CV 2021: Final Projects

Panoramic image construction with feature descriptors (LAB4-a)

Student: Vuk Ilic

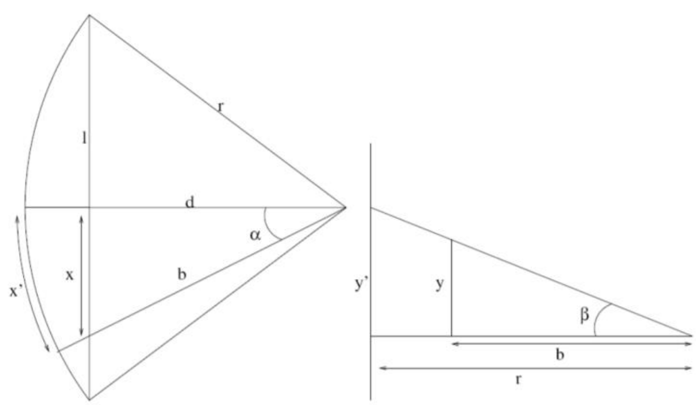
Professor: Zanuttigh Pietro

Panoramic images

Panoramic images cover a 360° field of view in the horizontal direction. They can be built from a set of pictures taken with a rotating camera from a single viewpoint. Panoramic photography is widely used on today's smartphones and cameras. The process of joining the various images together is called stitching.

Cylindrical Projection

Various projection methods can be used in order to represent a panoramic image. The most common solution is to project the image inside a cylindrical surface. In order to build a cylindrical panorama we start from taking a set of pictures from the same viewpoint by rotating the camera in the horizontal direction with a constant rotation step. The rotation step must be smaller than the area covered by a picture in order to have the images partially superimposed. This allows to join them, e.g., by locating common features (after the projection each couple of pictures is related by a simple translation). Each picture is then projected on the cylinder and covers a particular angular sector of it.



If we consider that (x, y) are original photo coordinates, coordinates on the cylinder (x', y') can be calculated by the following formulas:

$$x' = r \, arctan\left(\frac{x}{d}\right)$$

$$y' = y\frac{r}{d}cos\left(\frac{x'}{r}\right)$$

Algorithm

The algorithm consists of several parts:

1. At the beginning we have to project the images on the cylinder. As we are doing with color images, we need to apply the cylindrical projection on each of B, G and R channels and then merge them into one image.
2. Next we have to extract the descriptors for each image. Both ORB and SIFT are implemented in the code.
3. Find the matching features.
4. Estimate the translation between couples of adjacent images starting from the matches using a robust estimator (RANSAC)
5. Build the panoramic image by stitching the adjacent images. Cylindrical projection makes the image superposition much easier since the relation between the various images on the cylinder is a simple translation. It's important to mention that stitching is applied on color images, while the feature matching is applied on their grayscale version.
6. Final step is to visualize the panoramic image.

Implementation

The project is implemented in Microsoft Visual Studio 2017 in C++ programming language. The main library for project realization is OpenCV which is mainly aimed at computer vision tasks.

The project is provided in two files: "*panoramic_image.cpp*" and "*panoramic_utils.h*".

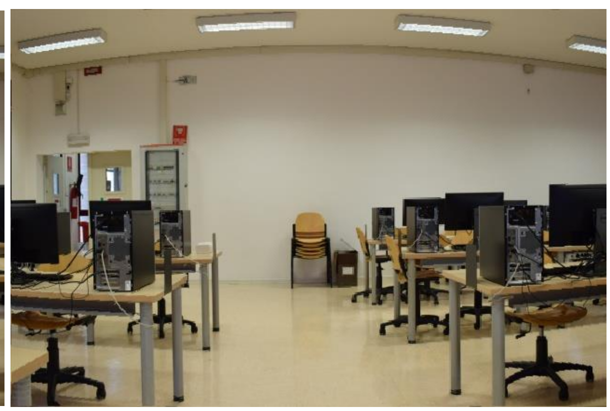
In the main() function we first choose the folder from which we want to import images. Then we define an angle value (in degrees) which is half of the FoV of the camera used to take the photos. The FoV of the camera is 66° (half FoV=33°) for all the provided datasets excluding the *"dolomites"* one for which it is 54° (half FoV=27°). After that we apply cylindrical projection on each image, but as we are working with the color images we have to apply it on each channel. This is done by function *PanoramicUtils::cylindricalProj(channel, angle).* Translation estimation between two adjacent images is applied in one of the two functions, *panorama_orb(img1, img2)* or *panorama_sift(img1, img2)* based on the desired feature extraction method. Panoramic image is finally created by horizontal concatenation of current panoramic image and extracted piece of new image by calling a function cv::*hconcat()*.

Functions *panorama_sift()* and *panorama_orb()* are working very similar. The main difference is the applied feature extraction algorithm: SIFT in first and ORB in second case. Another difference is the matching algorithm: FLANN based matcher with NORM_L2 in the first case and Brute Force matcher with NORM_HAMMING in the second case. In both functions we are selecting good matches with different approaches which are explained in detail in the code. We localize the overlapping part of 2 adjacent images by using a function *cv::findHomography()* with *method=RANSAC.* Because of the cylindrical projection relation between 2 adjacent images is a simple translation. That's why we only need parameter on the position (0, 2) of the homography matrix - horizontal translation. This estimated translation is finally used to extract the part of the second image that is not in the first image.

