplrep and bisplev

stable 1. For stable interpolation, we can use interpolate.bisplrep and interpolate.bisplev from scipy 2.

All changes were made in files *repo/gcv_v20211_hw1/fusion/interpolators* with unstable interpolator and *repo/gcv_v20211_hw1/fusion/interpolators_s* with stable interpolator. Also, in notebook *repo/notebooks/gcv_v20211_hw1.ipynb* some results are perfomed.