# Detecting contours of human organs in CT images using the Canny edge detector

**Author**: Mateo Kalem, 63170135

**Abstract**

In this paper we will represent the results of our seminar work, where we used the Canny edge detector to detect contours in CT images of the CTMRI database [1].

**Introduction**

Edge detection is a commonly used operation in the processing of images. An edge is usually defined as a discontinuity in gray level values. So it is safe to assume, that an edge is the boundary between the image and it's background. During our lectures we mentioned some edge detectors, such as the Marr-Hildreth and Canny edge detectors. For the purpose of our seminar work, we chose the task to implement the Canny edge detector on a selected sample of images from the CTMRI database.

**Methods**

In this section we will describe the methods we used to complete our seminar work.

**1. Edge detection**

Edge detection is a process which consists of three fundamental steps:

- Image smoothing for noise reduction.
- Detection of edge points, this is a local operation that extracts from an image all points that are potential candidates to become edge points
- Edge localization, selects from the candidate edge points only the points that are true members of the set of points comprising an edge.

**2. Canny edge detector**

In our seminar work we used the Canny edge detector to detect the contours in images from the CTMRI database. The basic steps of this detector are:

- Smooth the input image with a Gaussian filter.
- Compute the gradient magnitude and angle images.
- Apply non-maximum suppression to the gradient magnitude image.
- Use double (hysteresis) thresholding and connectivity analysis to detect and link edges.

**3. Gaussian filter**

The gaussian filter is a smoothing spatial filter. It's aim is to smooth sharp peaks by rearranging the intensities in a given image. One way to smooth the images is to calculate the averages of surroundings for every pixel and replace that pixel with the averaged value. The Gaussian filter works on a similar principle, except that the average is weighted. We calculate the Gaussian filter in 2 dimensions as follows:

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

### 4. Gradient magnitude and direction (angle)

We calculate the gradient and angle from the first derivative of a Gaussian function. Let f(x, y) denote the input image and G(x, y) the Gaussian function. The gradient is the tool of choice for finding edge strength and direction at location (x, y) of an image f(x, y) and it is defined as the vector [gx; gy]. We compute the gradient and direction (angle) as follows:

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x,y) = \tan^{-1}(\frac{g_y}{g_x})$$

Where gx and gy are any of filter mask pairs to compute the gradient. There are several masks to compute the gradient, during the lectures we mentioned the following:

- Roberts,
- Prewitt,
- Sobel.

In our assignment we used the Sobel mask [2], which is constructed as such:

| | | |
|---|---|---|
| −1 | −2 | −1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −2 | 0 | 2 |
| −1 | 0 | 1 |

Sobel

*Figure 1 Sobel's mask.*

### 5. Non-maximum suppression

Non-maximum suppresion is a scheme for thinning the ridges and works as follows:

- Find the direction dk that is closest to α(x,y) where d1, d2, d3, and d4 denote the four basic edge directions.
- If the value of M(x,y) is less than at least one of its two neighbors along dk, let gN(x,y) = 0, otherwise, let gN(x,y) = M(x,y).

### 6. Hysteresis thresholding

To solve the problem of “which edges are really edges and which are not” Canny uses the Hysteresis thresholding. In this, we set two thresholds TH and TL, where TH > TL. Hysteresis thresholding works as follows:

- Any edges with intensity greater than TH are the sure edges.
- Any edges with intensity less than TL are sure to be non-edges.
- The edges between TH and TL thresholds are classified as edges only if they are connected to a sure edge otherwise they are discarded.

In our seminar work we defined TH at 0.175 and TL at 0.075. To succesfully connect the edges, we used 8-connectivity.

## Results

In this section we will show the results of our script with the help of images generated by our script. In our script we defined sigma as 0.005 and we used Sobel's mask. We initially used Prewitt's mask [3], however, we weren't satisfied with the results. The images we tested were: 0090.png, 0070.png, 0013.png and 0005.png.

