LAB 4: Image completion with feature descriptors

Filippo Canderle - 2088236

June 2023

1 Introduction

The objective of this laboratory was to take corrupted images along with their corresponding patches (missing portions of the corrupted image) and seamlessly integrate them into the image to restore a complete and accurate representation of the original corrupted image. This goal was achieved by leveraging a feature extractor as part of the image restoration process, and applying an homography that maps the patch features on the image features.

2 Dataset

To evaluate the quality of our implementation, five datasets with varying levels of difficulty, details, and scenarios were utilized. These datasets, namely "pratodellavalle," "scrovegni," "venezia," "international," and "starwars," encompassed the following components: the corrupted image, a variable number of patches (typically 3 or 4) available in both "simple" and "difficult" versions, where certain rotations and translations were applied. Additionally, the original image was always provided for assessing the reconstruction quality.

3 Methods

In the main function, the code begins by loading an image to be completed from a specific dataset. The image is displayed using the imshow and waitKey functions of the OpenCV library. The code initializes variables such as the number of patches (k), the path to the patch images, and containers to store keypoints, descriptors, and patch images.
A loop is executed for each patch image. Within the loop, each patch image is loaded, and keypoints and descriptors are extracted using the Speeded-Up Robust Features (SURF) algorithm. The extracted keypoints and descriptors are stored in dedicated containers for further processing. SURF is also applied to the base image to extract keypoints and descriptors, which are then visualized using the drawKeypoints function of OpenCV. The resulting image with

keypoints is displayed and saved.
A brute-force matcher is created using the L2 norm, and matches are obtained between the descriptors of the base image and each patch image. A ratio test is applied to obtain valid matches, which are stored in a vector. Corresponding keypoints are extracted from the base image and patch images based on the valid matches. These keypoints will be used to compute the perspective transformation matrix (H) using the RANSAC algorithm.
Optionally, the code draws and displays the correspondences between the base image and the current patch image, depending on the value of the split variable. The perspective transformation matrix (H) is computed using the findHomography function, which takes the keypoints of the patch and base images as input. The pastePatch function is called to paste the current patch image onto the base image using the computed transformation matrix (H). The vectors storing keypoints and matches are cleared for the next patch image. After completing the loop, the final completed image is displayed using imshow and saved to a file.
The pastePatch function performs a perspective transformation of a patch image using the provided transformation matrix H and overlays it onto the base image. A binary mask is applied to the transformed patch image to avoid obvious overlaps. Finally, the transformed patch image is copied onto the base image using the mask.

4 Results

In this section, we aim to present the number of keypoints and descriptors computed by the SURF algorithm for each patch in every dataset, along with the number of matches between the keypoints in the patch and the keypoints in the corrupted image.
Subsequently, we illustrate the various steps of the restoration process for each image: starting with the corrupted image, followed by the computation of keypoints using the SURF algorithm, and finally, the restored image achieved through an affine transformation that maps the patch onto the original image.
This presentation provides a comprehensive overview of the keypoint extraction and restoration steps undertaken in our analysis.

Table 1: Starwars Dataset

Patch	Keypoints	Descriptors	Matches
0	113	113	68
1	163	163	149
2	316	316	303
3	459	459	565

Table 2: Venezia Dataset

Patch	Keypoints	Descriptors	Matches
0	375	375	178
1	627	627	509
2	343	343	673
3	410	410	823

Table 3: International Dataset

Patch	Keypoints	Descriptors	Matches
0	269	269	132
1	149	149	214
2	67	67	241
3	18	18	248

Table 4: Scrovegni Dataset

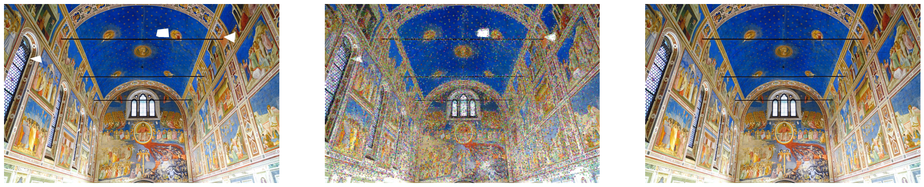
Patch	Keypoints	Descriptors	Matches
0	603	603	340
1	158	158	406
2	127	127	468
3	439	439	684

Table 5: Pratodellavalle Dataset

Patch	Keypoints	Descriptors
0	549	549
1	501	501
2	579	579

(a) Corrupted image (b) Keypoints (c) Final image

Figure 1: Starwars dataset



(a) Corrupted image (b) Keypoints (c) Final image

Figure 2: scrovegni dataset

(a) Corrupted image (b) Keypoints (c) Final image

Figure 3: pratodellavalle dataset

(a) Corrupted image (b) Keypoints (c) Final image

Figure 4: international dataset

(a) Corrupted image (b) Keypoints (c) Final image

Figure 5: venezia dataset

To achieve this outcome, we conducted multiple iterations by varying the value of the "ratiothreshold" parameter for each dataset. The objective was to identify the optimal value that would effectively map the patches onto the corrupted image while minimizing distortions. Among the datasets examined, the "international" dataset posed significant challenges, possibly due to the alternating regions of high-detail complexity followed by smooth areas: as we can observe on related "final image", we are not able to reconstruct a small detail on the wall, due to the fact that we are able to compute few keypoints on this region. This observation underscores the importance of fine-tuning the "ratiothreshold" parameter to accommodate diverse image characteristics and achieve accurate restoration results.

5 Final Remarks

This laboratory has provided practical insights into the workings of a feature extraction algorithm and demonstrated how image mapping within another image can be achieved through homography, while fine-tuning certain parameters such as the ratio threshold to optimize the final outcome.
This simple application offers an overview into how such algorithms can be applied to significantly more complex scenarios in computer vision and underscores the significance of feature extraction algorithms and their broader implications in advancing computer vision research and applications.