Results

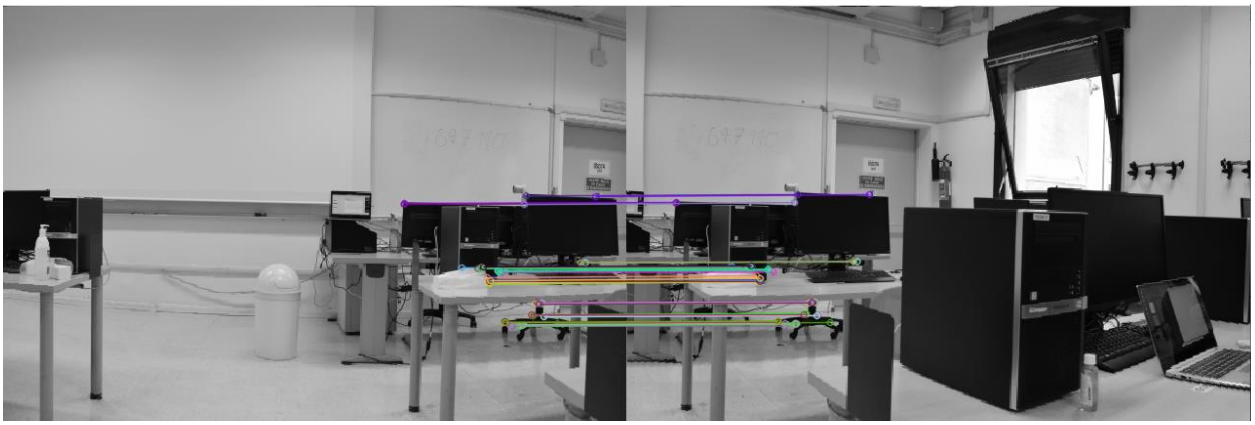
In this section we will go through the most important parts of the project and show some results and at the end we will see some final results on different sets of images.

First we will see an example of cylindric projection:

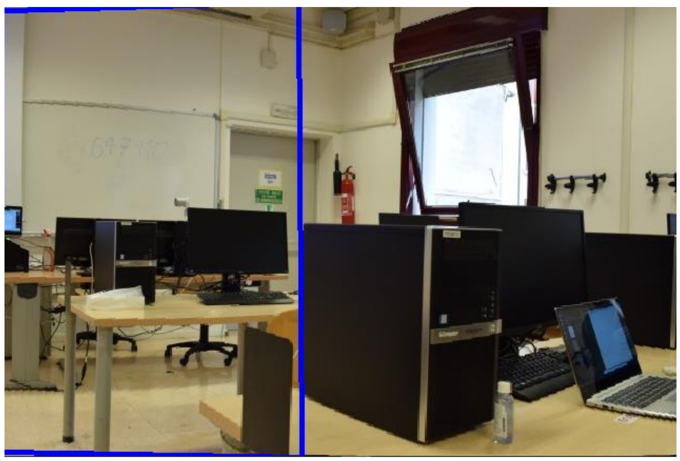



Next we will see some results of the feature detection:

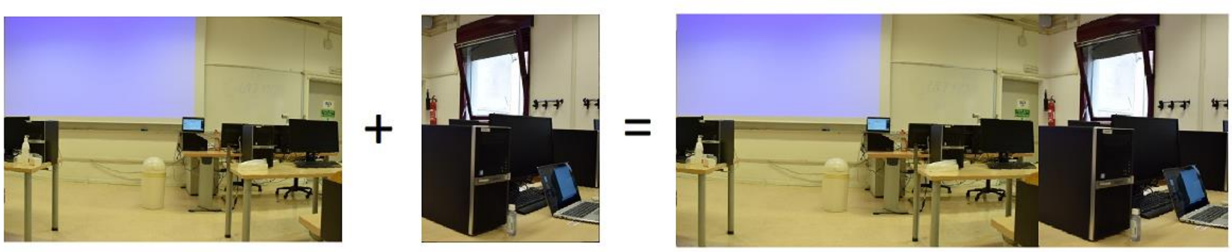

After that we did a feature matching:



Then we applied *cv::findHomography()* to estimate the translation between the images:



The blue rectangle represent the overlapping parts of two images. As we can see that rotation of blue rectangle is very small, that mean that we did a good job with applying cylindrical projection and now we can use simple translation for image stitching:



After multiple stichings we finally get the resulting panoramic image:

Some results on different sets of images:

As we can see the results on *lab* and *kitchen* image sets are very good, but in case of *dolomites* image set we have some artifacts because the feature matching task is much more challenging.

In case we were using ORB features instead of SIFT, we got the following results for the previous examples:

We can see that the result on *kitchen* image set is same as in SIFT case, but in the case of *lab* image set the results are worse than with SIFT.