Homework 2

Stefan Weber

Introduction to Signal and Image Processing

April 08, 2020

1 Linear Filtering

1.1

Figure 1 illustrates how an average box filter is created.

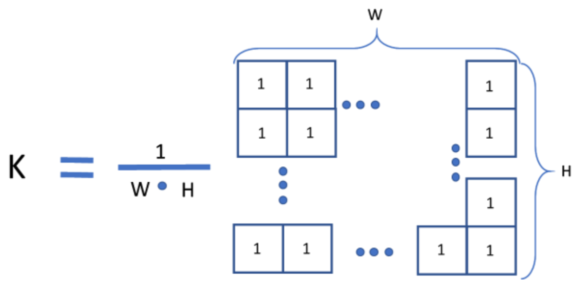

Figure 1: Graphical representation of an average box filter.

1.2

Figure 1 illustrates what happens in a convolution. A convolution differs from a correlation in the form that the filter kernel is flipped before the calculation is performed. The convoluted image has the size $(m+k-1),(n+l-1)$ which cannot be seen in Figure 2.

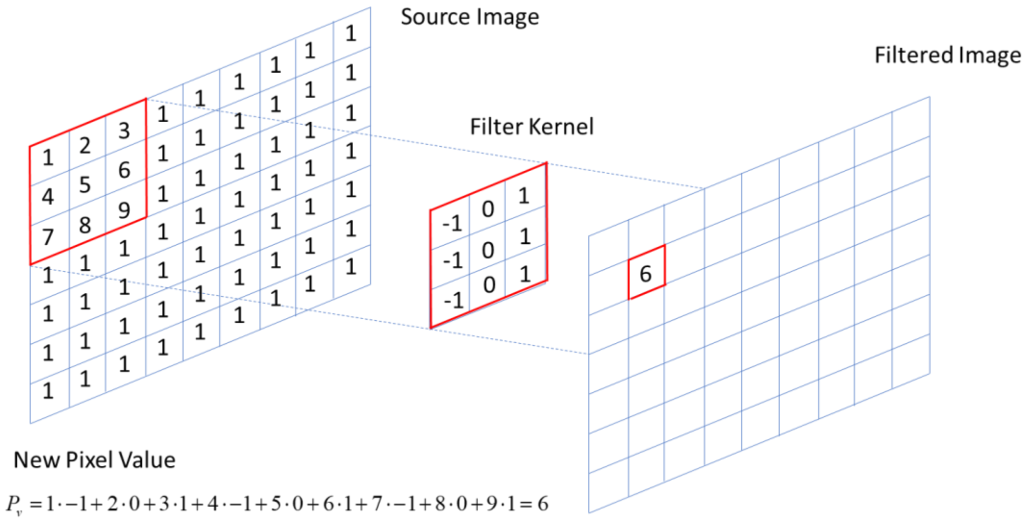
Source Image
Filtered Image
Filter Kernel
New Pixel Value
$P_v = 1\cdot -1 + 2\cdot 0 + 3\cdot 1 + 4\cdot -1 + 5\cdot 0 + 6\cdot 1 + 7\cdot -1 + 8\cdot 0 + 9\cdot 1 = 6$

Figure 2: Graphical representation of a convolution. Filter kernel was flipped before the operation otherwise it would be a correlation.

1.3

An average filter blurs the image and reduces noise as it can be seen in Figure 3.

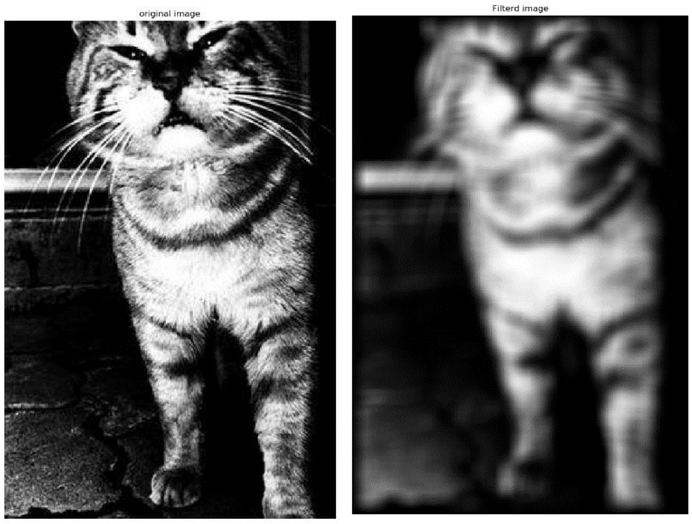

Figure 3: Left: Original Image, right: Box filtered image.

