Computer Vision VU

Exercise Course: Assignments II

WS 2015/16

Harald Scheidl, 0725084
Thomas Pinetz, 1227026
Velitchko Filipov, 0726328

Assignment 4: Image Stitching

Overview

This assignment is separated into 2 parts.

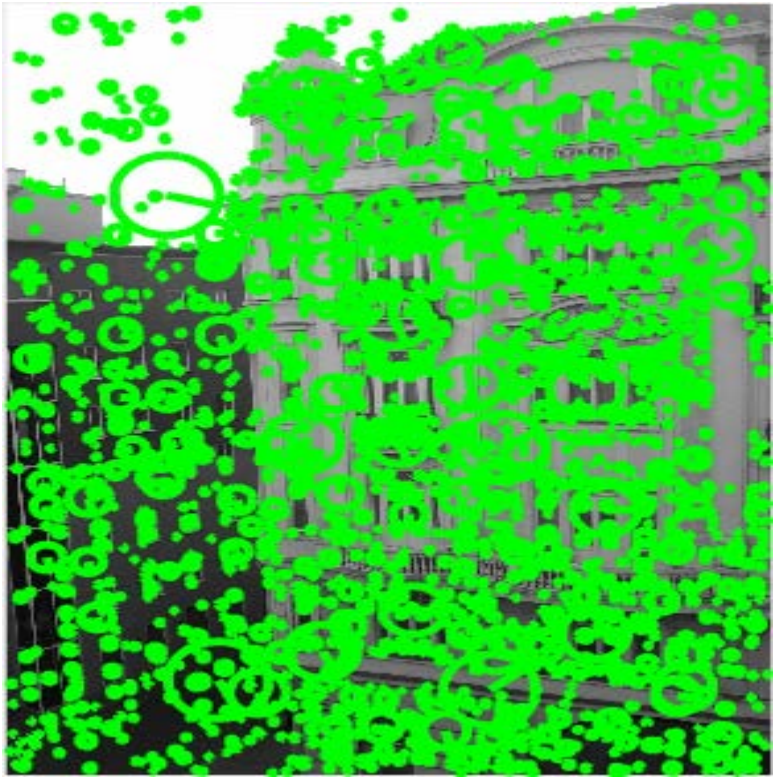
1. Image Recognition → RANSAC Scheme
2. Image Recognition → Image Stitching

In the first part, we need to transform one image onto another one with as little difference as possible. For this, we use SIFT to compute the features of both images. Then we match them with a brute force matcher. From those matches we then calculate the geometry of the image by using the RANSAC Scheme and overlay the first with the second image. In the second part we use SIFT features to stitch multiple images together.

Questions

PART A:

1.) Show the output of vl_plotframe on the chosen image. What is the meaning of the size of the drawn circles and lines inside the circles?



SIFT uses image pyramids to find features at various scales. The size of the circle in the image corresponds to the level of the pyramid in which the feature is present. The lines inside the circles describe the orientation of the gradient of the each feature.

PART B:

1.) Plot the matches after step 2. Describe in detail what vl_ubcmatch does.

As input for the Matlab function vl_ubcmatch, we provide a set of descriptors from image1 and image2 (computed from vl_sift). The function returns the matches of the descriptors of our points of interest.

2.) Plot the matches of the inliers after step 4. What is the difference to the set of all putative matches we plotted before?

When plotting the matches resulting from vl_ubcmatch, we can see that there are a lot of false matches. We need to improve the set of matches so that we have no 'outliers' present in the result. For this we use the RANSAC algorithm, which in turn does the following:

1. Pick 4 random points from the matches
2. Estimate the homography of these points using the Matlab cp2tform function
3. Transform all points of the matches in the first image using tformfwd
4. Compute the Euclidean distance between the transformed points in the first image and the points in the second image. Drop all points with a distance larger than 5 (in our case the threshold is T = 5 from the task description)

