# Computer Vision
# Lab 5: Panoramic Images

Riccardo De Zen. 2019295

17 May 2021

## 1 Introduction

This report covers the fifth lab experience, regarding feature matching for panoramic images.

## 2 Implementation

I organized my `PanoramicImage` class to hold a copy of the original images and of the projections on the virtual cylinder. While the example result in the assignment gave a black and white image, I decided it would be more similar to a real-life application if I kept the images in BGR format for the result (I modified `cylindricalProj()` to keep all channels). The SIFT and ORB features are still computed on the grayscale images. In the end I realized that histogram equalization sometimes helps the grayscale results but generally worsens BGR results (Figure 3), hence why I decided to keep both options available. In total, four options are available on a generic `PanoramicImage`:

- **BGR**
- **Equalized BGR**
- **Grayscale**
- **Equalized Grayscale**

Everything is computed lazily and as few times as possible: the images are projected only once, and so are the shifts to reconstruct the final images. The equalized projections are computed only if an equalized version of the result is requested. I also give the possibility to retain the drawings of the matches after trimming them with RANSAC, for debugging purposes. These are also computed only the first time they are requested, since the matches only need to be computed once. The choice between SIFT and ORB is made by instantiating the appropriate subclass.

Most given images have some distortion at the sides after the cylindrical projection. Since I initially pasted the images left to right with the rightmost image standing over the leftmost one, the visible portion around junctions was coming entirely from the right image. To solve this issue I decided to meet the images halfway trhough the shifts. There is a clear increase in quality (example in Figure 1).

When I read the suggestions in the live lab lecture about not considering vertical shift, I had already implemented such feature, so I left it in. After improving the junctions as described above, I wanted to take the procedure one step further and perform linear interpolation over a small portion near the junction (roughly 20%). A comparison is shown in Figure 1. The vertical shift made it a bit harder for me to implement this, but I had not yet taken it into account when adding the vertical shift itself.

As a side note, the class also allows images to be provided right to left. I simply reverse the image vector and still draw left to right.

## 3 Results and discussion

Very little can be measured here. Results are mostly qualitative. Most datasets can be used without issue, but some had trouble. Tweaking the distance ratio (only matches with a distance below `dist_ratio` times the minimum are kept) and some ORB parameters allowed to get good results with all datasets.

- **SIFT**: most datasets perform appropriately already with a distance ratio of 3, except for `dataset_lab_19_manual`, as can be seen in Figure 4. Since I assumed having more features would help RANSAC erase outliers, I simply raised the distance ratio. Setting it to above 10 solved the issue.
- **ORB**: to reach the bare minimum of 4 matches for computing the homography, the distance ratio had to be increased to 10 or 15. This was not enough to reach a good result on multiple datasets (e.g. `dataset_lab_19_automatic` in Figure 4). To solve the issue, it was necessary to increase the number of features retained by the detector from 500 (default) to 2000. Only `...automatic` required the number of features to be further raised to 5000. The dataset `lab` will not work with more than 500 features unless we enforce the minimum match distance to be $\geq 1$ because a match with 0 distance is found.

The best results obtained were similar with either feature detector. As shown in Figure 2 my linear interpolation fooled me into believing results were better than they actually were, so for the sake of showing the results in this report, it has been disabled (I decided not to add another parameter for it and leave it enabled by default in the code).

Figure 1: Comparison of default junction (left), middle junction (middle), and linear interpolation (right). Clean cut when meeting in the middle, but linear interpolation smooths the color change. This ends up blurring the shoulder of the student on the right, so the actual final quality is debatable.

