Assignment: Feature Detection and Scale-Space Theory
Signal and Image Processing

March 11, 2023

You can work on this assignment and submit your solution (report and code) as a GROUP. This assignment counts towards your grade and have to be submitted in order to pass the course. You must follow the report guidelines found in `guidelines.pdf`. The page limit for this assignment is **17 pages** including everything, i.e. illustrations and code snippets.

1. **Fixed scale feature detectors**

   1.1. (1 point) Use the `skimage.feature.canny` function on the `hand.tiff` image. Try different settings for the parameters `sigma`, `low_threshold`, and `high_threshold` of the `canny` function.
   Deliverables: Include an illustration showing the results of the different settings and explain what the effect is of each of the parameters based on these results.

   1.2. (1 point) Use the `skimage.feature.corner_harris` function on the `modelhouses.png` image to compute a Harris corner response image (a.k.a. feature map). Try different settings for the parameters `sigma` and `k` for `method='k'` of the `corner_harris` function. Also try to fix `sigma` and change method to `method='eps'` and try different values of the `eps` parameter.
   Deliverables: Include an illustration showing the results of the different settings and explain what the effect is of each of the parameters based on these results.

   1.3. (1 point) Write a Python function that finds Harris corners by finding local maxima in the feature map generated by the `skimage.feature.corner_harris` function. To find the local maxima you should use the function `skimage.feature.corner_peaks`.
   Deliverables: Include in the report the code for your function. Apply this function to the `modelhouses.png` image and create a figure of the 250 strongest corner points overlaid on the `modelhouses.png` image. Hint: `corner_peaks` has a parameter to automatically select N strongest points (e.g. having the largest response value). Indicate your choice of parameter settings in the caption text of the figure.

2. **Feature detection and image transforms**

   2.1. (2 point) The image `textlabel_gray_small.png` contains a label with some text on which we would like to transform such that the label is horizontal in order for better reading of the text (e.g. necessary if we want to pass it to an optical character recognition (OCR) algorithm). It is not enough to simply rotate

the image 90 degrees clockwise. Implement in Python an automatic method for detecting the corners of the label and use these to estimate a transformation that estimate the proper rotation of the image. You can use your Harris corner detector from question 1.3., but you need to adjust the parameters carefully and some heuristics to find the label corners from the points detected by the Harris corner detector. Hint: Maybe you only need two opposing sides of the label in order to estimate the required rotation. Once you have corner points, you can compute an angle needed for rotation around the center of the image and use `skimage.transform.rotate` to rotate the label image.
Deliverables: Briefly describe the essential steps in your solution including relevant code snippets. Remember to specify all relevant parameters in the report. Include illustrations of the detected label corner points overlayed on top of the image and the resulting rotated image and state your estimated rotation angle.

3. **Scale-space blob detector**

3.1. (1 point) The convolution of two Gaussian functions is also a Gaussian and we have that

$$G(x, y, \sigma) * G(x, y, \tau) = G(x, y, \sqrt{\sigma^2 + \tau^2}) \ , \tag{1}$$

where a Gaussian function is defined as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \ . \tag{2}$$

Consider the function made from this analytical expression,

$$B(x, y) = G(x, y, \sigma) \ . \tag{3}$$

We will use this function as a synthetic model of a blob in an image.
Using Python, make a discrete image from the function in eq. (3) for some fixed σ by sampling the x, y axes appropriately. Visually illustrate your solution and eq. (1) by calculating images from its scale-space,

$$I(x, y, \tau) = B(x, y) * G(x, y, \tau) \ . \tag{4}$$

Deliverables: Provide a code snippet and explanation of your solution as well as illustrations of your solution. Confirm eq. (1) visually constructing a blob image at scale $\sigma = 1$ and convolve it with a Gaussian with scale $\tau = 2$. Also construct a model blob image with scale σ corresponding to this convolution using eq. (1). Compute and visualize the difference image between these two images and comment on the result.

3.2. (1 point) Next you should prove that the expression in eq. (1) is correct.
Deliverables: Mathematical proof with essential steps and explanations. Hint: You can prove this using the Fourier transform and the convolution theorem.

3.3. Consider the 2-dimensional scale normalized partial derivatives of order $m + n$ at scale τ,

$$I_{x^m y^n}(x, y, \tau) = \tau^{\gamma(m+n)} \frac{\partial^{m+n} I(x, y, \tau)}{\partial x^m \partial y^n}, \tag{5}$$

where $\gamma \in \mathbb{R}$ is a parameter of the scale normalization and $I(x, y, \tau)$ is the scale space of an image. Now consider the scale normalized Laplacian of the blob image defined in eq. (3),

$$H(x, y, \tau) = I_{xx}(x, y, \tau) + I_{yy}(x, y, \tau), \tag{6}$$

Using $\gamma = 1$, solve the following:

i. (1 point) **Deliverables:** Write the closed form expression for $H(x, y, \tau)$ including the essential steps and explanation in the development of the expression.

ii. (1 point) Consider the point $(x, y) = (0, 0)$ and derive analytically the scale(s), τ, for which $H(0, 0, \tau)$ is extremal. A CAS tool such as Maple (or Wolfram Alpha) may be helpful, or simply solve it by hand.
Deliverables: Mathematical derivation with essential steps and explanations. Characterise these extremal point(s) in terms of being a maximum, saddle point, or minimum in (x, y, τ).

iii. (1 point) **Deliverables:** Confirm your result in Python by plotting $H(0, 0, \tau)$ as a function of τ using the expression from 2.3.i.

3.4. (2 point) Locating the maxima and minima (x, y, τ) in the scale-space of eq. (6) with $\gamma = 1$ applied to images $I(x, y)$ in general is called blob detection with scale selection. Write a Python script that detects the 150 maxima and minima in the scale-space of the `sunflower.tiff` image with the largest absolute value of eq. (6) (You may use the scikit-image function called `peak_local_max`). Indicate each detected point and corresponding detection scale τ with a dot and a circle centred on the point of detection with a radius proportional to τ. Choose different colors so you can distinguish maxima from minima.
Deliverables: An illustration of your results and code snippets showing essential steps in your implementation. What image structure does maxima of eq. (6) represent, and what image structure does minima represent?