

Imaging Fundamentals

Prof. Brian Frost

ECE-435-1, Fall 2021

Introduction

In class, we discussed some fundamental concepts in image processing. In this project, you will apply this newfound knowledge to a real volume scan – a CT scan of the thorax made available by Mulrecon. It is the first scan on their webpage:

<http://www.castlemountain.dk/atlas/index.php?page=mulrecon>.

The concepts tested here will be: 1) Noise, 2) Blurring, 3) Filtering, and 4) Edge detection. You have 2 weeks to complete this project. It should be submitted as a zip file emailed to me (b.frost@columbia.edu) containing the MATLAB or Python files and the documentation as a PDF. The documentation should contain the relevant figures and analyses asked for in each section below, and doesn't need to be too fancy.

MATLAB functions that may be useful: `rgb2gray`, `squeeze`, `imfilter`, `imgaussfilt`, `medfilt2`, `edge`.

I. Slicing

The stack is given in the form of 237 JPG images. Each image is a slice through the thorax at a fixed z . Call the horizontal (left-to-right) x , the vertical (up-to-down) y and the direction of increasing image index z (this is a right-handed coordinate system). This format is clunky – you cannot naturally see other cross-sections of the scan (fixed x or fixed y) looking at these images.

Create a 3D array containing “the volume” – that is, a stack of the images so that the first index corresponds to the x pixel, the second to the y pixel and the third to the z pixel. The volume should be grayscale. If you have done this correctly, the slice indexed by $(:,:,p)$ should correspond to the p^{th} jpeg image.

In the documentation, present one slice in each of the three dimensions near the middle of the subject. Label the axes with the corresponding dimensions, and label the pixel at which the slice is taken (for example, title a figure $x = 215$). Comment on the units of the axes, how they relate to meters, and what that means about the aspect ratio of your images.

II. Salt-N-Pepa

Salt and pepper noise or speckle noise is common in CT. This means that noise appears in the form of spurious very high- or very low-intensity pixels. In small neighborhoods, this could skew the mean downward or upward due to these statistical outliers. However, the spurious pixels should not affect the median.

In larger neighborhood, the statistical outliers probably “average out.” We might intuitively expect the mean and the median to behave similarly in large neighborhoods. However, filtering

over large neighborhoods has a larger smoothing effect, which is not helpful if we want to recognize finer features.

For one of the slices you presented in section I, show the signal median-filtered and mean-filtered in a 3x3 and 5x5 neighborhood (4 images total). In the text, describe the effects of either filter, compare the median with the mean filter at each neighborhood size.

For many cross-sections, you may see that both filters are very similar. To exaggerate this effect further, you can insert salt and pepper noise yourself! Write a function that replaces a pixel with a white pixel with probability p , a black pixel with probability p , or leaves the pixel unaffected with probability $1 - 2p$. Show the effects of the mean and median filter with neighborhood size 3x3 for three reasonable p values (in your report, show both the noisy and filtered images, so 6 images total), and describe the difference.

III. Gauss' Wrath

We will explore the effects of Gaussian noise, Gaussian filtering and Gaussian models of blurring, one by one.

Gaussian Noise – Gaussian noise is present in most measurements due in part to thermal effects on the hardware side. Traditionally, local averaging through Gaussian filters is the best way to deal with this sort of noise. Gaussian filters are low-pass filters, so they have undesirable smoothing effects if they are made too large. This is very similar to the mean/median filters above. Add Gaussian noise to the signal with different standard deviations (pick two reasonable but different values). Improve the signal quality by using Gaussian *filters* on these images. For these filters, just pick the standard deviation which gives you the qualitatively nicest results. Report these values and how they relate to the noise standard deviation, and discuss if or why this makes sense.

Gaussian Blurring – Blurring in imaging is usually caused by the image being out of focus. However, all images are out of focus (by the diffraction limit), so there will always be some blurring which we model as convolution with the PSF (impulse response) of the system. The PSF is usually an Airy pattern in 2D or 3D, but this is well-modeled by a multidimensional Gaussian pattern.

Assume that the CT scan is not a scan at all, but instead a perfect representation of our sample. That is, the CT intensity map is precisely the reflectivity of the sample. We will now model the imaging of this sample by a device with an anisotropic point spread function.

Say the resolution as quantified by the full width half max (FWHM) value of the PSF is δx in the x and y directions (lateral) and δz in the z direction (axial). Model this PSF by a three-dimensional Gaussian pattern in space – what is the form of this PSF (normalized so that it integrates to one)? It might help to consider the one-dimensional Gaussians in each direction – what standard deviation would give them the correct full-width value at half of the maximum? Once you have those three standard deviations you can write the formula quickly up to the normalization constant, which you can simply compute in MATLAB. Present this 3D formula in the documentation.

For each slice you presented in part I, present the slice blurred in the proper way with $\delta x = 2$ mm, $\delta z = 1$ mm. This is non-trivial, as the pixels in these images are rectangular as well. You may want to use the 2D functional form applied to a meshgrid, and one of convolution or an FFT multiplication equivalent. Compare these to the original scans.

III. The Plural of Foramen is Apparently Foramina

The spaces between your vertebrae are called the intervertebral foramina. Look at slice $y = 180$ – you can make out some features of the subject's spine, including several intervertebral disks and foramina. The disks appear as subtler intensity variations, with the foramina being more starkly contrasted from the vertebrae.

Use MATLAB's built-in Canny edge finder (you should recall this method from class) on the raw slice. You should be able to make out the intervertebral disks and foramina as well as many other edges. Do you see an issue with the result here?

Use this Canny edge detection method on four variations on the slice: a) a 3x3 median-filter is applied, b) the object is imaged by the system as described in part II (fancy 2D Gaussian filter), c) the image is impacted by one of your Gaussian noise paradigms from part II, and d) the image is filtered by a Gaussian filter with standard deviation 3 pixels in both directions.

Describe the effects on the ability to discriminate the intervertebral foramina and disks. What about other seemingly spurious edges?

Finally, for your own sake, look at how pretty the Canny edge detected images are for the z-slices!