

Assignment 3

Ehsan Emad Marvasti

Method

in order to obtain panoramic view the algorithm used 4 steps

1- find the corresponding points with sift.

2- obtaining the sift keypoints with `ubc_match`. `ubc_match` finds the corresponding points based on the descriptor distance for each two points.

3- using ransac in order to remove outliers in order to find the infinite homography
`ransacfindhomography` used and threshold .01 was selected for the outlier measure.

4- using the homographies the points in each image are mapped to a reference image with the size of maximum and minimum boundaries. I implemented the function to map several warped images to a single image.

When the algorithm computes the warp image since it is doing an interpolation the result image may have small values on the edge that is close to zero so in order to merge the photos i tried to look for pixels that are equal to zero and replace them with the new image. Since those values are not equal to zero the output will have black lines across the borders of each warped image. In order to fix this problem i used the value nan as the filling value.

Other problems that occurred was determination of the levels of sift in order to reduce total keypoints which i found the right level value by try and error.

Code Description

In order to simplify the code i have created an option object to pass to the mosaic function.

Opts has fields such as

`opts.sift.totalKeypoints`

`Opts.sift.levels`

`opts.sift.loadKeypoints`

Total keypoints are the keypoints that are passed to the homography finder. The set of all keypoints are permuted in a random order in order to have distributed keypoints in the image.

Levels define the total levels of sift algorithm which is set to 5.

Load keypoints can be set to true in order to avoid the program to recompute the sift keypoints.

If the sift keypoints does not exist in the folder the program will compute the sift keypoints and store them in a file in the program root folder.

`Opts.plot.descriptor.active`

`opts.plot.descriptor.indexRange`

`Opts.plot.descriptor.total`

This options will correspond to plotter function to visualise the sift performance

The index range defines the index of photos to be shown for the descriptors found for them
Descriptor.total will limit the total visualized descriptor in the image by randomly permuting the the keypoints

```
Opts.plot.mapper.activate  
opts.plot.mapper.indexRange  
opts.plot.mapper.totalPoints
```

This works the same as the descriptor plotter. The mapper will show images from the index range and their neighbor images and shows the corresponding points selected by ransac in order to compute homography.

RESULTS

As we can see i did not use the featuring algorithm to fix the illumination problem since the photos was taken by iphone that has auto exposure and etc so the illumination of the photo will change as the camera moves.



