1 Feature extractor implementation

We first apply a gaussian filter to the images to remove noise. Using this newly de-noised image we can compute the gradients in both x and y direction. This is done using numerical differentiation by subtracting column wise for x and row wise for y and dividing the result by two. The basic idea here is, the greater the resulting gradient value for a given pixel, the greater the change in color and thus the more likely we are looking at a corner.

We then compute the intensity matrices for each pixel, we do this by computing the three different intensity values Ix^2, Ixy and Iy^2 and storing them in separate matrices. For every pixel we then construct the surrounding intensity matrices and add them up which gives us the Harris matrix H. The surrounding neighbourhood we consider consists of a 3×3 patch. Finally, dividing the determinant of H by the trace of H gives us the Harris response.

Once we know the Harris response for each pixel we threshold it so that the remaining responses represent the 1% highest responses. We then apply non-maximum suppression to extract the local maximum of these responses. To match these remaining responses between images we compute a descriptor for each response, for this we use a 9×9 patch. If the squared distance between two patches is less than some threshold ε we connect them.

Since the code is written to be intuitively understandable it is not very efficient, which is why we divide the size of the images by half to reduce the computation time. This could also be implemented using highly optimized code profiting from only matrix operations and convolutions with kernels.

2 Comparison

Compared to the other two images the Harris corner detector works well on the default image as can be seen in figure 3. This is the result of only a small change from one image to the other. Since the descriptor patches are still very similar to each other the SSD matching performs equally well compared to SIFT, this cannot be said for the other two images. For the default images we chose $\varepsilon = 0.02$ as the descriptor matching threshold.

The star wars poster image was used as it represents an easy case for corner detection as the gradients are either zero or maximal as the image itself is only black and white. The amount of distortion by taking another image of the same poster from a different angle is enough to cause issues for the SSD feature matching, SIFT on the other hand is still able to find reliable matches as it does not only depend on the similarity of the neighbourhood. For this image pair the SSD feature matching is worse than for the default image but still far better than the feature matching of the next image pair. For the star wars poster image we chose $\varepsilon = 0.3$ as the SSD threshold.

For the last image the Harris corner detection still works well but fails to capture large parts of the green text on the top in the second image (see fig. 12). This leads to problems when trying to match those areas from the first image into the second image resulting in large parts being mapped onto a few features. The change in perspective is too strong for SSD feature matching resulting in a poor matching as can be seen in figure 13. SIFT, however, has no difficulties in finding the correct matches, again not having to rely solely on neighbourhood patches. An interesting observation is that for SSD many of the features corresponding to the black letters in the first image are mapped to the accute accent in the second image. This is because the acute accent in the second image most closely resembles a vertical line which is part of the neighbourhood of the letters in the first image. This was expected as SSD cannot handle rotation. For the book images we chose $\varepsilon = 0.45$ as the descriptor extraction threshold.

