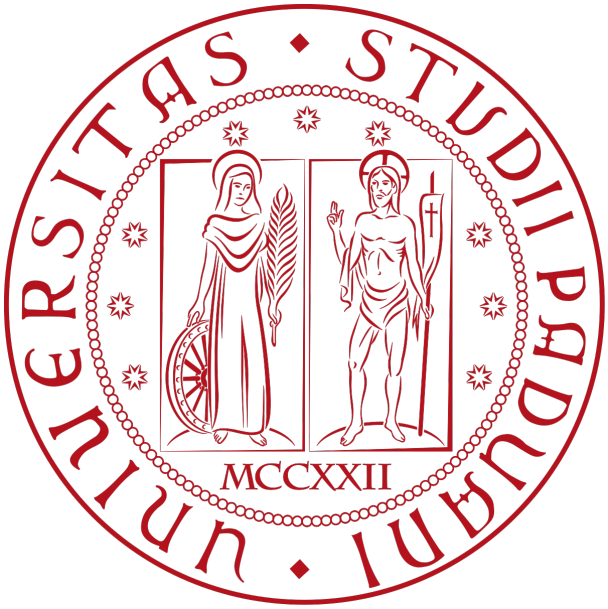

Computer Vision Project Report

Scapinello Michele (2087617)

Department of Information Engineering
University of Padova

Summary

The project developed targets the problem of image completion and uses features extractors (SIFT) and matching algorithm (BFMatcher) to try to solve it. The problem goal is to complete the missing pieces of a corrupted image with a set of patches and to make the problem more challenging also a set of patches with some modification have been given. The final results are positive for the non modified patches, whereas with the modified ones in some occasion the program is able to correctly overlay the images but in others it is not. Also a manual implementation for the RANSAC algorithm, used to identify the position for the overlay procedure, is developed.

Contents

List of Figures

1 Completion with unmodified patches

To accomplish the goal a simple algorithm has been developed as shown in 1. After loading the images and the unmodified patches, SIFT is computed on both to extract the key points and with the obtained results the matching between the key points descriptors is computed by using the `knnMatch()` method of the `BFMatcher` class.

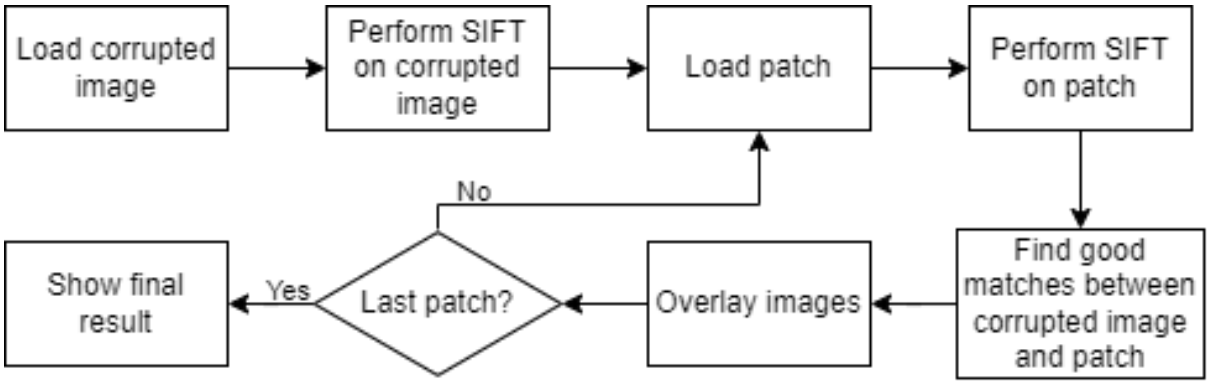

Figure 1: Flow diagram of image completion algorithm for the unmodified patches

The decision on whether a match is classified as good or discarded is made through a ratio test that basically discards all the matches that have a distance greater than the minimum distance found between the matches, multiplied by a ratio equal to 3 by default. After extracting the consistent matches the patch is overlaid over the corrupted image in the correct zone. This is done with the implementation of the RANSAC algorithm, explained in Ch. 3, that allows to find the transformation matrix which will be used to overlay the patch to the corrupted image (the overlap is done with the `warpPerspective` method). All these operation are performed until all the patches are analyzed and at the end the final results are shown. For the unmodified patches, all the images are correctly reconstructed without errors.