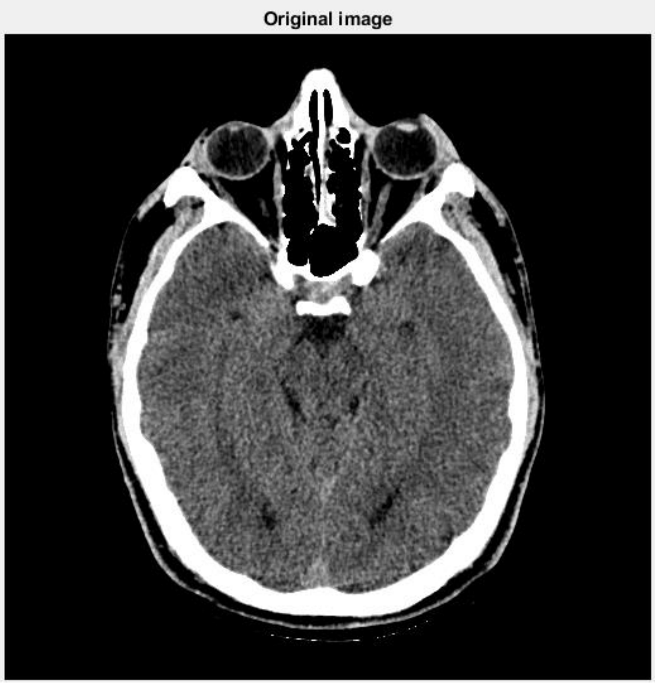


*Figure 2 Original image after loading it.*

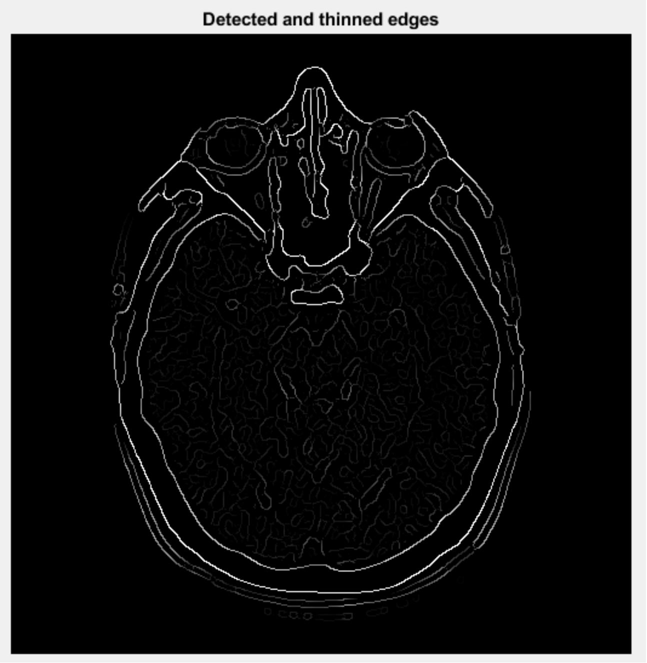


*Figure 3 Image after thinning the edges.*

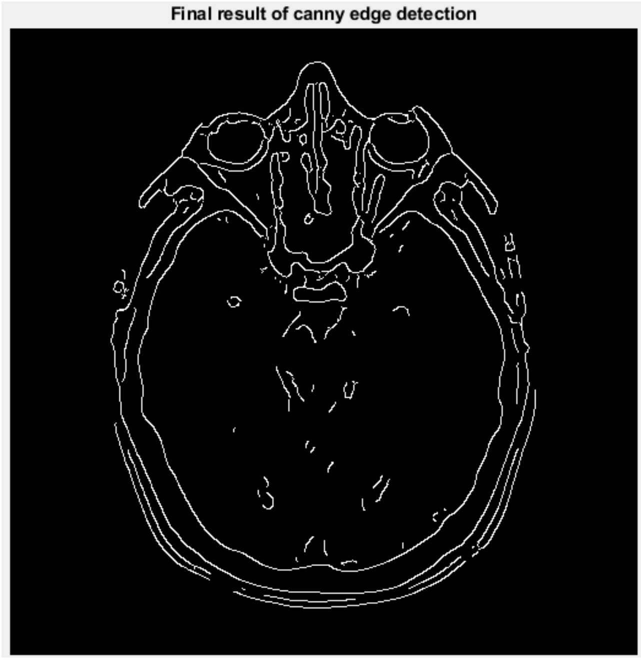


*Figure 4 Final result of canny edge.*

As we can see with these figures (we used 0090.png), our script has succesfully detected and connected the edges in the given image We printed out the images after loading the initial image, after edge thinning and after connecting the edges.. Here are the final results of the other images:

