MATLAB PROJECT

Digital Image Processing

Dr. K.K Gupta

Bone Segmentation from an MRI

By:

Shrihari Viswanath

2016A8PS0396P

PROJECT MOTIVATION

I have chosen this project because of my inclination towards biomedical engineering. The various applications of image processing in helping us diagnose various diseases and other health issues. Some experience in a niche field like this will go a long way. Also, a project with a clear aim and a goal is something to strive towards.

OBJECTIVE

The main objective of this project is to segment, i.e. separate the bone from the tissues and other visible regions/things in the MRI, be it the MRI of the knee, brain, hip, etc. It should be an automatic process for making the lives of doctors and hospitals easier by enabling us to understand and to read MRI's better. Using this MRI's can be used for scanning and testing our bones from every nook and cranny instead of an x-ray.

ALGORITHM AND FLOW CHART

DENOISING: The use of histogram equalisation for controlling contrast for effective thresholding is a main part of the de-noising and pre-processing. Because the backgrounds of MRI scans are so noisy and contain significant lighting artifacts, it is important to filter out the background before proceeding with other image processing techniques. Otherwise, the background compromises the effectiveness of these methods. Hence, filtering using a median filter to remove noise is a good method. Both have been implemented for an optimal result.

METHOD 1: This involves edge detection using Sobel masks. The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. Then the image is thresholded at a value of 200 to binaries the image. Finally a border clearing function is applied to get rid of the boundary (the skin/surface of the MRI). This gives us the edges of the bones along with a little from the tissue as perfect segmentation without machine learning application is very difficult to implement.

METHOD 2: A simpler implementation for automatic segmentation might use adaptive Otsu thresholding. Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum. In theory, this would work well to separate the bright trabecular tissue or the dark cortical tissue of the bone from the other tissues. A fudge factor has been introduced which can be inputted by the used, the thresholding value will be scaled by that amount if it isn't satisfactory.

Both these methods given above have been implemented and compared below in this very report.

