Computer Vision Exercise 1

Pierre Beckmann

October 6, 2016

1 Feature extraction

The Harris Corner detector starts by blurring the input image using a gaussian filter. This reduces the noise and therefore eliminates potential noise induced and non relevant corners. I decided to add the sigma for the gaussian matrix as a parameter to the function. After some testing sigma = 2 seems to yield the best results. The importance of the filter was the most obvious when proceeding with the matches. Without a filter the algorithm would return a lot of wrong matches induced by the randomness of the noise (Fig. 1).

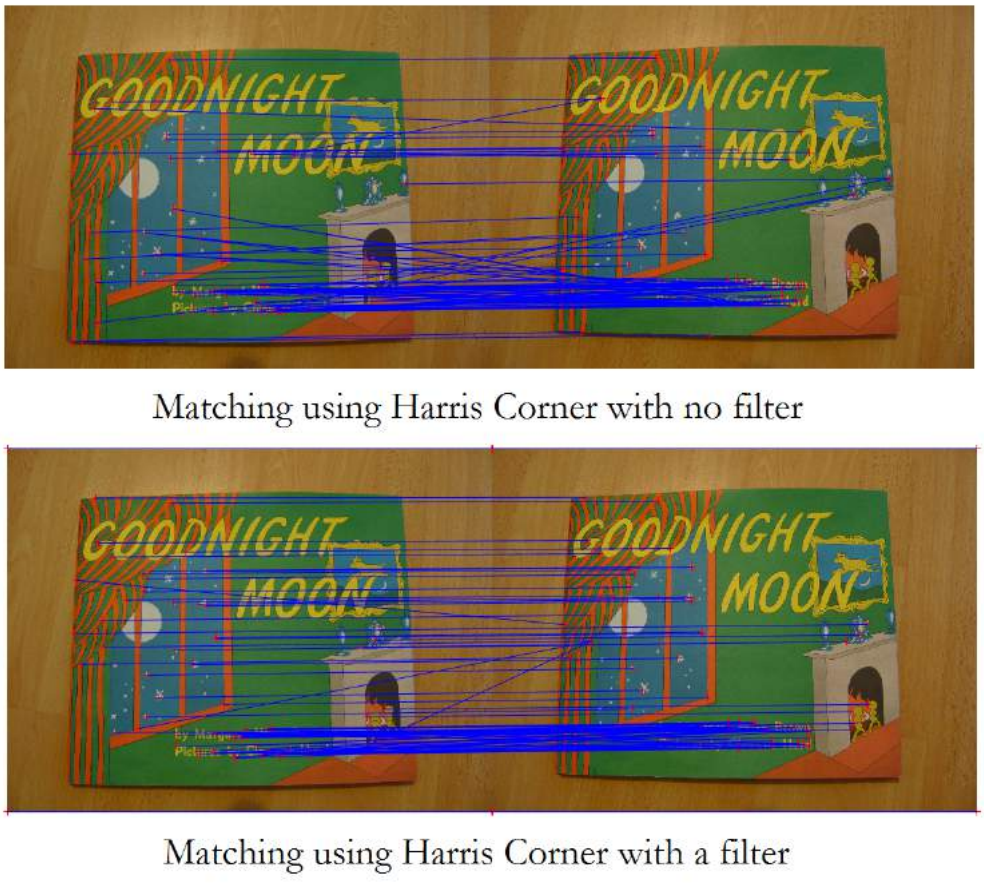

Figure 1: **Importance of the Gaussian filter.** The two cases were done using different thresholds to get a comparable number of Harris corners.

The algorithm then computes the gradients I_x, I_y of the image in the x and y direction. Next the trace is

calculated for each pixel and stored in a matrix H. Only the traces above a certain threshold are stored and kept as Harris Corners.

$$H = \begin{bmatrix} Ix^2 & IxIy \\ IxIy & Iy^2 \end{bmatrix}, K = \frac{det(H)}{trace(H)} = \frac{Ix^2Iy^2 - (IxIy)^2}{Ix^2 + Iy^2}$$

To sum the traces of the 8 neighbours we use matrix convolution and the following kernel.

$$C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

1.1 Non Maximum Supression

Non Maximum Supression is used to eliminate non maximum keypoints in a 3 pixel radius region (Fig. 2). This is done using $ordfilt2()$ which sets all traces of a 3∗3 pixel patch to the maximum of the same patch in a new matrix M. Afterwards we only keep the corners that are found in H and M.