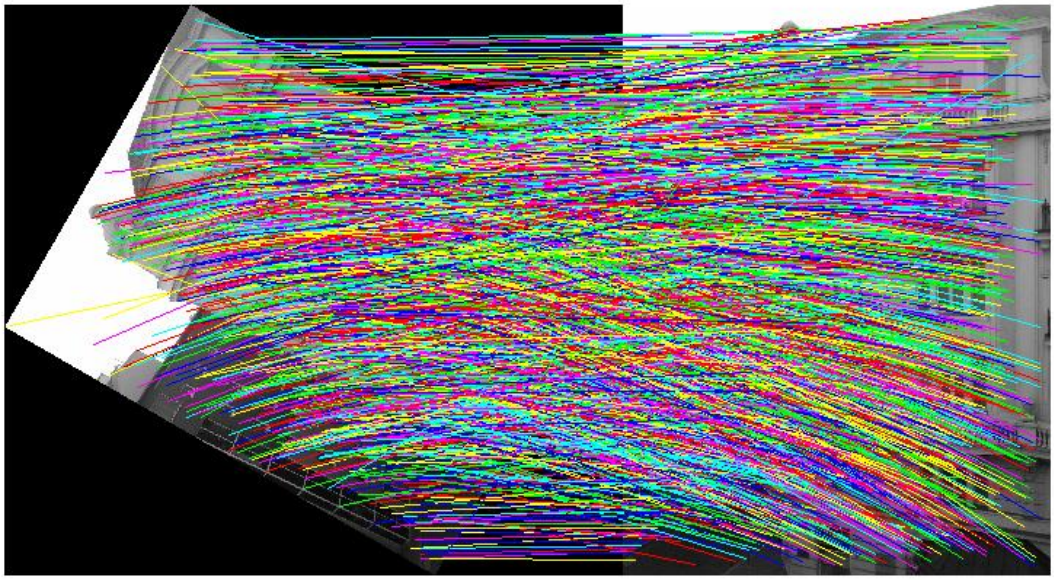
The result of this is we have less matches, but the matches that we have are correct.

3.) Apart from small errors, the images should be aligned after step 5. Demonstrate that by showing the absolute differences between the two images.

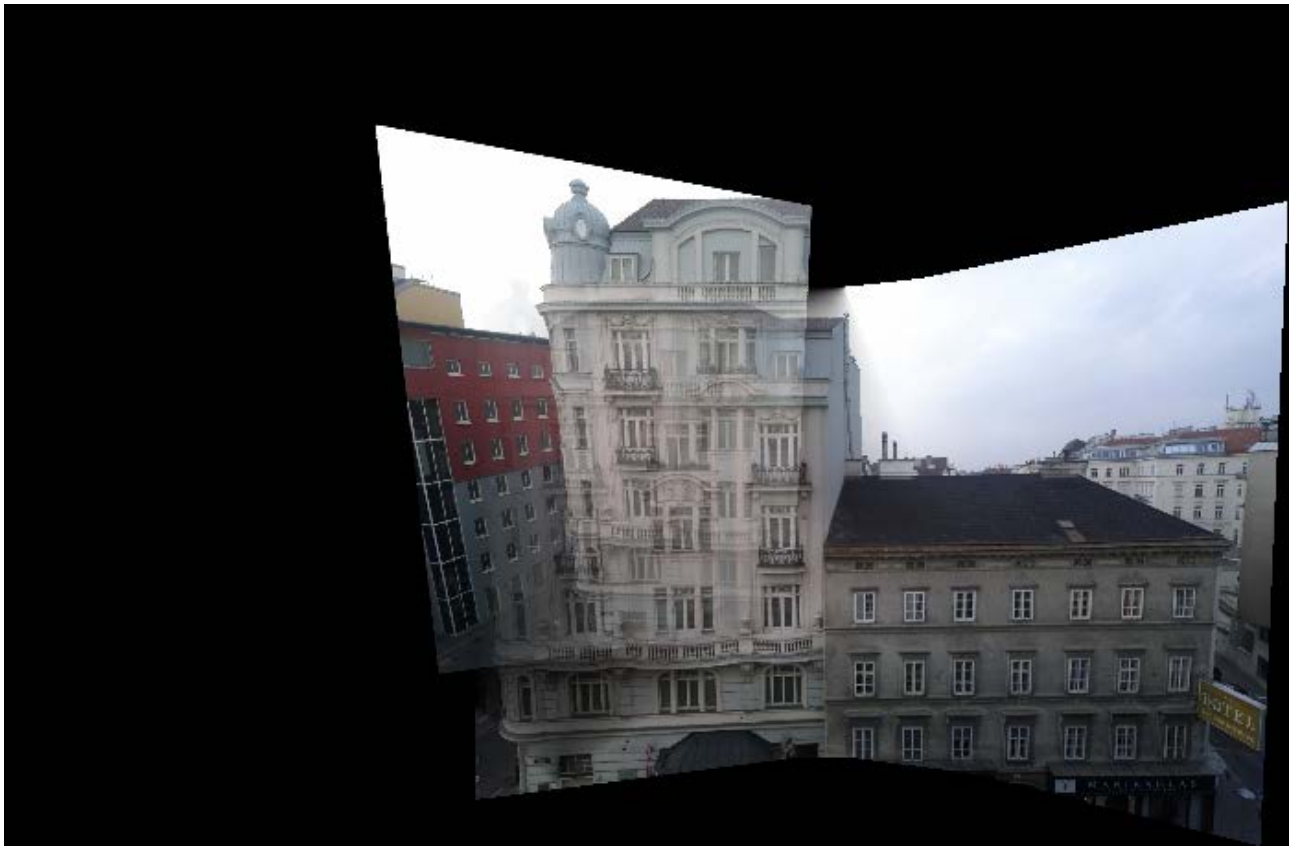
The left image was transformed and plotted on top of the right image to visualize the difference between the images.