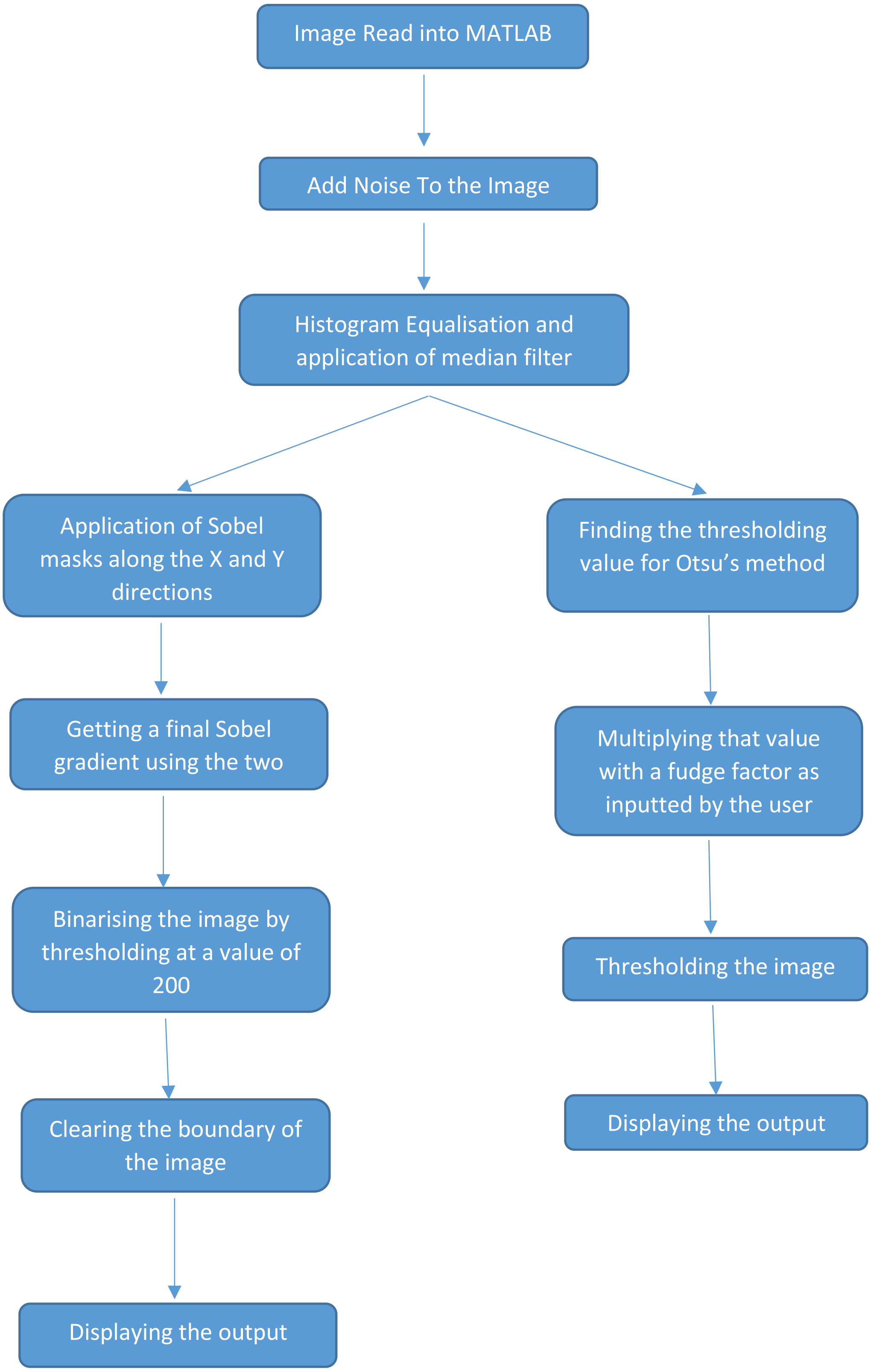
Image Read into MATLAB
Add Noise To the Image
Histogram Equalisation and application of median filter
Application of Sobel masks along the X and Y directions
Getting a final Sobel gradient using the two
Binarising the image by thresholding at a value of 200
Clearing the boundary of the image
Displaying the output
Finding the thresholding value for Otsu's method
Multiplying that value with a fudge factor as inputted by the user
Thresholding the image
Displaying the output