1.4

x_i are n equally distanced distributed segments from $-\frac{G_{Length}}{2}$ to $\frac{G_{Length}}{2}$ and n is an odd number. The values of the filter $\vec{G}$ sums up to 1.

$$\vec{G} = \frac{\left(e^{-\frac{x_1^2}{2\cdot\sigma^2}}, \quad e^{-\frac{x_2^2}{2\cdot\sigma^2}}, \quad \ldots \quad e^{-\frac{x_{n-1}^2}{2\cdot\sigma^2}}, \quad e^{-\frac{x_n^2}{2\cdot\sigma^2}} \right)}{\sum_{i=1}^{n} e^{-\frac{x_i^2}{2\cdot\sigma^2}}}$$

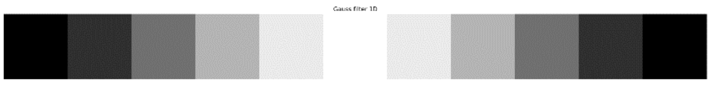

Figure 4: Graphical representation of a 1d gaussian filter with length 11.

1.5

To create a 2D Gaussian filter kernel a 1D Gaussian filter kernel can be convolved with its transpose.

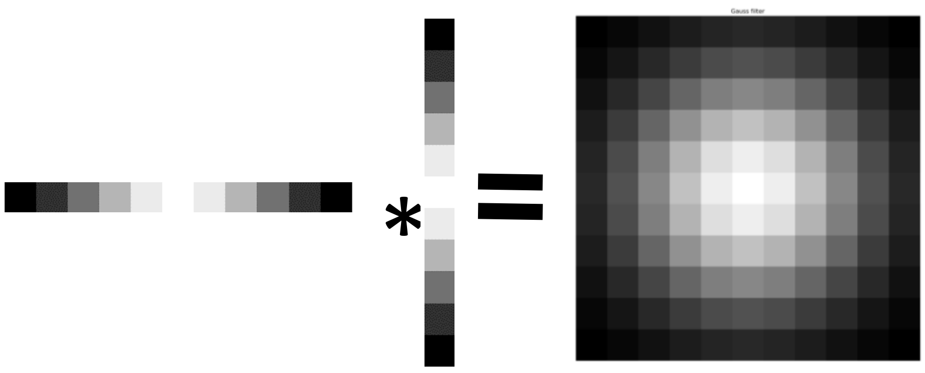

Figure 5: Graphical representation of a 2D Gaussian filter kernel with length 11.

1.6

A gaussian filter has similar effects on an image as an average filter. Noise is reduced and the image is blurred. The gaussian weights the neighbouring pixels differently from the unfiltered image compared to the average filter. This results in less spread contrast boarders.

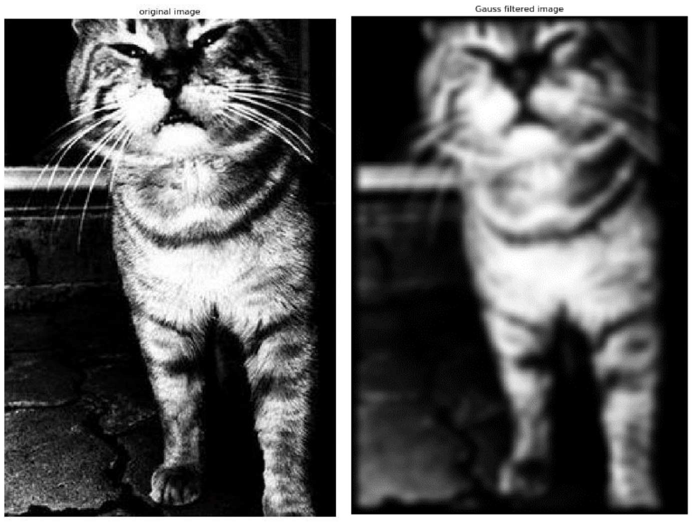

Figure 6: Left: Original Image, right: Gaussian filtered image.

1.7

It takes less multiplications when a 1D filter is applied twice. First in the x-direction and then in the y-direction. Figure 7 shows that the result is the same when a 1D filter is applied twice and a 2D filter is applied once. In the first case $2n$ multiplications are needed in the second case n^2 multiplications are needed.

