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 colorspaces)
   b. Pick another image and segment object of interest/background potentially by combining approaches

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

1. Implement your own Harris corner detection with the following steps:
   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 correposnd to: Ix2, Iy2 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

2. Implement a simplified of SIFT. In a first time, the scale invariance can be neglected, which yields the following steps:
   a. Apply Harris corner detection to detect keypoints
   b. Estimate the dominant orientation at each keypoint by computing gradient orientation of each pixel of a neighboorhood around your keypoint. Build a histogram of the orientation with 36 bins (for 360 degrees) and select the largest bins
   c. Compute descriptor
      i. Consider a 16x16 neighborhood of your keypoint
      ii. Apply gaussian window
      iii. Compute gradient orientation and magnitude at each pixel
      iv. For each subregion of size 4x4, build a histogram using 8 bins (8 orientations) in which the value for each bin is the sum of gradient magnitudes for pixels having gradients in those orientations
      v. Modify your histogram bins indices to handle rotation invariance (account for b. by subtracting dominant orientation)
      vi. Reorganize your feature into a vector of size 128
   d. Apply your SIFT on two images and compute distance between features of keypoints

Related useful functions

Purpose	Matlab	Python
Load an image	`imread`	`img=Image.open('imagepath')` `np.asarray(img)` with Pillow and NumPy
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`	`plt.imshow(img)` from Matplotlib
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	
Find threshold/binarize	graythresh imbinarize	cv2.threshold link
Segmenting methods and utils	imsegkmeans labeloverlay boundarymask superpixels	
Gradient orientation	atan2	
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)