RESULTS AND OBSERVATIONS

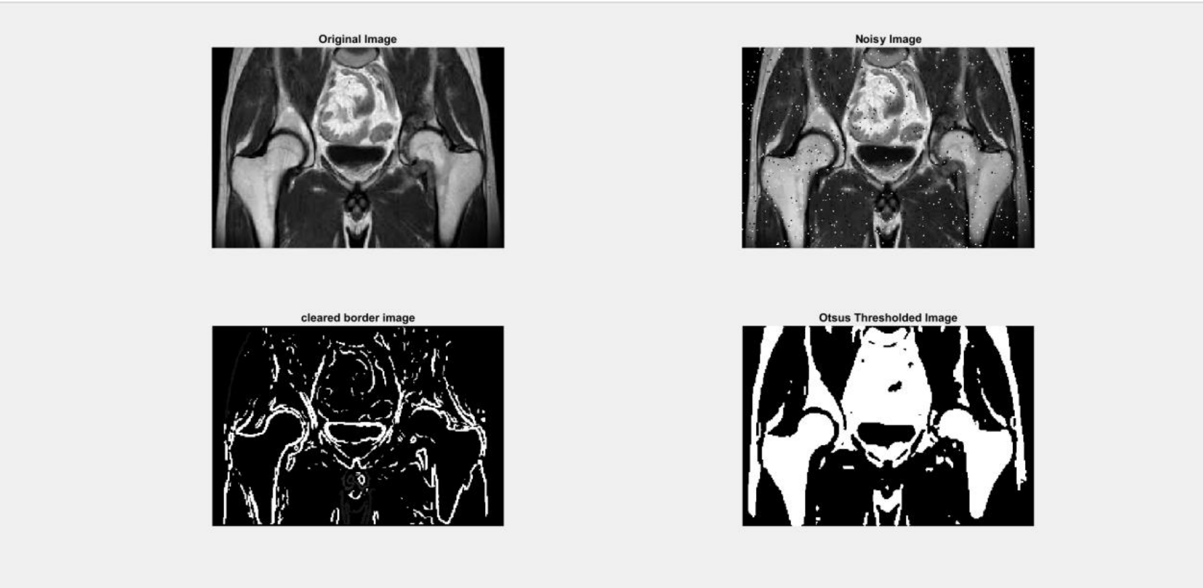

Fig 1. MRI of the hip from https://www.researchgate.net/figure/MRI-of-the-hip-the-synovium-is-diffuse-at-the-hip-with-bone-invasion-Synovium-appears_fig3_304617089 and its processed images

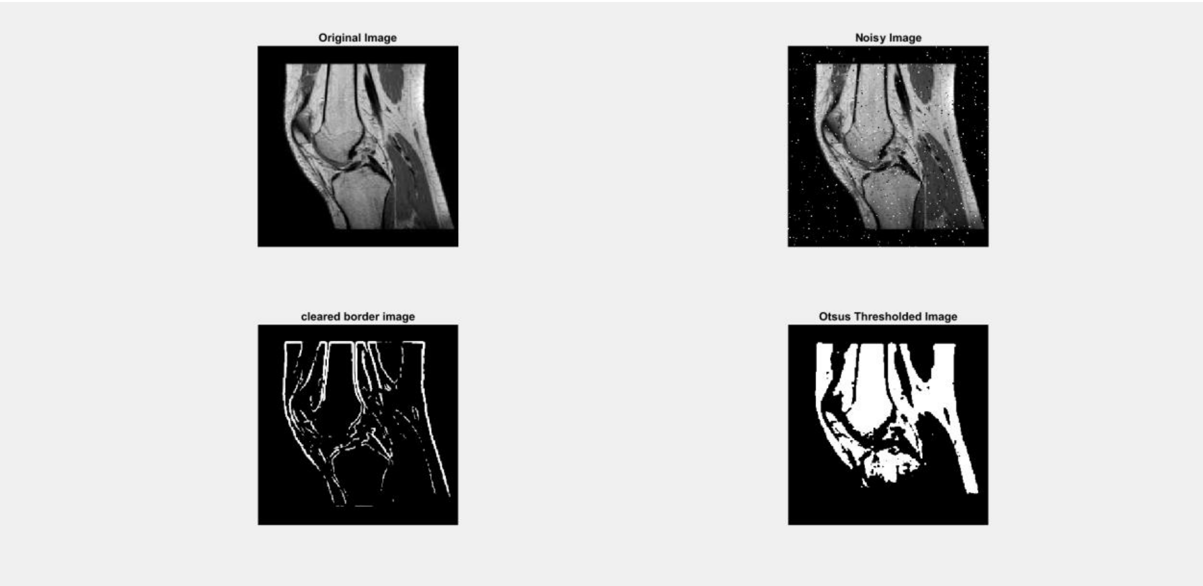

Fig 2. MRI of the knee from https://radiopaedia.org/cases/normal-knee-mri and its processed images

In the various images there are four outputs which are respectively the original image, the noisy image, the output of method 1 (Sobel edge detection) and the output of method 2 (Otsu's thresholding). Similarly we have performed the same for the brain and the back as shown below.