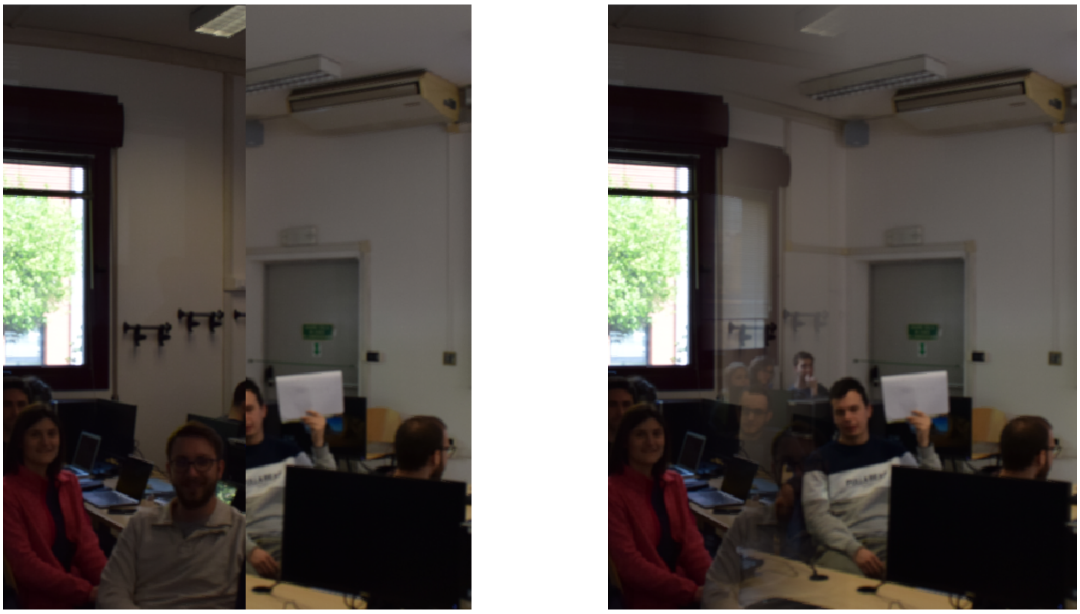


Figure 2: The bad ORB result (2000 features) to the left was harder to notice on the right due to linear interpolation.

Figure 3: Equalization on BGR images is not always desirable.

(a)

(b)

(c)

(d)

Figure 4: Examples of bad results:
(4a) SIFT bad result on **`dataset_lab_19_manual`**.
(4b) ORB (500 features) bad result on **`dataset_lab_19_automatic`**.
(4c) ORB (2000 features) on **`dataset_lab_19_automatic`**.
(4d) Whenever the result is quite bad the reason is a heavy fraction of matches are outliers like these.

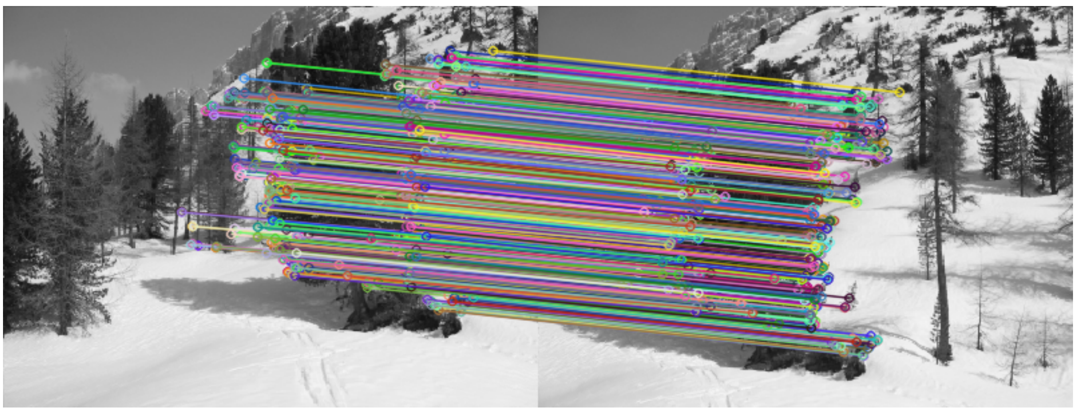
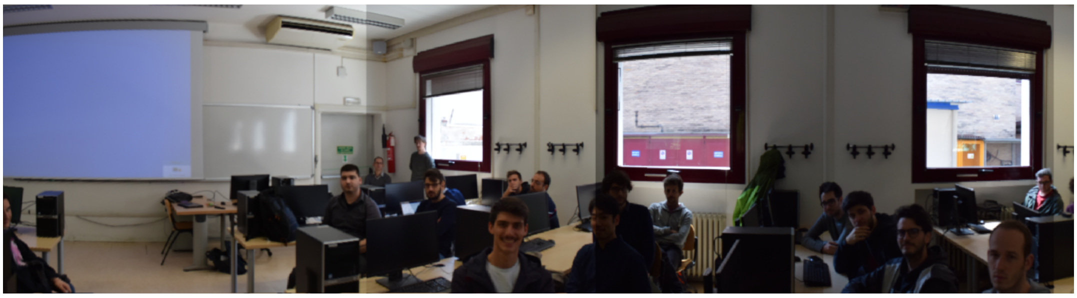


Figure 5: Example of good matches (ORB on **`dolomites`** dataset).

(a)



(b)

(c)

(d)

(e)

Figure 6: Example final results with SIFT (cut to one third of the width for a better view). No equalization nor linear interpolation.