4.) Examine if the presented scheme of SIFT interest point detection and RANSAC-based alignment is invariant to changes in image rotation and scale. Thus, likewise to assignment 3, resize and rotate the second image and repeat the alignment procedure.

Since SIFT is invariant to scaling, when scaling the image the matches remain. In our assignment if there is no change in the viewpoint from which the images are captured SIFT also seems to be invariant to rotations applied to the images. The matches between the original image and the rotated (by 60 degrees) and scaled (by a factor of 2) image are displayed in the figure below. (Note: this is before RANSAC, there are outliers in the image). The SIFT algorithm seems to find matches between the rotated and scaled image and the original image.



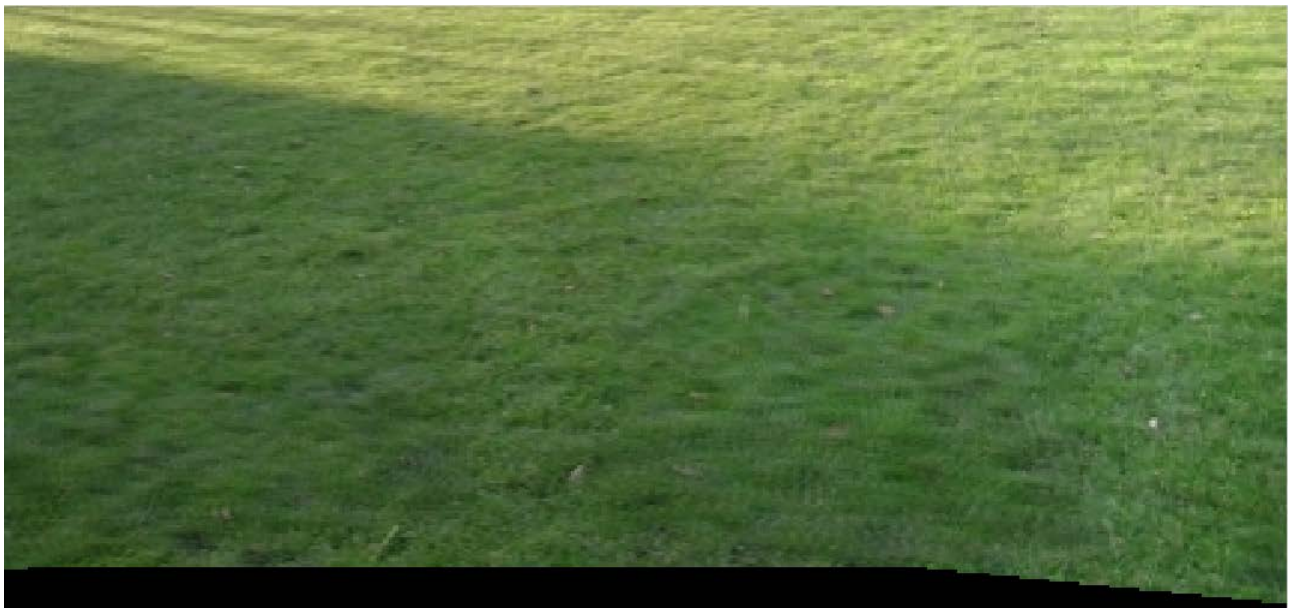
The panorama stitching seems to be incorrect when using the rotated and scaled image. Due to lack of time, we could not improve upon this result.