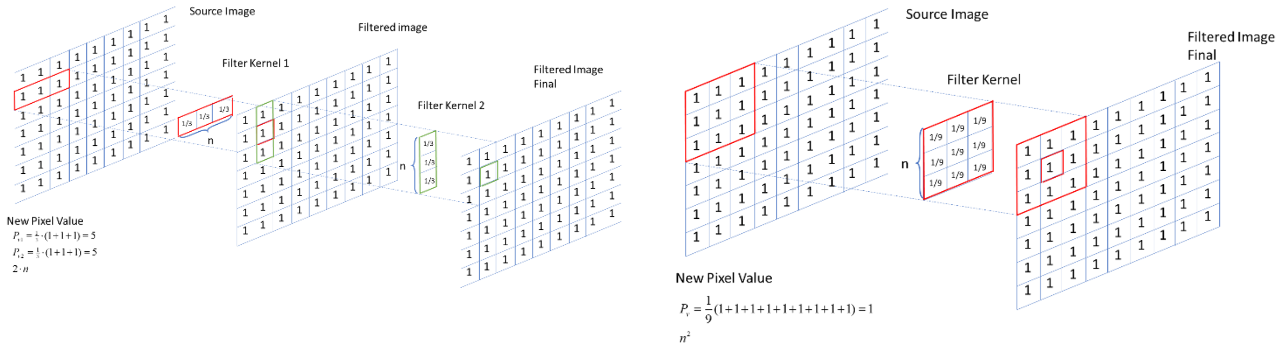

Figure 7: Left: Graphical representation of a 1D filter applied twice. right: Graphical representation of a 2D filter applied once.

1.8

Figure 8 shows that the computation time rises linear in 1D filtering and quadratic in 2D filtering.

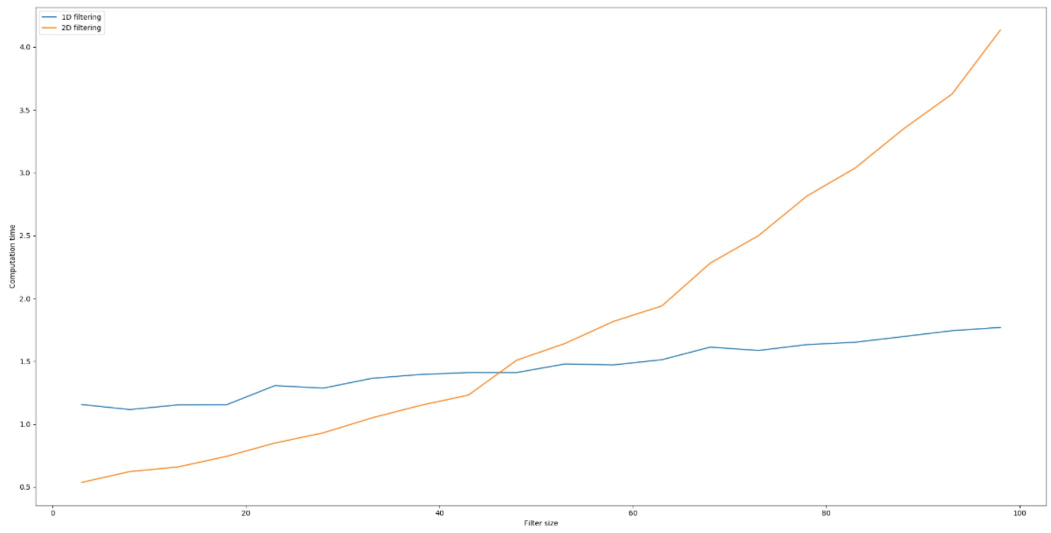

Figure 8: Filter size vs computation time for 1D filtering and 2D filtering

2 Gradients and Edge Maps

2.1

The derivative and the gaussian filter can be combined because convolution is associative.

