DTU Electro
Department of Electrical and Photonics Engineering

Exercise Topic 6
Segmentation and Keypoints detection

Part 1 - Segmentation (see useful functions below)

1. On the image Blob:
    a. Apply the an edge detection mask (e.g. Sobel)
    b. Add Gaussian white noise to the image and reapply the same edge detection technique
2. On the color images, segment in 2 classes (object of interest/background)
    a. Start with the fish image and a thresholding approach (think of which colorspace would be best)
    b. Pick another image and segment object of interest/background potentially by combining tools

Part 2 - Corner detection - Keypoints detection and characterization (see useful functions below)

On either the Notre Dame images or the Tintin images (e.g. to locate specific episodes in the shop shelves).

1. Implement your own Harris corner detection (cf steps below) and visualize which points you obtain
2. Use SIFT to detect and characterize key points in two images with similar content and calculate the matches

Steps for implementing your own Harris corner detection:

a. Turn the input image into greyscale
b. Compute derivatives along horizontal and vertical direction (Ix and Iy)
c. Compute the components of the second order moment matrix M. Create an array *structM* with dimensions: same as input image and third dimension 3. The 3 channels in the third dimension correspond to: Ix^2, Iy^2 and IxIy respectively
d. Apply a gaussian filter on *structM*
e. Compute the array C whose intensity is the cornerness measure at each pixel
f. Threshold C
g. Apply non-maximum suppression to C

Related useful functions

Purpose	Matlab	Python
Load an image	`imread`	`import imageio.v3 as iio` img=iio.imread('imagepath')
Display an image	`imshow` (be aware that the data type sets the range of expected values → use [low high] range as parameter or [] to scale between min and max), `imagesc`	`import matplotlib.pyplot as plt` Create figure with `fig, ax = plt.subplots()` Display image with ax.imshow(img)
Spatial filtering	fspecial to create filter masks (incl.laplacian), imfilter to apply filter imgaussfilt for Gaussian filter	`cv.Laplacian` to calculate laplacian of an image `cv2.filter2D()` to apply filter `cv2.GaussianBlur()` for Gaussian filter
Additive noise	imnoise to add noise	random_noise from scikit-image link
Find threshold/binarize	graythresh imbinarize	cv2.threshold link
Segmenting methods and utils	imsegkmeans labeloverlay boundarymask superpixels	kmeans2 from scipy link mark_boundaries and slic in scikitimage segmentation link non maximum suppression in scikit-image link matching of descriptors link
Gradient orientation	atan2	`np.arctan2`
Histcounts (remember to specify the bin edges or the number of bins)	Without plot `np.histogram` With plot matplotlib.pyplot.hist	Histcounts (remember to specify the bin edges or the number of bins)