Computer Vision 1: Exercise Sheet 4

Summary:

1. Non-linear filters: Median filtering, other rank filters
2. Image gradient

1 Non-linear filters

Non-linear filters do **not** satisfy both the additivity and homogeneity properties. That is, given two images X and Y, two scalars $\alpha, \beta \in \mathbb{R}$, and a filter that is a function f that maps an image to the filtered output, a non-linear filter in general does *not* satisfy $f(\alpha X + \beta Y) = \alpha f(X) + \beta f(Y)$.

1.1 Median filtering

We use the `scikit-image` package for many tasks in the course. The package website can be found at `https://scikit-image.org`.

- Load the image `woman.png` from Moodle.
- Use `skimage.util.random_noise` to add salt and pepper noise, with the default proportion 0.5 of salt vs. pepper noise.
- Using `matplotlib`, plot the elements of the 1st row of the noisy image as a line plot. Verify that you see the salt and pepper noise in the plot.
- Visualize the original image and the noisy image.

A median filter is a non-linear filter that is effective in filtering salt and pepper noise. Recall that the median of a set is obtained by 1) sorting the items in the set in an increasing order, and 2) returning the middle element.

- Use `skimage.filters.median` to apply a median filter on the image with salt and pepper noise. Use a structuring element (`selem`) defined as a 3-by-3 array of ones[1]. Use the default mode for handling array borders.
- Visualize the noisy image and the median filtered image.
- How does the output change when the structuring element is a 15-by-15 array of ones? Draw the image.

1.2 Rank filters

The median filter is an example of a *rank filter*. Rank filters use the local gray-level *ordering* to compute the filtered value. Rank filters in `scikit-image` are listed at `https://scikit-image.org/docs/dev/api/skimage.filters.rank.html`.

- Apply the maximum filter on the image with salt and pepper noise. Use a structuring element of size 3-by-3, and draw the output.
- Now apply a structuring element of size 15-by-15, and draw the output. Why does the result look (almost completely) white?

[1] You can think of the structuring element as defining the local neighborhood over which to apply the median.

2 Image gradient

Define the two gradient kernels K_x and K_y as follows:

$$K_x = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad K_y = \begin{bmatrix} -1 & 1 \end{bmatrix}.$$

Make sure they have the datatype `np.float64` and that they are 2-dimensional arrays. Use `np.reshape` to ensure correct sizes.

- Load the image `woman.png` from Moodle.
- Convolve the image with K_x and K_y to calculate the partial derivatives in the x and y direction, respectively. Use the padding mode `same`.
- Visualize the partial derivative images.
- Calculate the magnitude of the gradient and visualize it. Do you see why the magnitude can be useful for edge detection?
- Calculate the orientation of the gradient using `np.arctan2` and visualize it.