

Code Annotations

Question 3: Region Descriptors

d) Which descriptor performs better and why?

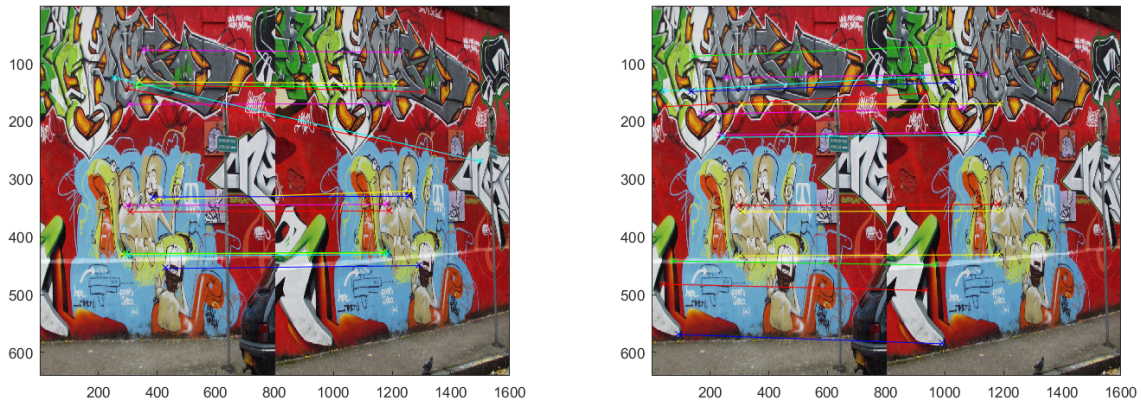


Abbildung 1: *Left*: Harris detector with rg histogram descriptor. *Right*: also Harris, but using the dx-dy histogram descriptor

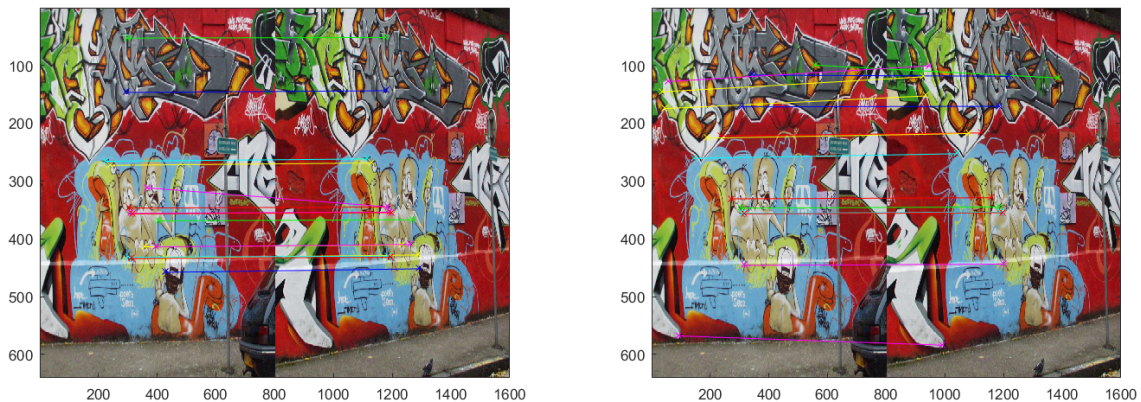


Abbildung 2: *Left*: Hessian detector with rg histogram descriptor. *Right*: also Hessian, but using the dx-dy histogram descriptor

If examined carefully, one can notice from the images in figures 1 and 2 that for both, Harris and Hessian detectors, the dx-dy-histogram descriptor seems to outperform the rg-histogram descriptor in this example, based on the amount of mismatched points

between the images. This may be due to two reasons: first, the images present very hard edges which virtually don't change if the angle of the image is changed. This makes it ideal for dx-dy as it can more easily distinguish two different sections of the image. Another factor could be that the image itself presents very similar colors in different parts. One such example is on the baby's faces in figure 2 left, where, even though the matched points are close to each other, they do not correspond to the right match. There are of course other more extreme examples where points near some of the letters get matched to points in completely different locations. It is important to note, never the less, that this does not mean that using dx-dy-histograms is better than using rg-histograms, but is instead domain dependent, and even if dx-dy beats rg-histograms in this image, there will be others where it will not be so.