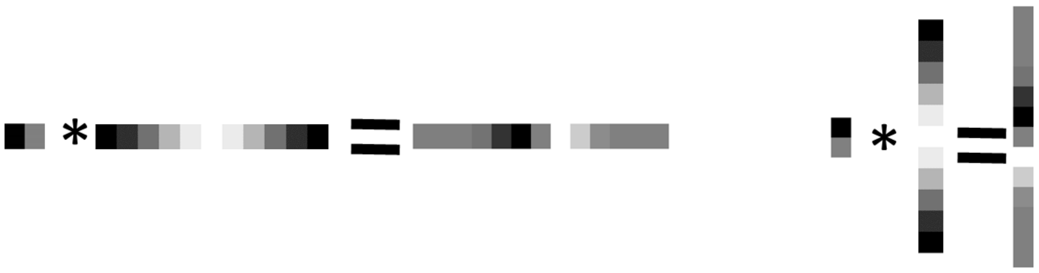

Figure 9:Left: Convolution in the x-direction with the derivative and the gaussian filter. Right: Convolution in the y-direction with the derivative and the gaussian filter.

2.2

After deriving the image in x- and y-direction all changes in the intensity are visible. At the edges the changes in intensity between neighbouring pixels are high.

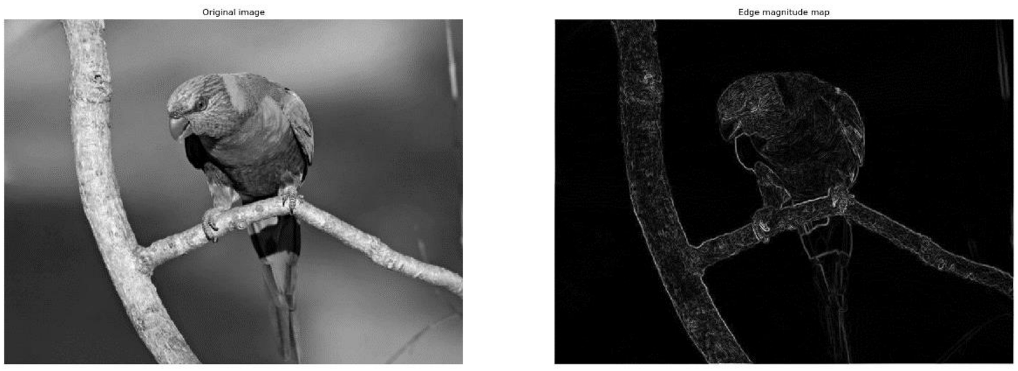

Figure 10:Resulting edge magnitude map after applying a the derivative in x- and y-direction.

2.3

A threshold of 30 was chosen to generate the magnitude map. The threshold should be that hight that the edge remains as single pixel. If the threshold is too high too few edges will be detected and the edges will have gabs. Is the threshold too low too many false positive edges will be detected.

To find the threshold a trial and error approach was used.

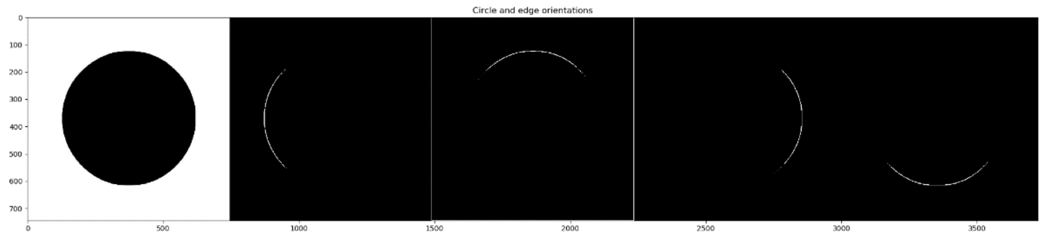

Figure 11: Resulting edge magnitude map for four different direction intervals and the original image circle.jpg

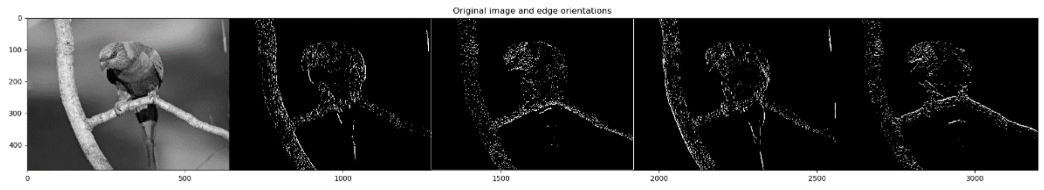

Figure 12: Resulting edge magnitude map for four different direction intervals and the original image bird.jpg

3 Harris Corner detection

To generate the image with the corner response R following steps were done:

