Mathematical Morphology
Signal and Image Processing

February 26, 2019

This is a group assignment.

1 Fourier Analysis, continued

1. **Fourier Transform – Theory**: The following investigate theoretical properties of Fourier series and transform. Remember to include crucial steps in derivations, and a short comment to each answer.
   (a) Solve Brigham, Problem 5.1
   (b) Solve Brigham, Problem 5.2
   (c) Solve Brigham, Problem 5.3
2. **Fourier Transform – Practice**: Each of these answers should include examples of the input and output, possibly crucial Python code snippets, and definitely a description of which problems were solved, how, and an evaluation of the results.
   (a) Write the function `scale`, which implements convolution with an isotropic Gaussian kernel, parametrized by its standard deviation σ - the scale. Apply it to `trui.png` for a range of scales.
   (b) Spatial derivatives may be written as the multiplication of a kernel in the Fourier Domain. Derive the exact relation and discuss its practicality.
   (c) Implement a function which takes as input two derivative orders (for the x- and y-directions), as well as a 2-dimensional image, and returns the derivative of the image using FFT.

2 Morphology

1. Consider the following masks (black represents foreground/'on' pixels):
   1.1. Using the morphology functions in `skimage.morphology`, and the relevant masks in Figure 1 as structuring elements (SE), answer the following questions. Use `skimage.morphology.binary_dilation` function, and create a small (7 x 7) binary image of a digit (e.g. A, B, 1, 2... and a white background, black foreground) as your input:
      i. Does an SE need to be symmetric? (Test with Mask 1)

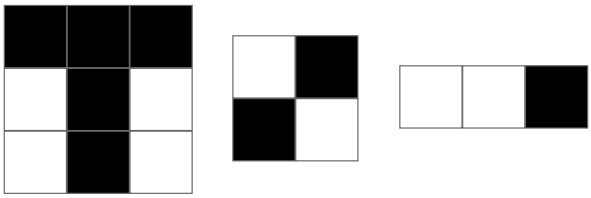


Figure 1: Masks - black represents foreground/'on' pixels

ii. Does an SE need to have an odd number of pixels in both directions? (Test with Mask 2)

iii. Does an SE need to have the middle pixel set to true/on? (Test with Mask 3)

For each question, indicate which pixel of the SE that `skimage.morphology.binary_dilation` defines to be the centre pixel, show the output and provide an explanation.

1.2. Now interchange the input image and the SE (i.e. use the SE as input, and your test image as SE), and show the results for each of the three masks. What is going on?

1.3. Does this mean that the SE and input image can be interchanged?

1.4. Assuming that they can indeed be interchanged, why would this be useful?

2. Apply, separately, morphological 'opening' and 'closing' with a ball SE (`disk`, radius 2), to the input image `cells_binary.png` (available on Absalon).

2.1. For each, show the resultant image. Highlight a few key locations and describe the changes.

2.2. Why are the two operations not commutable?

2.3. What are the challenges of using just these two operations for segmenting this particular image into individual cells?

2.4. What other morphological operations would be useful for segmenting these cells?

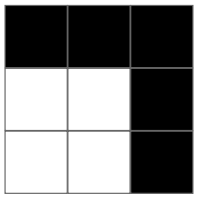
3. Apply the following operations to the `blobs_inv.png` example image on Absalon.

- Hit-or-Miss
- TopHat (white top hat in `skimage`)
- BottomHat (black top hat in `skimage`)

using the following structuring elements: vertical line (5 x 1), disc (5 x 5), corner of a single pixel width, 3 x 3,

What features do each combination pick-up?

4. Load the binary image `digits_binary_inv.png` available on Absalon. Create a mask using the first digit from the first row. Use it in a simple hit-or-miss morphology operation to try and locate other examples of the same digit within



the image. Show your result as overlaid markers (e.g. coloured crosses) located where the hit-miss mask finds a match. How well does it work? Why? How can you relax the method to deal with this?

Now adapt the mask to try and improve the matching. Show the adapted mask, and the modified result. Comment on the limitations of this approach (consider scale, rotation, sample-set).

5. Put together a chain of operations (a "processing pipeline") in order to automatically extract the in-focus areas of the image `bokeh_purpleflowers.jpg` (available on Absalon). To achieve this, you will need to generate a mask image which can then be used to show only those pixels that are inside the mask.

   To do this you may find SOME (not necessarily all) of the following ideas useful:

   - Convert to greyscale
   - Examine the different colour channels or do a convertion to an appropriate color space.
   - Blur the image. (Use a gaussian filter. Parameters to vary: filter size, sigma), then calculate the difference image.
   - Perform edge detection using the morphological gradient. (Parameters to vary: structuring element).
   - Dilation followed by erosion.
   - Filling by "thickening"

   Note that you can repeat tasks at different stages, and you can branch processing into multiple streams and then merge the results. You will probably need 5 - 10 stages in total.

   Finally, apply your mask to the image. (E.g. Use the alpha channel to mask out the areas outside of the mask)

   Show the result of each step in your pipeline to generate the mask. Report and explain your parameter choice (choice of SE, size of SE, filter parameters etc) for each step.

6. Using mathematical morphology, can you count how much money there is in the image `money_bin.jpg` on Absalon (see below for labeled version), *without* performing segmentation? You are also given the additional information of the labeled image `money_labels.png`. Please include both a description of what you did, and the final amount.

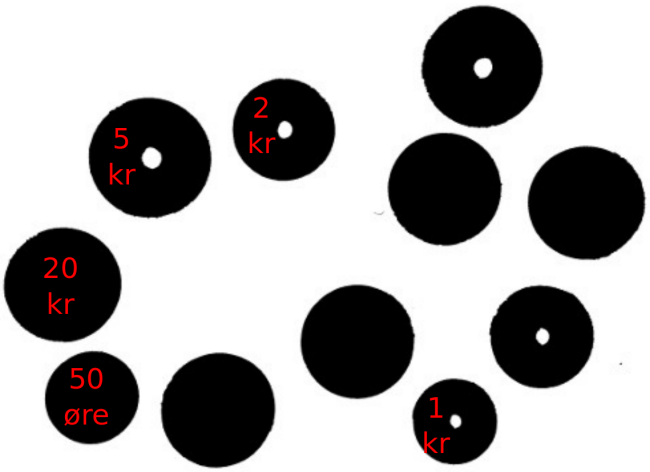


Figure 2: Given the underlying binary image `money_bin.jpg` and the labels seen in this figure, can you count the money without performing segmentation?