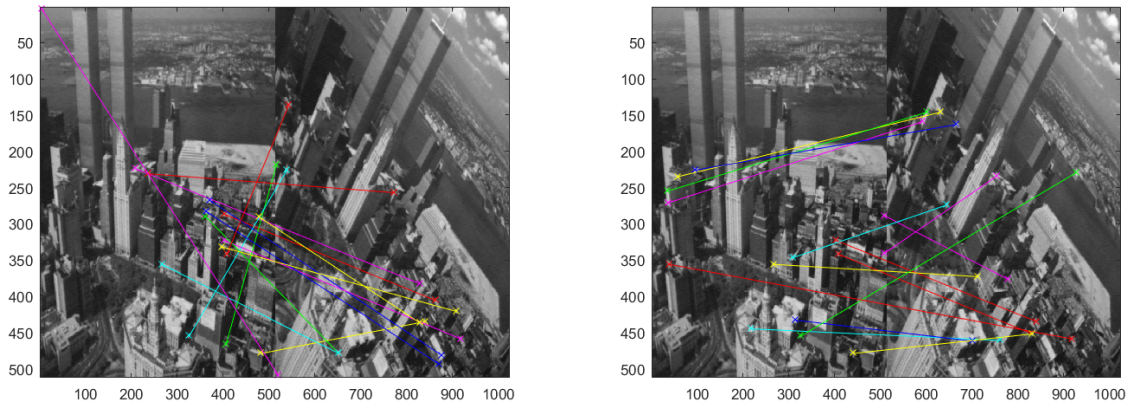


Abbildung 3: *Left*: Hessian detector with dx-dy histogram descriptor. *Right*: also Hessian, but using the maglap histogram descriptor

In the case of the New York images, one can see a few more inconsistencies in the case of the dx-dy-histogram compared to the graffiti images. The main cause for this could be the quality of the image, by being not very high the image becomes a little blurry and pixelated, which in effect, makes the the corners and edges less distinguishable. Another contributing reason could be because it is an image from a city (3D) no of a graffiti (2D) which means changing the angle of view has a bigger impact on the image and therefore the first derivative, which is what the dx-dy-histogram uses. As for the maglap-histogram, one can see less inconsistencies than the dx-dy-histogram. It is no surprise that it performs better as it also uses the first derivative, in part, but also the second derivative. So in a way, it is more informed than just the dx-dy histogram and can, thus, make better matches. As opposed to the comparison of rg-histograms vs dx-dy-histograms one would expect that in general maglap-histograms outperform dx-dy-histograms.

Question 4: Panorama Stitching

a) Homography estimation with RANSAC.

- **What does homography estimation do?**

Homography is the projective transformation between two images of the same planar surface, i.e. it maps the points of one image to the points of another image taken by a camera which was projectively transformed from the first one. To estimate the homography of two images a set of pairs of points is required, which represent projections of the same world point in each image.

In the case of `get_ransac_hom.m`, it tries to compute the homography as a transformation matrix that transforms one given set of points to another. In practise a homography can usually not be computed exactly due to noise in the data. Therefore it has to be estimated, for which it is important to 'ignore' outlier data (possible errors in data collection, ...) to preserve a realistic and precise estimation.

- **What are the steps of RANSAC algorithm and why it is useful?**

- 1) We compute a random sample of Data-values
- 2) We compute the Homography of these randomly selected values
- 3) We check if this random sample contains more inliers compared to the best random-sample we computed earlier, if that is the case we save our new random sample as the new best random-sample.
- 4) There is a statistic method which estimates the number of necessary ransac iterations using the ratio of inliers and outliers. With every iteration we can improve our estimation of necessary iterations, thats why we update this number every time
- 5) Once all iterations are finished, we are going to improve the best sample we found:
 - Every point inside the threshold around the computed Homography of the best sample found, is used as input set of which the homography is computed again, to enlarge the subset on which our homography is based on. This is done exactly once in our implementation.
- 6) Now we have succesfully computed the resulting Homography.