1. Create a filter for smoothening the image and compute the derivatives:
   $F_\sigma^x = D_x * G_\sigma, F_\sigma^y = D_y * G_\sigma$
2. Compute the derivatives in x and y directions: $I_x = F_\sigma^x * I, I_y = F_\sigma^x * I$
3. Compute the product of the derivatives elementwise: $I_{x^2} = I_x I_x, I_{y^2} = I_y I_y, I_{xy} = I_x I_y,$
4. Sum up the values for each pixel in a given window: $S_{x^2} = w * I_{x^2}, S_{y^2} = w * I_{y^2}, S_{xy^2} = w * I_{xy^2}$
5. Compute the determinant and the trace of the matrix
   $$H(x,y) = \begin{bmatrix} S_{x^2} & S_{xy} \\ S_{xy} & S_{y^2} \end{bmatrix}, det(H) = S_{x^2} S_{y^2} - S_{xy}{}^2, trace(H) = S_{x^2} + S_{y^2}$$
6. Compute the corner response: $R = \det(H) - k \cdot trace(H)^2$

3.1

The corners have large positive values and appear bright in the corner response. The edges have large negative values and appear dark. The flat areas are around zero and appear grey in the image.

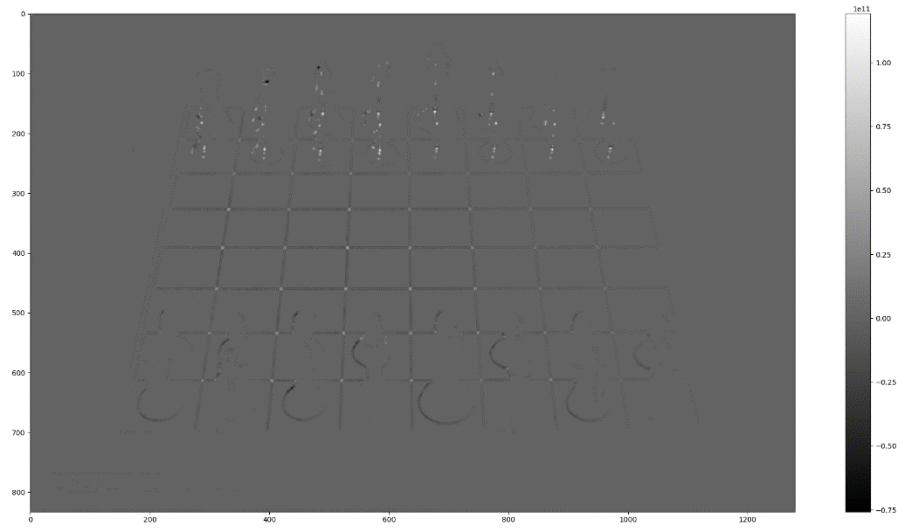

Figure 13: Corner response R of the image chessboard.jpg

3.2

The R values of the rotated image do not change significantly. The corner response R is invariant of rotation.

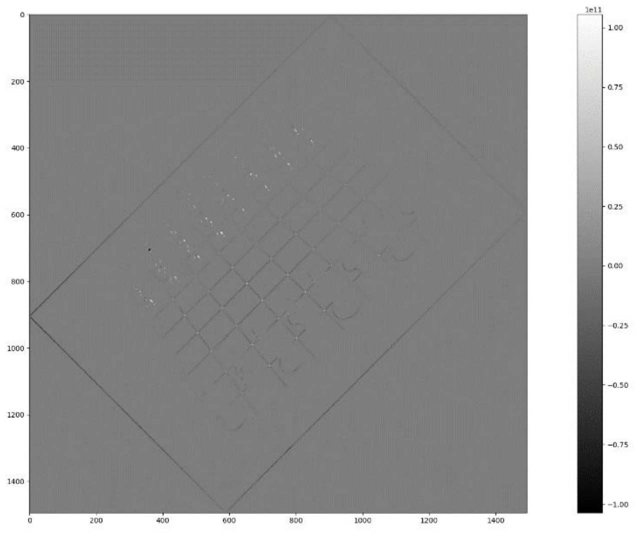

Figure 14: Corner response R of the 45° rotated image chessboard.jpg

3.3

The R values of the rotated image change and all edges and corner are enlarged. The corner response R is variant of scaling.