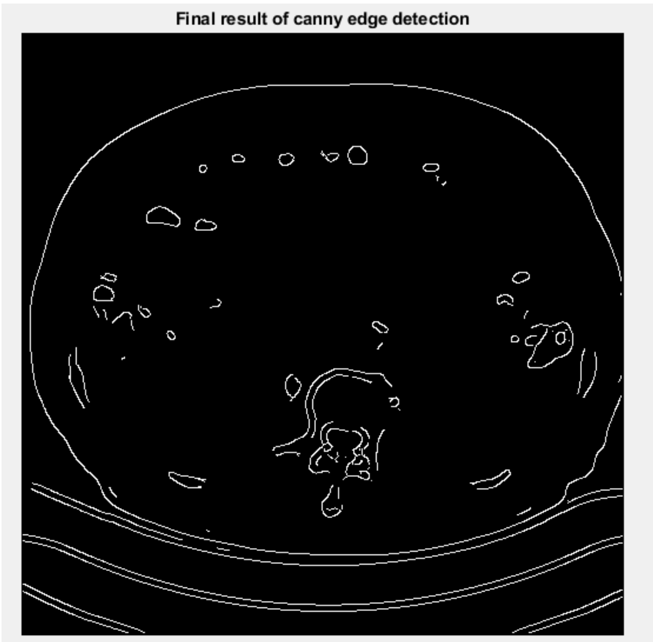


*Figure 5 0070.png.*

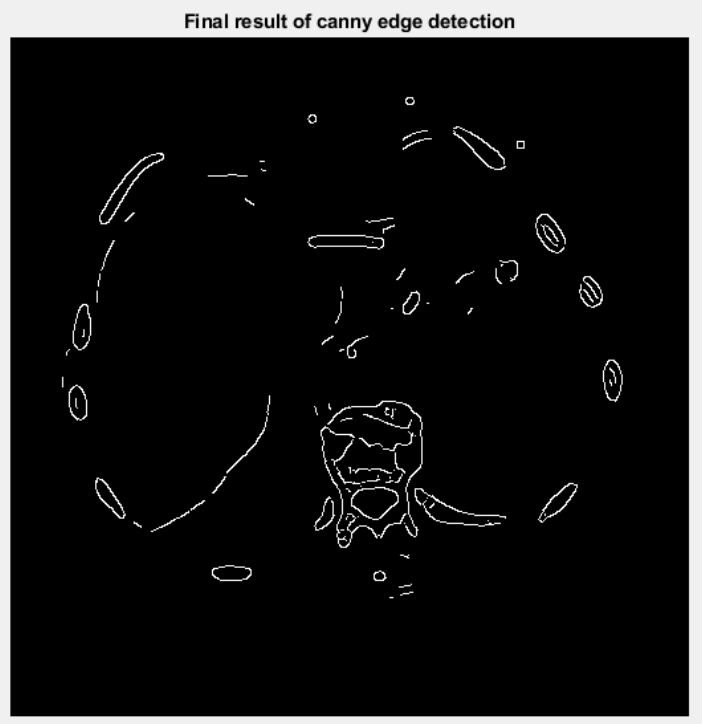


*Figure 6 0013.png.*

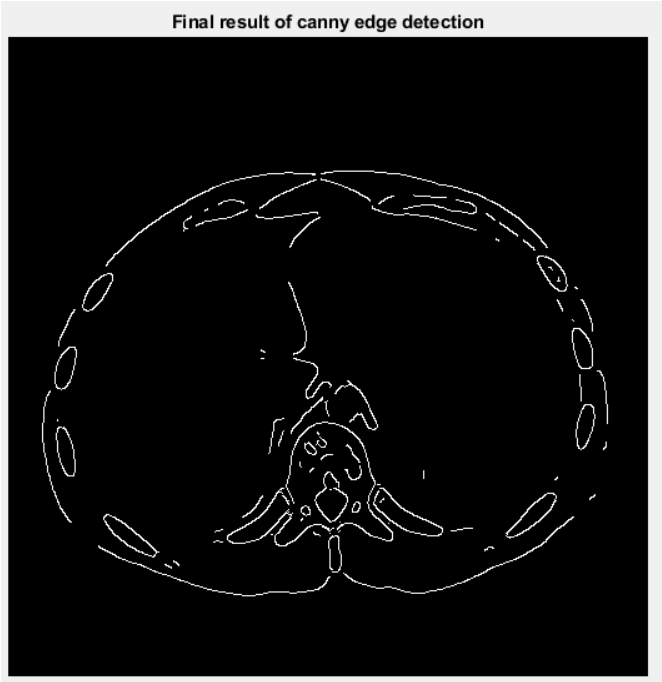


*Figure 7 0005.png.*

## Discussion

We are satisfied with the results of our Canny edge detector. The TH and TL values we used are generally good, however we could improve our script in a way, where we calibrate the TH and TL to each specific image, since our values (TH at 0.175 and TL at 0.075) are not guaranteed to work for every possible image in the CTMRI database. What we also could have done is implement our own function for the Gaussian filter, instead of using Matlab's built-in functions.

## References


[1] CTMRI DB. (18. January 2021). Available at: LBCSI: https://lbcsi.fri.uni-lj.si/OBSS/Data/CTMRI/

[2] How does matlab produce thin edges. (18. January 2021). Available at: Mathworks: https://www.mathworks.com/matlabcentral/answers/151729-how-does-matlab-produce-thin-edges

[3] Image edge detection using Prewitt operator. (18. January 2021). Available at: GeeksforGeeks: https://www.geeksforgeeks.org/matlab-image-edge-detection-using-prewitt-operator-from-scratch/