Comparing Equivalent Diameters of CT Phantom Image (Known diameter), from the algorithms programmed in Mathematica, MATLAB, and Python.

Introduction

This document is the same as listed in the Mathematica Code section. It has been changed to MATLAB code. The version of MATLAB is R2018**.** To complete execution of this code, you will need to place a DICOM image of a CT Phantom in the directory listed in the code. Phantoms come with different diameters, so you will also have to measure the diameter of your phantom.

This is not an update of the previously uploaded program. The purpose of this computable document is to provide comparisons of equivalent diameters between 3 similar algorithms to identify body pixels for calculation of SSDE. The results of the comparison are listed in the summary. I have tested 3 programs using an algorithm programmed in 3 computer languages, Mathematica, MATLAB, and Python. Because I have access to a CT Phantom image of known diameter, I have compared the automated measurement diameter among the 3 programs. The MATLAB and python versions should be uploaded shortly after this computable document.

Disclaimer: I am not a professional programmer. I have only rudimentary skills in the three computer languages. These programs are not examples of optimum algorithms, nor correct style. Much of the programming is based on publicly available tutorials in the different languages. Feel free to make suggestions for improvement in style or optimization.

I began programming the task of automated Size Specific Dose Equivalent (SSDE) calculation in Mathematica. I chose parameters for thresholds based on calibrations using the algorithm with the CT phantom. The Mathematica program was then verified against actual clinical cases (all completed through an Institutional Review Board and adhering to HIPPA guidelines). Based on the results, the intended users (medical physicists) could feel comfortable using the program to demonstrate to CT facilities how much better SSDE is for controlling CT radiation exposure. The programs are not qualified to be used for clinical decisions. They may be used to educate facilities on the advantages of SSDE radiation dose parameter versus CTDIvol.

The experimental process required manually measuring diameters from CT images. The automated measurement was then compared to the standard method of measuring cross sectional diameters for calculation of SSDE. I do not have the interest to repeat the process for the slight variation between algorithms programmed in the 3 languages.

The rest of the introduction repeats most of the introduction included with the original program documentation. You may skip to the results.

Mathematica, MATLAB, and Python are more likely to be familiar to medical physicists, who are the main professionals who might find these programs useful. I have written these programs for use with publicly available CT data from the National Institutes of Health. The publicly available data is from older CT machines. It does not have a slice CTDIvol tag to use for more accurate calculation of SSDE. It does not have a dose summary page to indicate the study average CTDIvol. Consequently, in all of the programs I have simply included CTDIvol as a constant.

None of these programs are suitable for clinical use. They would have to be submitted to the FDA for certification for actual clinical use. These programs may help others develop their own programs for calculation of SSDE. The programs could be used for demonstration purposes. Such demonstrations might convince CT facilities to begin a program for monitoring and optimizing CT radiation exposure using SSDE. Because the image processing algorithms are well established from nearly 50 years ago, I think the use of the algorithms should be free of worries of copyright or patent infringement (not a guarantee).

The goal of the use of the algorithms are to separate those pixels in the CT image which represent the patient's body, from external structures such as the table or clothing. Based on the number and area of body pixels, the size of the patient can be related to the size of standardized CT phantoms used to calculate Size Specific Dose Estimates. The Mathematica algorithm here relies on morphologic image processing as described in Gonzalez text book on digital image processing. The MATLAB and Python programs rely on publicly available examples on the WWW, many of which are included in the documentation of the programming languages, or freely available libraries (such as Scikit-Image). The underlying mathematics has more similarities than differences among the algorithms. More computationally intensive algorithms often have the advantage of greater reliability with a larger selection of images for processing. However, the underlying assumptions of SSDE do not suggest that the size adjustment of dose estimate based on measurements of a specific type of phantom are universally applicable. For example, a CT image of the abdomen obtained with the patient with the arms at the side would represent a significant variation from the standard phantom (based on arms positioned above the head). Reliability for excluding the arms from the calculation of the body diameter does not indicate a better program for calculating SSDE. I would argue that those cases are exceptions (often necessary due to clinical constraints), and should be excluded from monitoring radiation dose by calculation of SSDE.

The simplest algorithm has the advantages that:
1. It is easier to read and understand.
2. It is easier to review and insure the lack of malware.
3. It is usually more predictable in what images may cause failure.

Gonzalez textbook algorithms have been made available in Mathematica, MATLAB, Python, C++, and other languages. The differences in algorithms (and compilers, interpreters, hardware) result in differences in computational speed.

Ultimately, image characteristics, and the goal of computation determine what are the most suitable algorithms for a particular application. If your planned use is for commercial or clinical application, then you should hire professional coders from the companies that provide the examples in the various languages.

MATLAB Code

```
cd ‘C:\Users\Public\’
filelist = dir(‘*.dcm’);

area1=0.0;sum1=0;
for file = filelist’
  sum1=0;
  I = dicomread(file.name);
  figure, imshow(imadjust(I)), title(‘original image’);
  bI=imbinarize(I);
  fbI=imfill(bI,’holes’);
  cc=bwconncomp(fbI);
  numPixels = cellfun(@numel,cc.PixelIdxList);
  if numPixels > 1000
    [biggest,idx] = max(numPixels);
    grain=false(size(I));
```

```
    grain(cc.PixelIdxList{idx})=true;
    figure,imshow(grain), title('filled binary image');
    metainfo = dicominfo(file.name);
    pixelarea=metainfo.PixelSpacing(1)*metainfo.PixelSpacing(2);
    area1=bwarea(grain)*pixelarea;
    test1=sum(grain(:))*pixelarea;
    equivdiameter=(sqrt(4*area1/pi))/10;
  end
  rescale_intercept= single(metainfo.RescaleIntercept);
  hadj=@(x) ((single(x)+rescale_intercept)/1000.) + 1.;
  masked_attens=arrayfun(hadj,I);
  masked_attens(~grain)=0;
  tot_pixel_atten_area =sum(masked_attens(:));
  area2=tot_pixel_atten_area*pixelarea;
  equiv_pix_attn_diam=(sqrt(4*area2/pi))/10;
  figure, imshow(imadjust(masked_attens)),title('Masked image');
end
```

Summary

This document lists the equivalent diameter determined from automated calculation from a CT phantom image of known diameter (20 cm). This is not a test of the accuracy among the languages. Variations may occur because of different thresholds, differences in algorithms (the image hole filling function may be a source of difference), or simply my programming ability. The differences are inconsequential. The SSDE is based on measurements of radiation exposure in phantoms of different sizes. The phantoms are only a close approximation of human anatomy. In fact, reporting of the SSDE of values 10 or higher is recommended to be rounded to the nearest whole number.

Mathematica: 20.0189 cm
MatLab: 20.0659 cm
Python: 20.0572 cm

The diameter of the phantom is a known value. Calculating the equivalent diameter using the attenuation values to adjust to the attenuation of water does not have a known value from this phantom. The acrylic walls of the phantom have a higher attenuation than water. The air within the phantom has a lower attenuation than water. Consequently, I did not list the water equivalent diameter in this comparison. It is calculated in each example.