1.1 Results and Comments

The matches between the images key points descriptor and the patches descriptor is consistent and allows a correct overlay of the patches as shown in Fig. 2. The overlay of the patches is done with the implementation of the RANSAC algorithm, used to estimate the homography matrix and explained in 3.

Figure 2: Final result with unmodified patches (Scrovegni)

2 Completion with modified patches

The modified patches made the completion goal more challenging and thus required a more sophisticated algorithm, shown in flow chart through 3. Here, the decision on whether a match is classified as good or discarded is different. In this case we perform a comparison between the ratio of the distance of the 2 closest key points found for a match through the `knnMatch()` method. The match is classified as good if this ratio is lower than 0.8. If the number of consistent matches found is lower than 20 (parameter found by testing the algorithm) a flip of the patch is performed and the algorithm restarts. If after flipping the patch in all possible ways the number of good matches is still lower than 20, the patch is resized and the iterations repeated.

When the number of good matches is big enough, the homography matrix is estimated and after computing it, the patch is overlaid over the specific corrupted region. This procedure is done for all the patches.

2.1 Results and Comments

As shown in the results, the implemented algorithm works correctly in some scenario while in others it does not, Fig. 5. This might be due to the amount and type of modifications carried out on the patches, that makes it harder to compute and find the key points through SIFT and, consequently, makes it harder to match the key points of the patch and the corrupted image key points. The RANSAC estimation algorithm then computes a wrong homography matrix and the overlaid result is not correct. It is clear that the resize of the patches (performed with Lanczos for better quality) is not always a good idea, since the quality of the latter may deteriorate, thus finding wrong matches between the patch and the corrupted image. The threshold for the number of good matches might be adapted as well depending on the study scenario.

