Crispin Bernier

The Matlab script is the main.m file in the Matlab folder. Running this script produces the images that are also included in that folder. The C++ code is in the C++ folder. This code is compiled with 'make' and executed with 'make run' and will also generate the images that are included in the folder.

Section 1.

Please assess your skill level in the following areas using a 1-10 scale where a 10 is exceptionally masterful and 1 is no knowledge of the topic. Provide a rationale for each item if useful:

- Signal processing
  6 – I am currently learning signals and systems in my electronics class, including convolution integrals, Fourier Series, and Fourier Transforms.

- Image processing
  6 – I am currently taking a computer vision course where the focus so far has been on image analysis.

- C/C++ Programming
  7 – I took a course last semester in C++ called program and data representation.

- Python Programming
  5 – I used Python last summer to write small programs for a robot.

- MATLAB Programming
  8 – I am using MATLAB to implement many image analysis algorithms as well as for signal processing.

- GPU Programming
  1

- Ultrasound Imaging
  1

Section 2.

Review the filter described by Eqn (1) in the Foroughi et al paper. The filter in Eqn (1) uses shadowing characteristics in the ultrasound image to enhance bone surface contrast. In the attached 'TestImages' directory, there is code that loads sample ultrasound image data of a spine cross-section. The code then scan converts the image from polar to rectilinear coordinates for display. We would like you to create your own Matlab code that implements an optimized version of a shadow-based algorithm, such as the one described in the Foroughi et al paper.

Note: Do not strictly implement Eqn (1) in Foroughi et al as it most likely contains errors. Instead, use the concepts described in the paper to implement your own method that uses shadowing to improve bone-to-tissue delineation.

Please perform the following tasks and summarize results in a single word document or pdf-based report:

1. Create a Matlab script that implements a shadow-based bone enhancement algorithm, applies the algorithm to the sample data provided, and creates images of the pre- and post- filtered image.

2. Provide a concise one paragraph description of your algorithm and justify your approach. Do your results qualitatively match those of Foroughi? Why did you pursue this particular algorithm design and implementation?

3. Translate the bone enhancement section of your Matlab script to C/C++. Create a test to verify that the output of your C/C++ version matches the result produced by MATLAB. Why did you choose the test that you implemented?

1.

The Matlab algorithm script is the main.m file located in the Matlab folder. Running this script generates the 8 images that are also in this folder and that show the entire segmentation process.

2.

The images in the Matlab folder document the process of the algorithm. These images are in order sequentially with how the algorithm runs. My approach was very similar to that of Foroughi however I chose a different shadow enhancement technique. My approach was to take the difference of the weighted sums above and below each pixel. These sums are weighted by the Gaussian distribution according to their proximity to the current pixel, which keeps the detection of shadows more local rather then across the entire image. I felt this algorithm would highlight shadows by giving a large value where there are sharp intensity transitions in the image rather then just detecting places where the average intensity below a pixel is small. I was unable to implement the thresholding function for drawing the bone curves that Foroughi describes, however I implemented a simpler thresholding function that produces similar results in outlining the bones.

3.

My test for comparing the C++ and Matlab outputs was to take the average difference of the probability intensity images from the two outputs, these are images 6b and 6 in the Matlab folder. Each intensity pixel goes from 0 to 255, and the average difference was around 9.25 which gives about 3.6 error per pixel on average between the two images. Using this error percentage and qualitatively looking at the two images shows they produce relatively the same result. I chose to test the probability images because I felt this image was the most important output of my algorithm. This image shows where the bones are most likely located and running the same algorithm in C++ and Matlab should produce the same values at each pixel. The thresholding step relies on these probability images to determine where the bones are in the original image.