PART C:

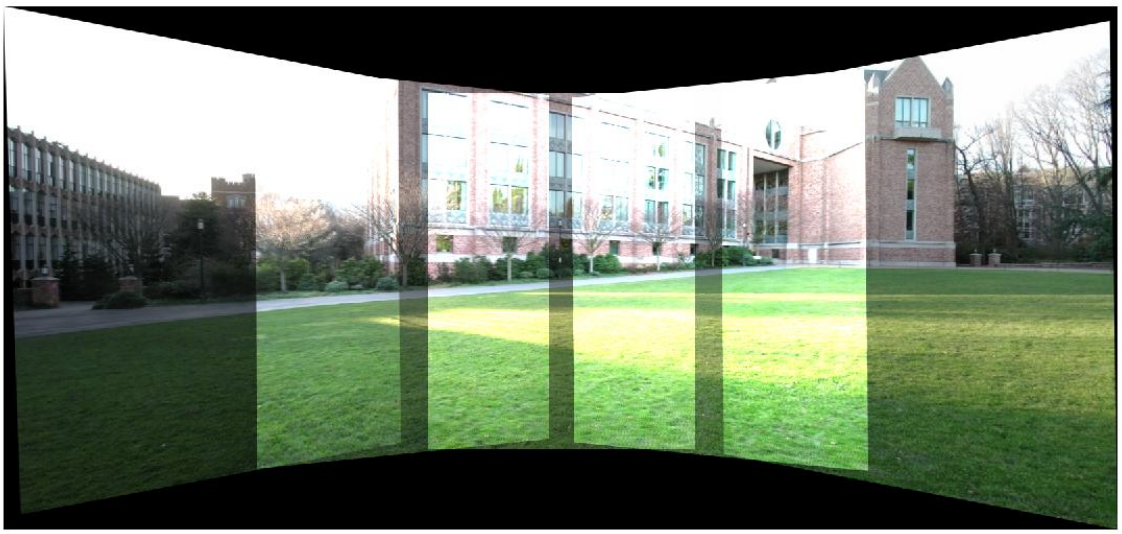
1.) Show and discuss the achieved results (with the two provided image sequences and your own sequence). The result might look quite realistic at a first glance but can you spot any errors by looking on details?

The results of the panorama stitching of officeview{1-5}.jpg and campus{1-5}.jpg are displayed below. Upon detailed inspection, it seems that some parts of the image are not perfectly aligned. This is visible in the campus panorama when zooming in on the grass region in the middle of the result.



2.) Compare a result achieved with feathering to a result where no blending has been performed (i.e. the color values of only one image are taken for the stitched image). What is the difference of the two results?

When no blending is performed and just the colors of the original images are accumulated, the result is overexposure (due to the accumulation or RGB values) in the overlapping parts of the image. The blending compensates for this and achieves correct colors in the overlapping regions of the image.



Assignment 5: Scene Recognition with Bag of Visual Words

Overview

First, let us have a short look at how this classification method works.

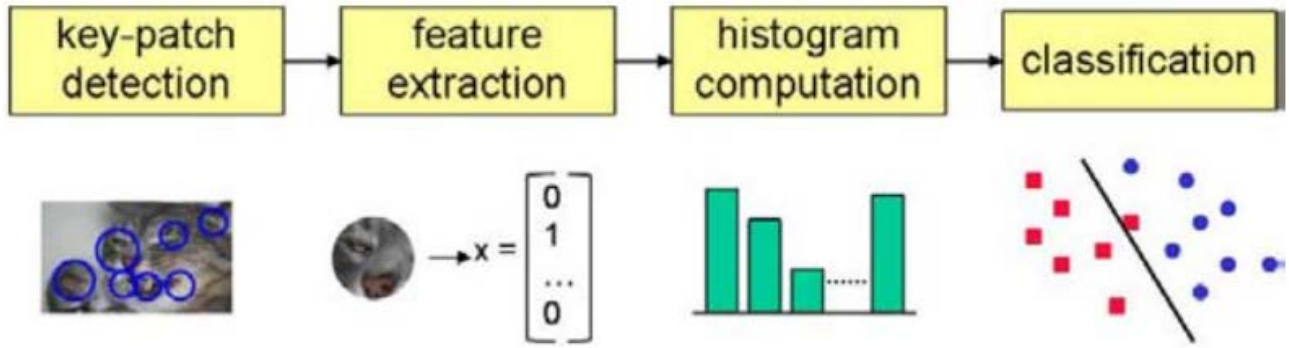


Fig: Overview of the classification step [Szeliski]

- **Build a vocabulary of words**: a huge set of SIFT descriptors of all training images are collected and then clustered into 50 clusters in the 128-dim SIFT space. The centroids of those clusters are called visual words and are nothing but elements of a 128-dim vector space.
- **Build training set**: now SIFT descriptors are calculated once more for the training images, and for each SIFT descriptor, the nearest visual word is searched by the KNN algorithm. For each image, a distribution (histogram) over the set of all visual words is calculated. This distribution is normalized to sum up to one. For each image, the distribution and the group label (e.g. “street”) are saved. One can again think of these distributions as elements of a 50-dim vector space.
- **Classify**: for a new image, a number of SIFT descriptors is calculated and then assigned to the nearest visual word. Again, the distribution over all words is calculated. The KNN algorithm is used once more for classification, this time in the 50-dim space of distributions. The three (k=3) nearest elements of the training set are searched in this space and the class is determined via a majority voting.