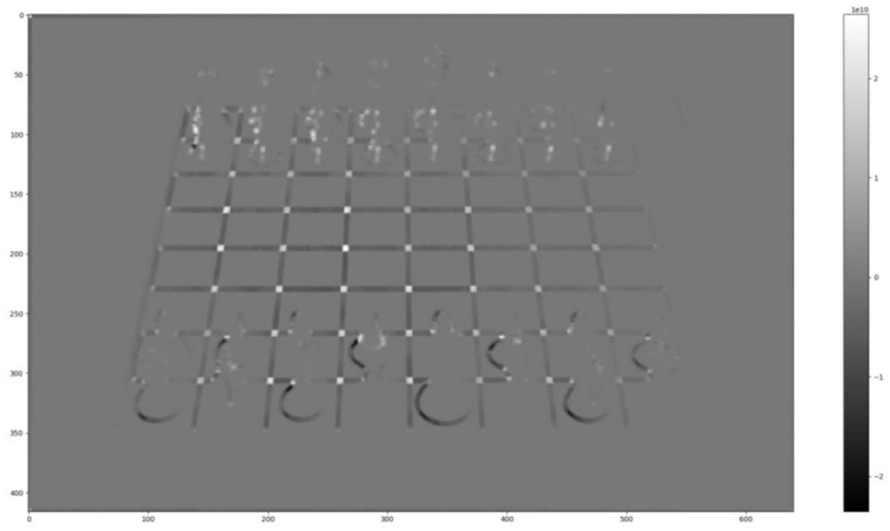

Figure 15: Corner response R of the downscaled image chessboard.jpg *by a factor of 0.5*

3.4

The corner response R is invariant of rotation. Figure 16 shows that the ellipse of the magnitudes of the eigen-values remains the same after a rotation.

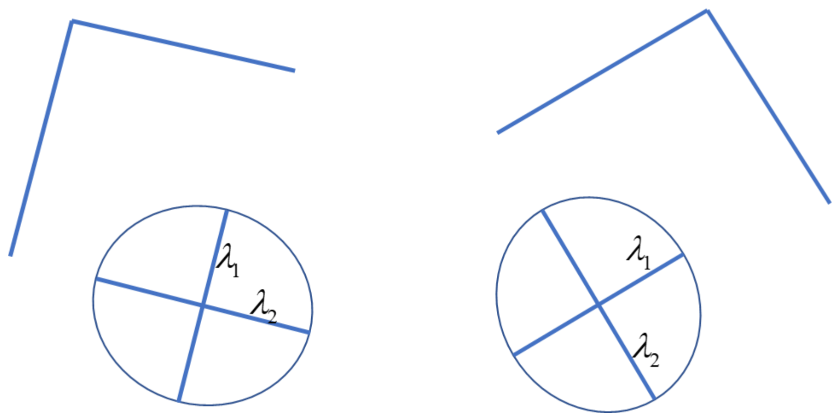

Figure 16: A corner which is rotated 45° and the corresponding ellipse of the eigenvalues.

The corner response R is variant of scaling. Figure 17 shows that in the image with a high number of pixels the corner will be detected as an edge and in the image with fewer pixels the corner will be detected as corner.

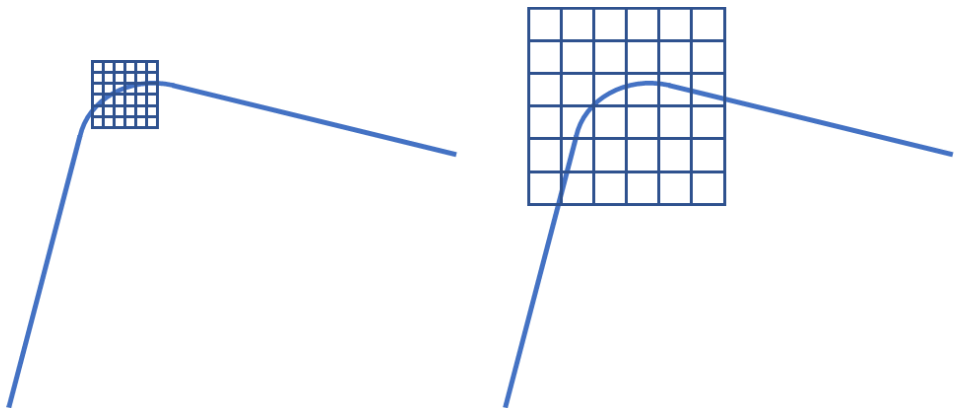

Figure 17: Corner detection window of the same pixel size Left: original image, Right: downscaled image.