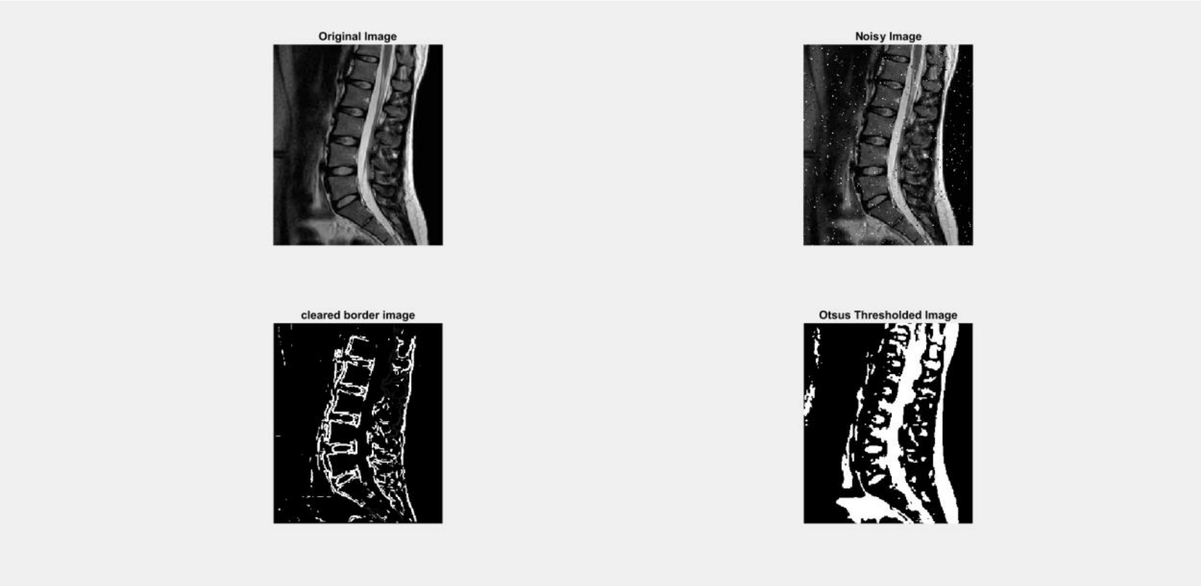

Fig 3. MRI of the lower back from *https://mayfieldclinic.com/pe-mri.htm* and its processed images

