KOGW-PM-KNP: Edge detection Questions - Gaussian filtering

July 6, 2016

Complete the tasks using python. Your answers should include the code you wrote and written answers to the questions with relevant output images.

Task 1. Gaussian filtering of an image

1. Using the given equation for a 2D Gaussian filter, plot the output of a filter-convolved image with an appropriate sigma σ value.

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

   Hint: Use PIL.Image.open to import your image, and from scipy.signal use the convolve2d (with mode='same') function for the convolution.

2. What type of filter results from a high σ value?

3. Can you think of a way of extracting high frequency components from an image using only a low frequency filtered image? Plot such an image and comment on its quality

Task 2. Difference of Gaussian filtering of an image

1. Create a Difference of Gaussian (DOG) filter by applying two different sigma valued Gaussian filters to the original image and then subtracting the two outputs from each other.

2. Try to find optimum sigma values for detecting edges in an image.

3. Comment on differences between this filter and the first.

Task 3. Hybrid Images

Using the Gaussian filter from Q1.1 and the knowledge of spatial frequencies from Q1.3, extract low spatial frequency information from the image of Albert Einstein provided and add this to the high spatial frequency information of the Marilyn Monroe.

Experiment with two different sigma value Gaussian filters, until up close the combined image looks like Monroe but from afar like Einstein.

Hint: You may find the cv2 library useful for import and resizing the image

Task 4. Phase and Fourier Transforms

1. Using the Fourier transform split the Einstein and Monroe images into their component magnitude and phase information

2. Recombine the components for each image using the reconstruction equation for one image below:

   $Recontruction = M(ft1) * cos(\phi(ft2)) + M(ft1) * sin(\phi(ft2)))i$
   where M is magnitude and ϕ is phase.

   Hint: Use numpy.fft.fftn for the Fourier transform.