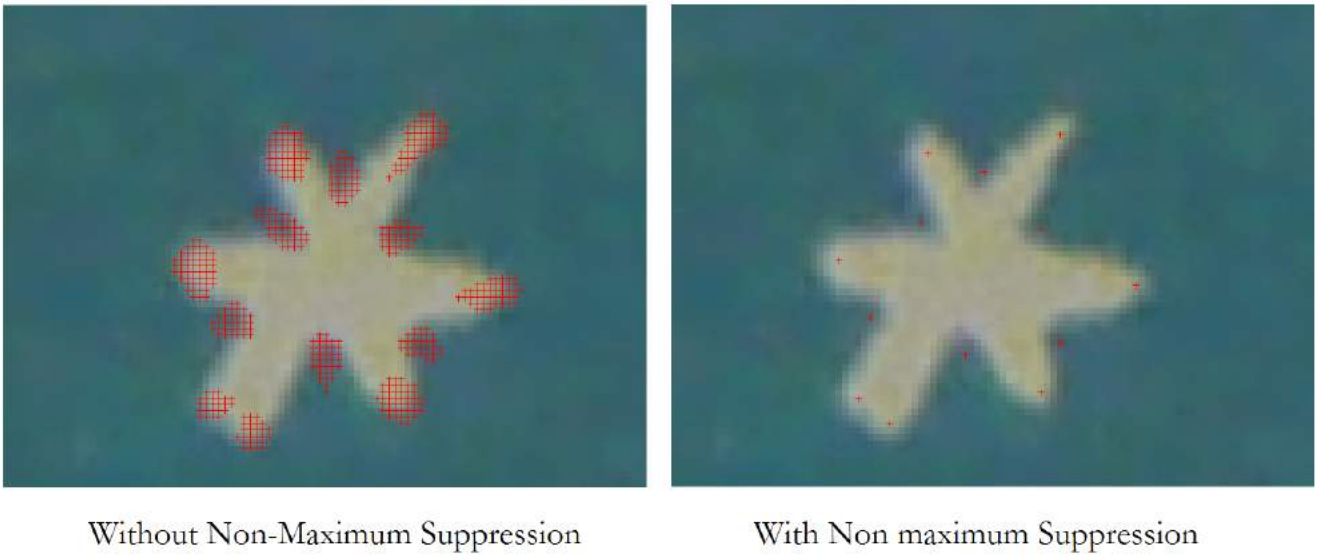

Figure 2: **Non Maximum Supression.** Non Maximum Supression eliminates a lot of unecessary Harris Corners.

2 Feature Matching

2.1 Feature Descriptor

For every Harris Corner we extract a 9∗9 pixel patch called descriptor. Instead of storring this in a 9∗9 matrix the descriptor is stored in a 81 long vector. This way the n descriptors can be stored in a 81∗n matrix called descr. Keypoints situated in the corners or borders don't always have 4 pixels on each side. This is why we proceeded with a zero padding of the image matrix putting 4 layers of zeros around it. Now we have to be careful with accessing the matrix with indexing because everything is shifted 4 pixels to the right and down.

2.2 SSD Feature Matching

By summing the squared differences of the 81 long vectors we can determine the similarity between corresponding pixel patches. This way we can compute the best matches as a matrix soring the indices of matching descriptors using a certain threshold.

3 SIFT

3.1 SIFT Features

SIFT bundles a powerful feature detector and a feature descriptor. The feature frames can have different sizes and different orientations opposed to the Harris Corner detector we implemented.

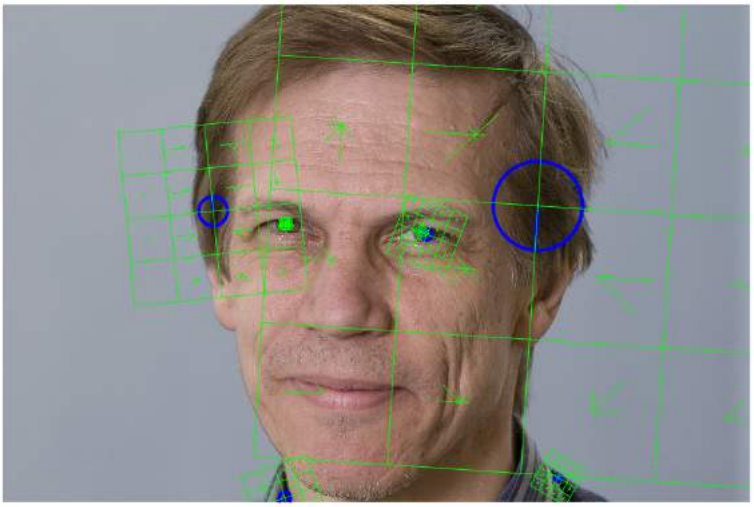

Figure 3: **SIFT Features detector and descriptors.** The frames have different sizes and orientations.

3.2 Comparing Feature Matching

Figure 4: **Harris Matches versus SIFT Matches.**

We can see that the SIFT Matches all seem to be correct matches whereas the Harris Matches have some false positive. The option to have different feature frame sizes allows to recognize part of the moon as a bigger circle for the SIFT matching. This allows to identify and match features more precisely. Because Harris only detects pixels it finds a lot of matches in the text in the lower half of the book cover. SIFT also detects the stars as an entity instead of finding multiple matches as in Harris. Clearly the SIFT Matching is superior in every aspect.