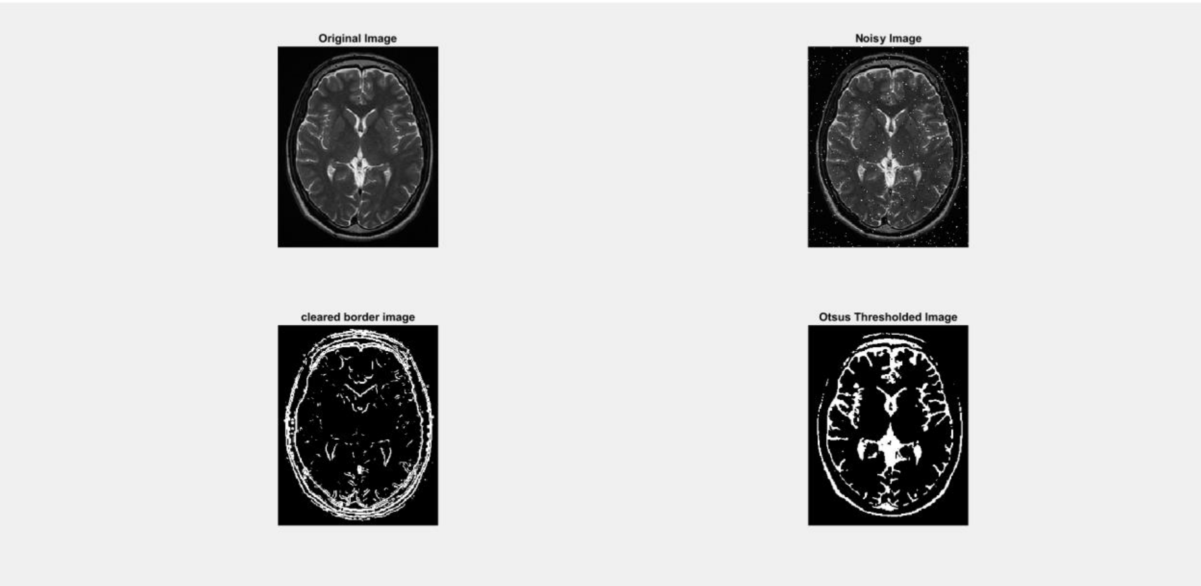

Fig 4. MRI of the brain from *https://en.wikipedia.org/wiki/Magnetic_resonance_imaging_of_the_brain* and its processed images

We can clearly see that both the methods are giving us decent results and are able to segment the bones roughly. The issue with Otsu's thresholding is that in areas where the dark cortical tissue is lighter, it gets labelled as a bright area, thus connecting the bright trabecular and muscle tissues. Also, because the areas of connection between bone and muscle tissue are so large at the ends of the bones, they cannot be easily split with operations such as morphological opening.

The Sobel segmentation is cleaner than adaptive Otsu segmentation, but like the Otsu method, Sobel connects the bone tissue with muscle tissue at the ends of the bones where the cortical layer is thin.

The noise removal using the median filter and also the contrast enhancement using histogram equalisation are working well. They are giving optimal results without losing too much detail in the image. Also, in Sobel's edge detection there are fake and weak edges that create a lot of confusion for the person to understand. Using dilation and other morphological operations are also giving unreal and wrong outputs.

REFERENCES

1. Sharma N, Aggarwal LM. Automated medical image segmentation techniques.
2. Andrea Aprovitola, Luigi Gallo. Knee Bone Segmentation from MRI: A classification and literary review.
3. Toki Migimatsu. Automatic MRI Bone Segmentation.

APPENDIX

1. ALTERNATIVE APPROACH :

   Snakes or active contour model. The snakes model is popular in computer vision, and snakes are widely used in applications like object tracking, shape recognition, segmentation, edge detection and stereo matching. A snake is an energy minimizing, deformable spline influenced by constraint and image forces that pull it towards object contours and internal forces that resist deformation. Snakes may be understood as a special case of the general technique of matching a deformable model to an image by means of energy minimization. Currently trying out the implementation for this method.

2. IMPLEMENTED CODE:

```
close all;
clc;

%add name of file here
a=imread('brain.jpg');
Z=rgb2gray(a);
X=imnoise(Z,'salt & pepper',0.02);

%hist eq and noise removal using median filter
Y=histeq(X);
B=medfilt2(Y);
I=B;

subplot(2,2,1)
imshow(a);title('Original Image')
subplot(2,2,2)
```

```
imshow(X);title('Noisy Image');
pause(2)
I=double(B);

%Sobel Mask
for i=1:size(I,1)-2
for j=1:size(I,2)-2
%Sobel mask for x-direction:
mx=((2*I(i+2,j+1)+I(i+2,j)+I(i+2,j+2))-(2*I(i,j+1)+I(i,j)+I(i,j+2)));
%Sobel mask for y-direction:
my=((2*I(i+1,j+2)+I(i,j+2)+I(i+2,j+2))-(2*I(i+1,j)+I(i,j)+I(i+2,j)));
B(i,j)=sqrt(mx.^2+my.^2);
end
end

%Define a threshold value
Thresh=200;
B=max(B,Thresh);
B(B==round(Thresh))=0;
B=uint8(B);

%Clearing the border for a binary image
BWnobord = imclearborder(B, 4);
subplot(2,2,3), imshow(BWnobord), title('cleared border image');

level = graythresh(I);
%input fudge factor f
f=input('Input the fudge factor by which you want to scale the threshold. Preferrably under 0.005, keep altering this for desired and optimised results');
BW = imbinarize(f*I,level);
subplot(2,2,4), imshow(BW), title('Otsus Thresholded Image');
```