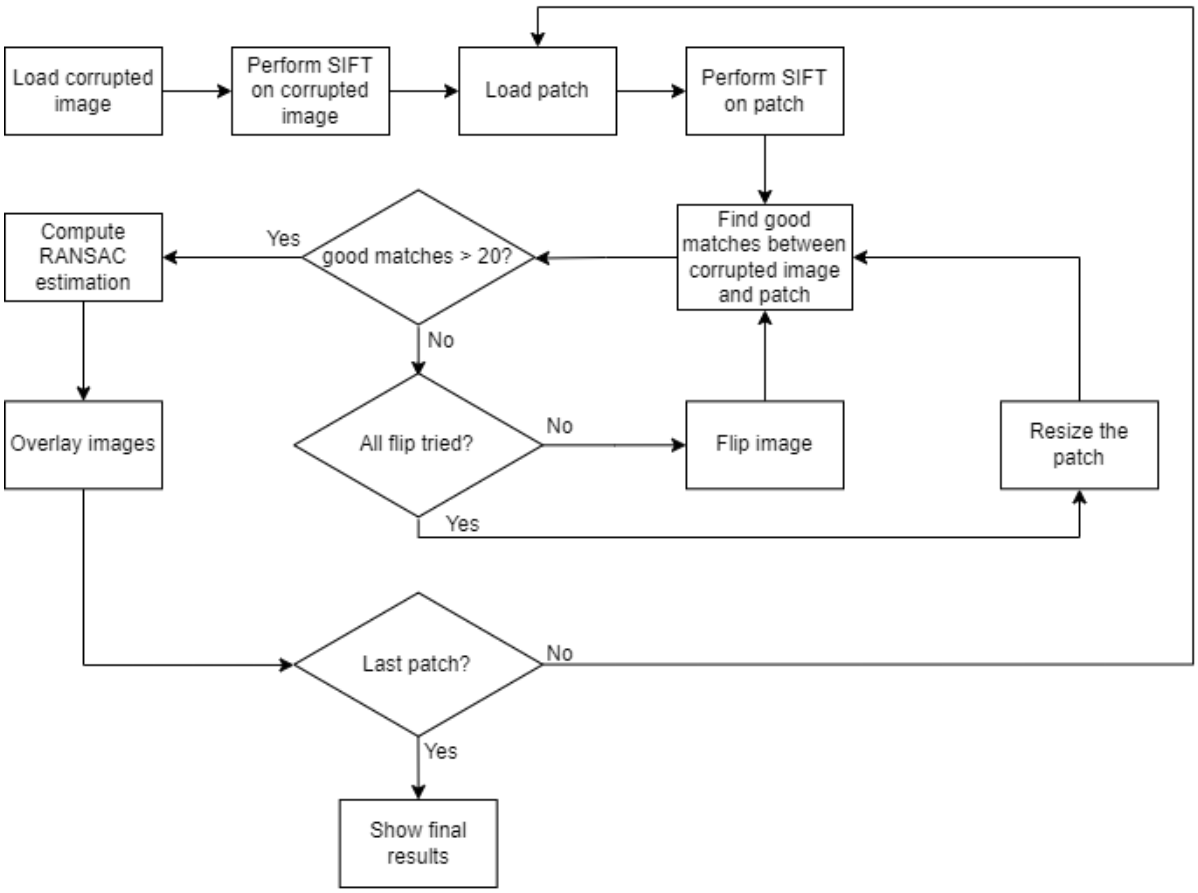

Figure 3: Flow diagram of image completion algorithm for the modified patches

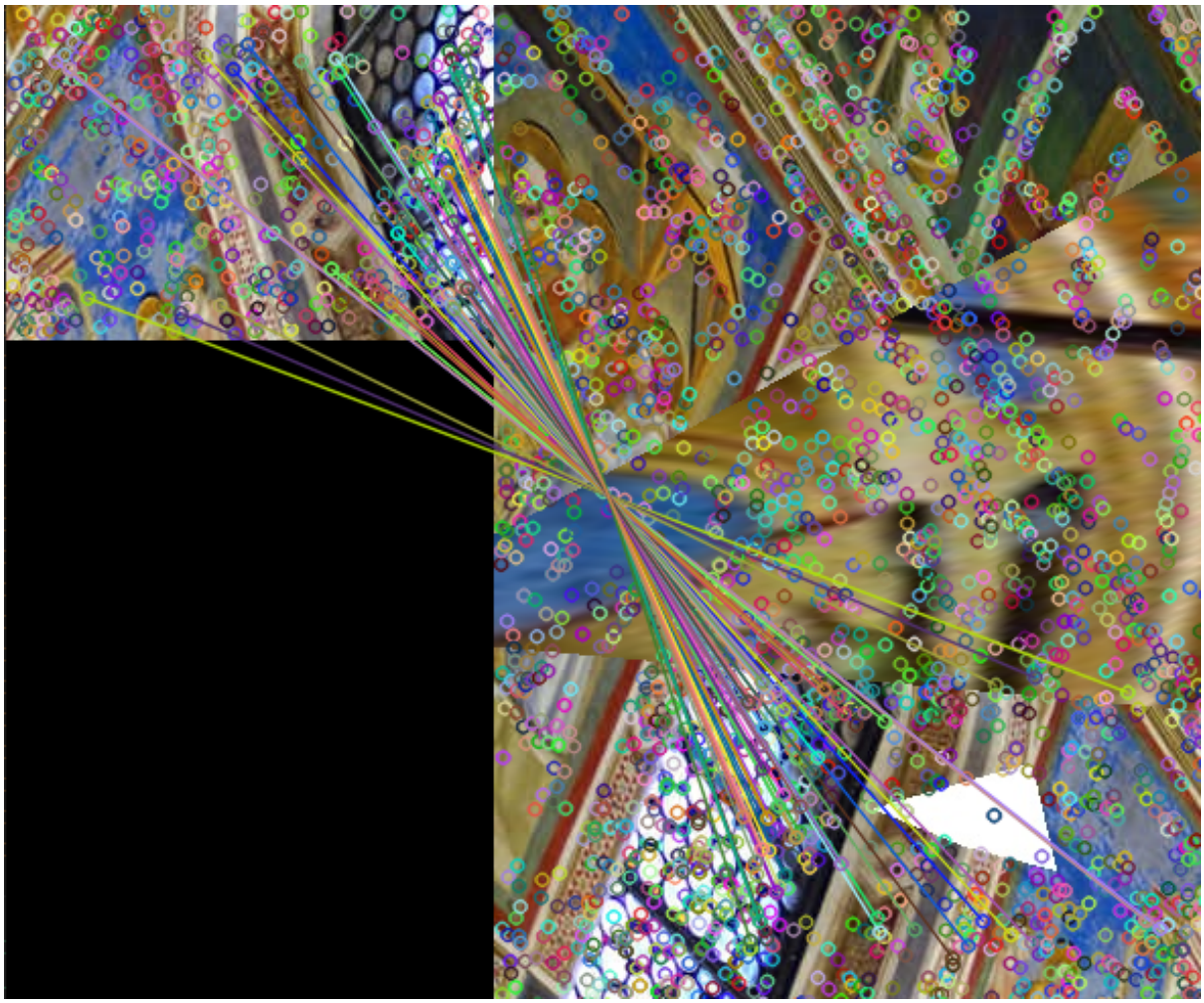

Figure 4: Matches between modified patch and corrupted image (Scrovegni)