(a) no suppression (b) non-maximum suppression

Figure 1: I1 using Harris corner detector showing difference in suppression

(a) no suppression (b) non-maximum suppression

Figure 2: I2 using Harris corner detector showing difference in suppression

Figure 3: matched features, extracted using Harris corner detection

Figure 4: feature extraction using SIFT

Figure 5: matched features, extracted using SIFT

(a) no suppression

(b) non-maximum suppression

Figure 6: sw1, features extracted using Harris corner detection

(a) no suppression (b) non-maximum suppression

Figure 7: sw2, features extracted using Harris corner detection

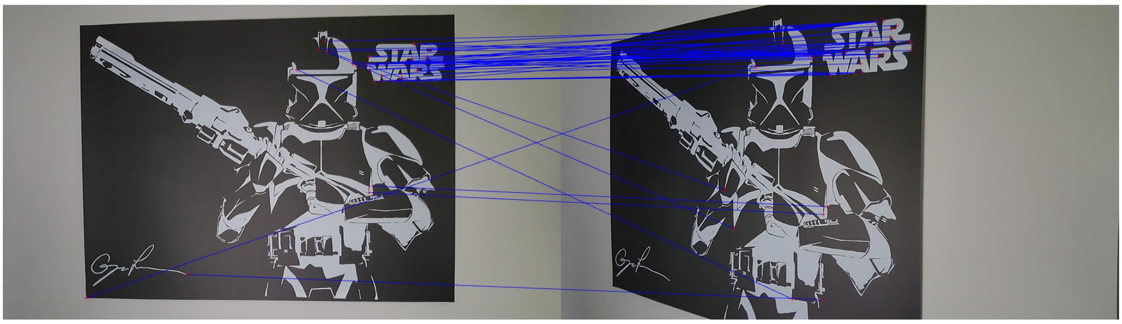


Figure 8: matched features, extracted using Harris corner detection

Figure 9: feature extraction using SIFT

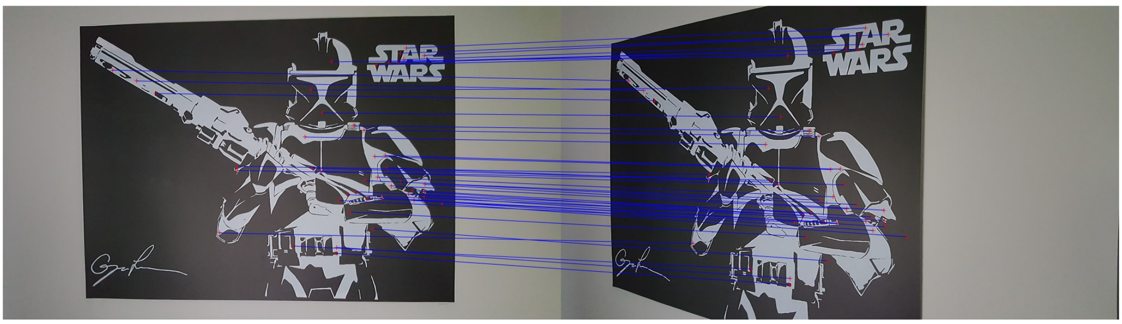
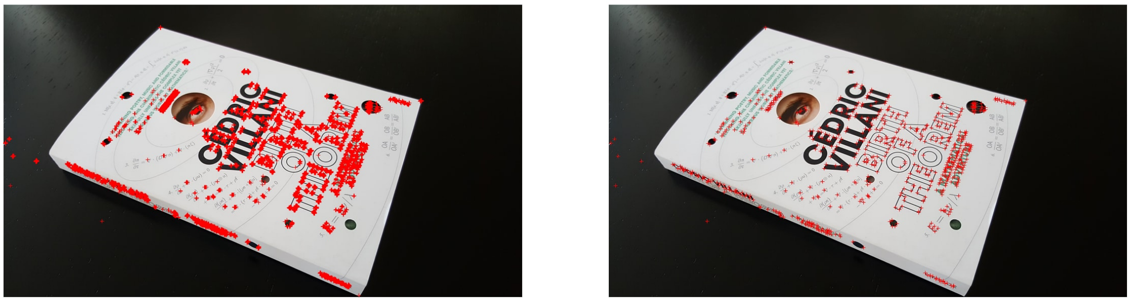


Figure 10: matched features, extracted using SIFT

(a) no suppression (b) non-maximum suppression

Figure 11: cv1, features extracted using Harris corner detection



(a) no suppression (b) non-maximum suppression

Figure 12: cv2, features extracted using Harris corner detection

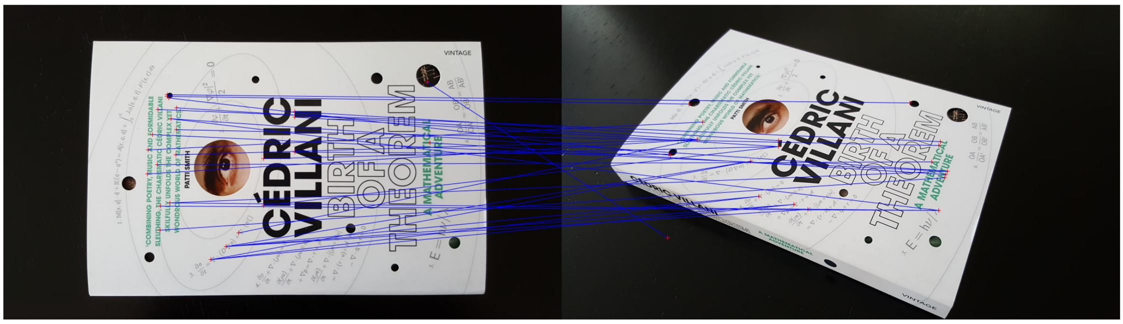


Figure 13: matched features, extracted using Harris corner detection

Figure 14: feature extraction using SIFT

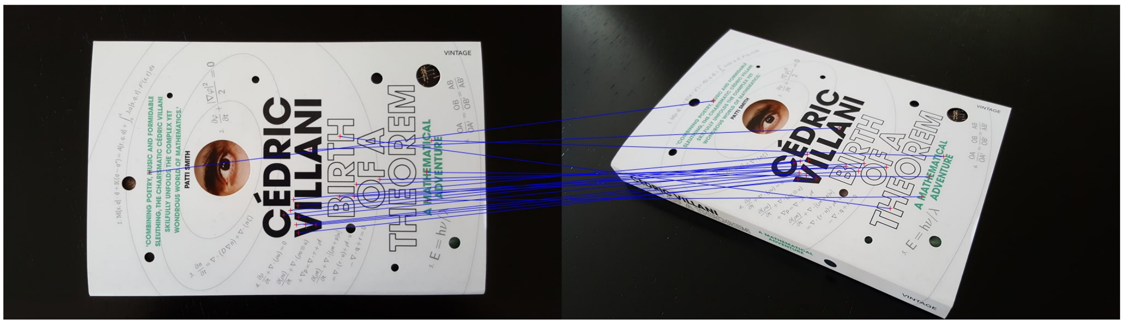


Figure 15: matched features, extracted using SIFT