

1 Goal

The objective of this homework is create a panoramic image from a set of single shot obtained from the rotation of the camera. To do this is required the construction of the `PanoramicImage` class that, given a set of images, compute the panoramic image.

2 Procedure

To achieve the goal the first step is load all the images inside the `PanoramicImage` class through the constructor or the method `addImages()`. The constructor take moreover an other parameter, the field of view, used to compute the cylindrical projection of the image made by the method `cylindricalProj()` of the `PanoramicUtils` class. This is an important step: projecting the images on the inside wall of a cylinder bends the image making the stitch continuous. In particular we can observe that this projection bends the horizontal lines while keep straight the vertical ones. Since now the photos are now projected and we are moving on a cylinder the stitch can be made through a simple translation of the images along the horizontal axis. The computation is made in the `doStitch()` method. This method implement the following pipeline:

1. For all the images are obtained some keypoint thanks to the ORB feature detector, in the maximum quantity specified by `orbPoints`, and for every point is associated a descriptor. Is chosen the ORB instead of SIFT since the results with the provided dataset are comparable but the computation is faster in the first case.
2. For all couple of consecutive images are found the points that match through the descriptors thanks `BFmatch()`. Since the ORB is used the parameter for this matcher is `cv::NORM_HAMMING`.
3. To maintain only a small set of matching points an upper limit to the Hamming distance is selected by a `ratio` multiplied by the minimum distance found. In this case is chosen `ratio=3` since after a trial and error step it gives good results.
4. To the remaining good matching points are calculated the distances from the position of the points in the first image and the same point on the second one. These distances are used in a simple RANSAC method to calculate the translation horizontally and vertically. In this step the maximum number of iteration and the threshold can be chosen by the `maxRansacIter` and `thresholdRansac` parameters. The vertical translation is also taken into account since the values are small and don't give problems with the cylindrical projection previously performed.
5. After the copy, under and over some images, there are black bands due the vertical translations. To remove these bands from the resulting image is made a crop taking as bounds inferiorly by the highest vertical tranlastion and superiorly the size of a single image minus the lowest vertical translation. This gives us the largest rectangular are that contain only the images.

The panoramic image computed can be obtained through the `getResult()` method.
The results shown below are not optimal due the change of exposition between the single shots and the merge in some case are not perfect. Some improvement can be made like an histogram equalization before the merge or a merge between the image not made by a simple copy of the single data but a weighted mean based ond the distance from the center of the single shots.



(a) Planar picture



(b) Cylindrical projection

Figure 1: Cylindrical projection from a planar photo



(a) "Lab" set, 12 images, $\text{fov} = 66^\circ$



(b) "Kitchen" set, 23 images, $\text{fov} = 66^\circ$



(c) "Lab 19 Manual" set, 13 images, $\text{fov} = 66^\circ$



(d) "Lab 19 Automatic" set, 13 images, $\text{fov} = 66^\circ$



(e) "Dolomites" set, 23 images, $\text{fov} = 54^\circ$

Figure 2: Result obtained from the given photo set