Figure 5: Final result with modified patches (Scrovegni)

3 RANSAC implementation

RANdom SAmple Consenseous (RANSAC), is an algorithm used in computer vision to compute the homography matrix. The provided implementation works as follows:

1. Randomly extract a set of 3 correspondences between the set of patch descriptors and the set of the corrupted image descriptors;
2. Find the transformation matrix. This is done by considering the equation $Ax = b$, where A is the matrix with the patch coordinates, b is a vector containing the respective corrupted image coordinates, and x is the transformation matrix. To compute x:

$$x = bA^{-1}$$

3. Count the number of consistent correspondences between the set of good matches and the extracted ones. Compute the error between the estimated u, v values found with the transformation matrix, with the real u, v values. Compare the results with a threshold. In the algorithm:

$$|\Delta u| + |\Delta v| < T \tag{1}$$

   where T is equal to 3;
4. Repeat the steps for n times and keep the estimated homography matrix with the highest number of consistent matches.

The implementation of the algorithm is also shown via flow chart diagram in Fig. 6.

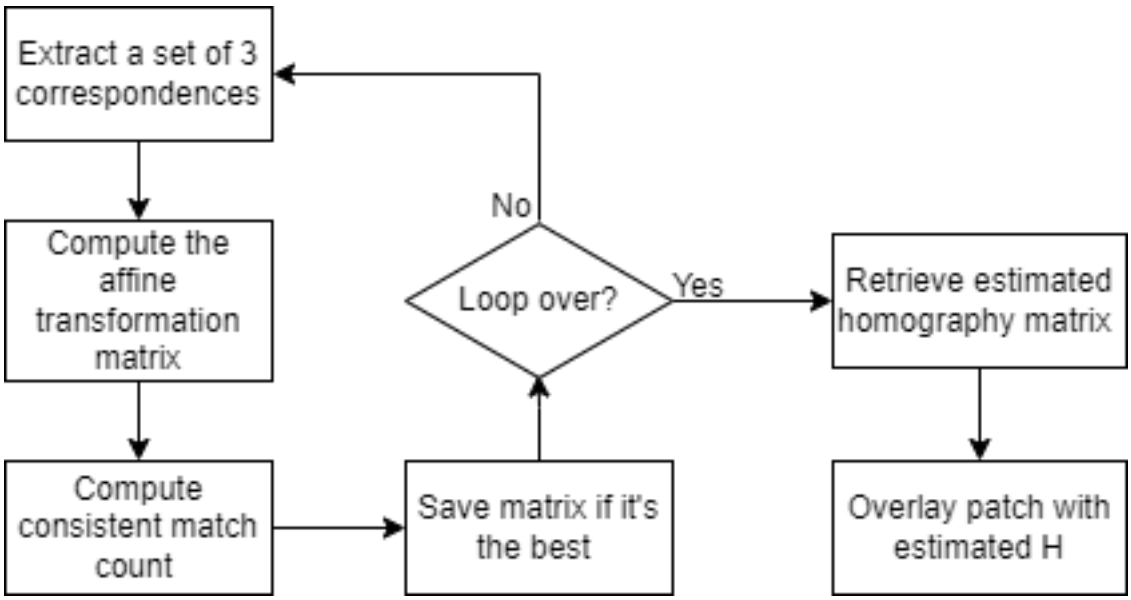

Figure 6: Flow diagram for the RANSAC algorithm implementation

3.1 Comments

The implementation of the RANSAC algorithm works fine with all the unmodified patches and is capable to compute an estimated homography matrix that allows a correct overlay. With the modified patches instead, some small errors occurs as well as other bigger errors. Sometimes the patches aren't overlaid at all or are overlaid in a wrong way, as shown in Fig. 5. The errors in the results with the modified patches happens to occur also with the `findHomography()` method in more complex images 7, thus directing the focus on the image quality of the patches and on the correctness of the previous steps of the algorithm such as the SIFT and match operations performed on size modified patches.

Figure 7: Final results with modified patches and `findHomography()` method