Submission before: 18.01.2016
Discussion on: 19.01.2016
Submission on stud.ip, submission folder for sheet.
Please submit a zip file containing the .m files for Matlab programming tasks.

Exercise 1 *(SIFT keypoint detection – 20p)*

Implementation of SIFT keypoint detection, inspired by [1]. Use the image `lighthouse.png` for testing.

(a) Calculate the scale space for an image I (no built-ins allowed). Calculate 4 octaves with 5 scales each. For the calculation of one octave proceed as follows:
Smooth the image with $\sigma_0 = 0.5$ with a kernel size of $\texttt{ceil}(4 \times \sigma_0)$. This is the first image of the octave. For the calculation of the other scale images ($i = 1, \dots, 4$), the value of σ is calculated as $\sigma_i = k^i * \sigma_0$ with $k = \sqrt{2}$ and kernel size $\texttt{ceil}(4 \times \sigma_i)$.

(b) Calculate the Difference of Gaussian images (DoG) from the scale space for each octave ($\text{DoG}_{i-1} = G(x, y, \sigma_i) - G(x, y, \sigma_{i-1})$). Where $G(x, y, \sigma_i)$ is the Gaussian smoothed image with kernel σ_i. The result will be 4 DoG images per octave.

(c) Detect extrema in the DoG images. Check every pixel in the DoG images and compare it to all its neighbors (surrounding, as well as those neighbors in the adjacent DoG images). This is a total of 26 comparisons. If the pixel is greater than or less than all its neighbors, it is a possible keypoint. This check is not performed for the top- and lowermost DoG images within each octave, because there are not enough neighbors.

(d) ~~Actually, we would have to calculate subpixel locations now, but we will omit that part. Instead~~ we will remove keypoints that lie on edges. This procedure is similar to that of the Harris corner detector, however, in the following the Hessian matrix is used. Calculate the Hessian matrix of the DoG images on which (possible) keypoints were found. If $\text{Tr}(H)^2 / \text{Det}(H) > (r+1)^2/r$, with $Tr(H) = D_{xx} + D_{yy}, \text{Det}(H) = D_{xx}D_{yy} - D_{xy}^2$ then the keypoint is rejected (section 4.1 in [1]). Find a suitable value for r.

(e) Keypoints are also rejected if their magnitude in the DoG image is less than a threshold (find a good threshold yourself).

(f) Display the remaining keypoints on the image.

Please keeep in mind, that you work on the DoG image on which that keypoint was found!

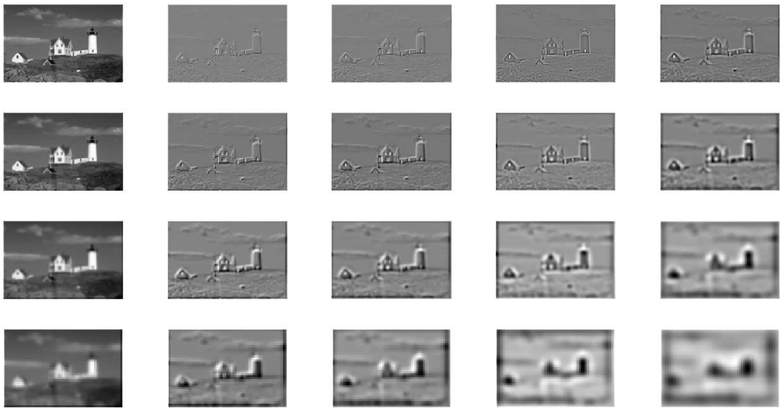

Figure 1: Images of 4 octaves with 5 scales (images are resized) Right: Sample keypoints

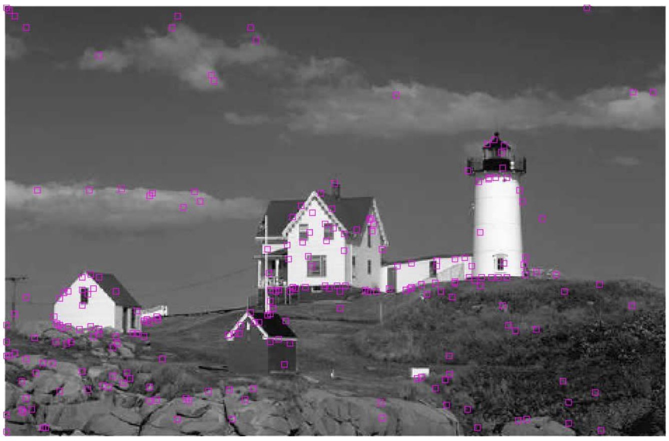

Figure 2: Sample keypoints

References

[1] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.