Discussion of results

To measure the quality of this method, we use a confusion matrix C. Entries at the diagonal Cii represent the number of correct classified images with class label i.

Wrong classified images can be found on all non-diagonal entries Cij, i≠j: such entries represent the number of instances of (real) class i which are (wrong) classified as class j.

We added two visualizations of the confusion matrix to this report. In one visualization the entries are colour-coded, in the other visualization the numeric values can be found.

In both visualizations, rows represent the real classes, while the columns represent the classified classes.

The class labels are values between 1 and 8. The mapping between the numeric value of the classes and the names of the classes can be found in this list:

- 1 = bedroom
- 2 = forest
- 3 = kitchen
- 4 = livingroom
- 5 = mountain
- 6 = office
- 7 = store
- 8 = street

To get the percentage of correct classified images, we have to sum up all diagonal entries of C and divide this sum by the sum of all entries of C. This gives a value of 473/800=0.5913, which is about 60% as supposed in the assignment.

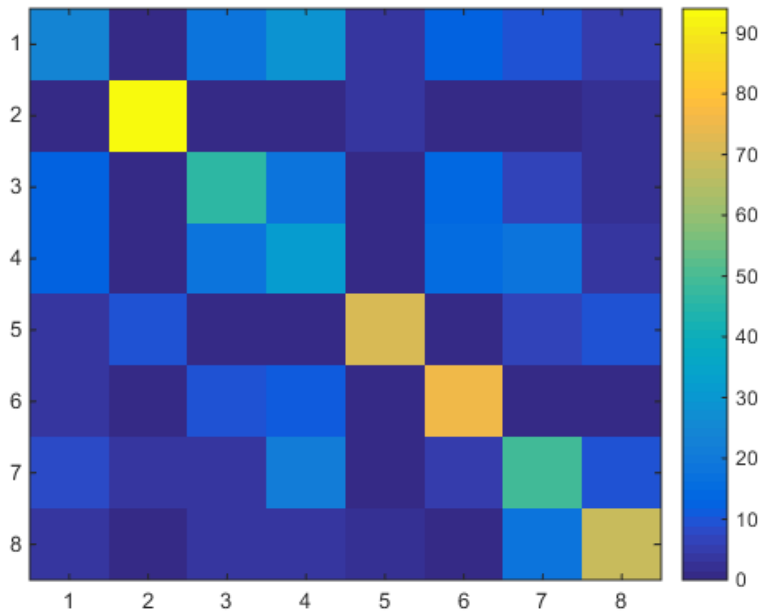


Fig: confusion matrix, colour-coded

	1	2	3	4	5	6	7	8
1	26	0	14	30	2	14	9	5
2	0	94	0	0	4	0	0	2
3	10	0	44	20	1	15	8	2
4	11	0	19	38	0	15	14	3
5	3	10	0	0	71	0	4	12
6	4	0	13	16	0	67	0	0
7	4	5	4	18	1	4	55	9
8	0	1	2	4	2	2	11	78

Fig: confusion matrix, numeric values

When comparing the numeric class labels with the class names, one can see that the classification of forests works pretty well. In addition, other “natural” (outdoor) scenes have a good classification rate.

Problems occur when classifying categories such as bedroom, kitchen or store. They appear to be similar for this method, i.e. with respect to visual words.

Own images

We used three own images of three different categories. As one would expect after looking at the confusion matrix, there was no problem identifying the forest. Also the image of the mountain was classified correct.

The image of the street however was classified as a living room. Looking at the confusion matrix, one would expect a classification rate of about 78%. Inspecting the training images of the category street shows the reason for the incorrect classification: the training images represent streets in urban areas, while our own image shows the Overseas Highway which passes through non-urban areas.

Fig: forest (group=2), classified as forest (group=2)

Fig: mountain (group=5), classified as mountain (group=5)

Fig: street (group=8), classified as living room (group=4)