

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

Sara Bizjak

January 10, 2021

## Abstract

This paper represents results of detecting contours of human organs in the CT images from the DTMRI DB ([1]) using the Canny edge detector.

## 1 Introduction

Edge detection is one of the most commonly used operations in image analysis. An edge can be defined by a discontinuity in gray level values. We could say that an edge is the boundary between the object and the background. Some of the most popular edge detecting algorithms are Marr-Hildreth, Canny edge, the local threshold and boolean function based edge detector and others. For our analysis we used Canny edge detector.

## 2 Methodology

### 2.1 Terms

**The Canny edge detector** is an algorithm to detect contour on images, in our case, to detect contours of human organs in CT images. Basic steps of the Canny edge detection algorithm:

1. Smooth the input image with Gaussian filter
2. Compute the gradient magnitude and angle images
3. Apply nonmaxima suppression to the gradient magnitude image
4. Use double thresholding and connectivity analysis to detect and link edges

**Gaussian filter** is a smoothing spatial filter. It is used for rearranging intensities in image with the aim to smooth sharp peaks. One way to smooth the images is to calculate averages of surroundings for every pixel and replace that pixel with averaged value. The Gaussian filter works on similar principle, except that the average is weighted. That means that the pixel we are calculating value for has the largest weight (it is considered the most important in this

calculation) and the further we go from that pixel, the smaller impact other pixels' values have. The Gaussian function in 2-dimensions is defined as:

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

Since we are not in continuous space, we would like to get matrix of values (discrete values

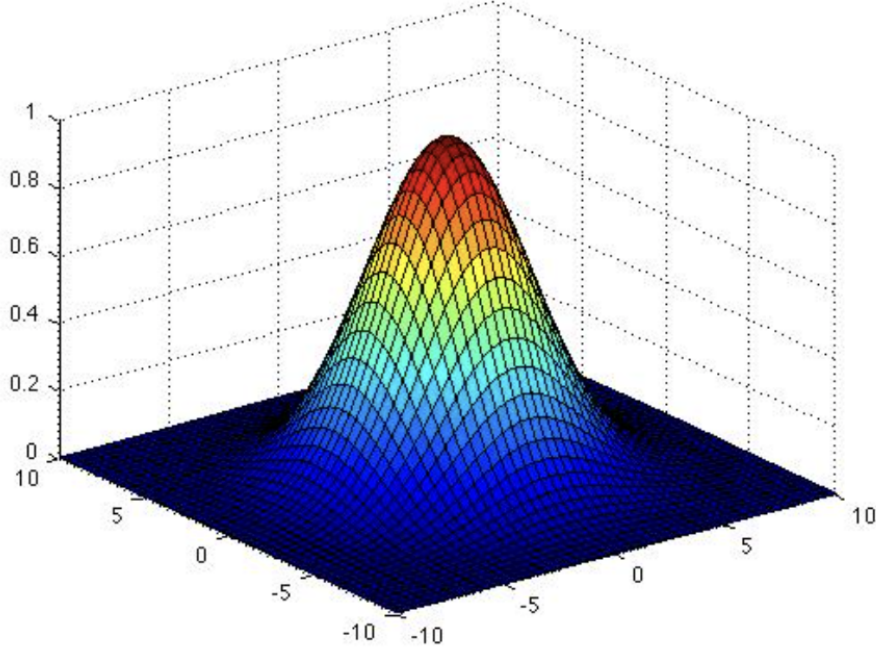


Figure 1: The two-dimensional Gaussian filter visualization.

of pixels) from the equation for the Gaussian function above. That matrix, we call it kernel, is then applied to every pixel to filter the whole image (the visualization of that matrix is similar to the figure 1). The kernel size (determined by parameter  $\sigma$ ) depends on the image size.

**Gradient magnitude and angle images** is calculated from the first derivate of a Gaussian function. Let  $f(x, y)$  denote the input image and  $G(x, y)$  denote 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  $[g_x; g_y]$  ( $g_x$  and  $g_y$  are the gradients along the horizontal axis and along the vertical axis, respectively, or along two perpendicular axes). The magnitude of the gradient vector is defined as the vector length while the direction  $\alpha(x, y)$  is given by the angle of the gradient vector with respect to the horizontal axis:

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

$$\alpha(x, y) = \text{tg}^{-1} \left( \frac{g_y(x, y)}{g_x(x, y)} \right)$$

There are several *masks* to compute the gradient. Let us mention the three from the lectures:

**Roberts:**

$$k_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$k_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

**Prewitt:**

$$k_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$k_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

**Sobel:**

$$k_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$k_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

**Non-maxima suppression** is a scheme for thinning the ridges and works as follows:

- Among the four basic regions (vertical, horizontal and diagonal) find the one closest to angle  $\alpha(x, y)$
- If the value of  $M(x, y)$  (magnitude) is less than at least one of its two neighbour along detected region, then the pixel at  $(x, y)$  is set to 0. Otherwise it is set to  $M(x, y)$

**Double thresholding** is a detector for determining "safe" edges depending on pixel intensity. That means that we use two thresholds, denote them as TH and TL (high and low). If the intensity of a pixel is higher than TH, we say that selected pixel is definitely located on the edge. Similarly, if the intensity of a pixel is lower than TL, we say that selected pixel is definitely *not* on the edge. If the pixel intensity drops somewhere in between TL and TH, we cannot determine whether it is on the edge or not. TH and TL are chosen as a proportion of the maximal intensity on the picture.

**Connectivity analysis** serves to connect the edges properly, that means that *a pixel close enough to an edge is also the same edge*. To say it more formally, we form longer edges using 8-connectivity from weak pixels to strong pixels, that is, for every pixel, we check whether the neighbours are a part of an edge. If there exists one, we connect the observed pixel to the same edge. To make the better results, we repeat the process several times and in several directions.

## 2.2 Usage

For our algorithm, we used Gaussian filter with parameter  $\sigma$  determined as

$$\sigma = (\text{minimal value of pixel on the image}) \cdot 0.005$$

and Prewitt's mask. Thresholds were set as:

- TH is set as 20% of maximal magnitude intensity,
- TL is set as 10% of maximal magnitude intensity.

For connecting edges we used window size 1 and made multiple repetitions.

### 3 Results

#### 3.1 Testing example

In this section are presented results of stages for one selected image from CTMRI DB. The stages of the process are written in the images' descriptions.

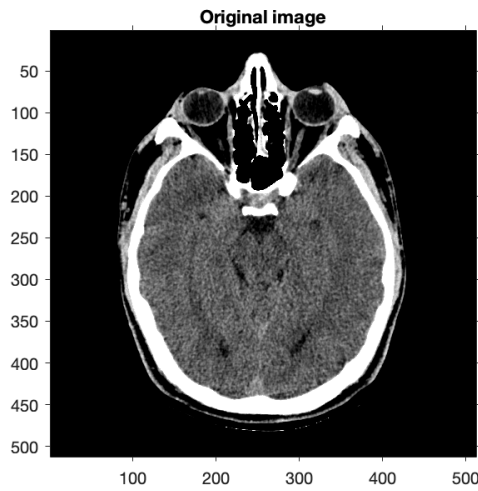


Figure 2: Original image.

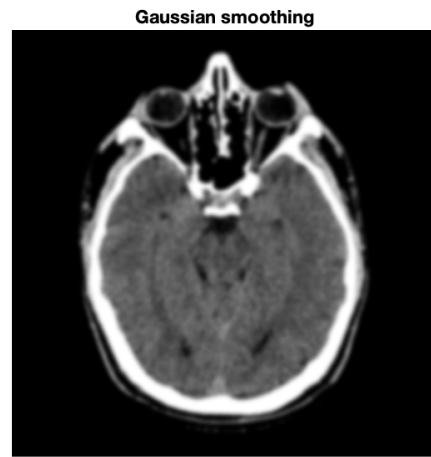


Figure 3: Image after Gaussian smoothing.



Figure 4: Horizontal gradient visualization.



Figure 5: Vertical gradient visualization.

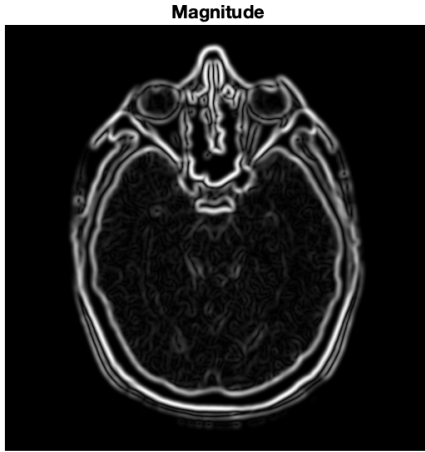


Figure 6: Magnitude visualization.

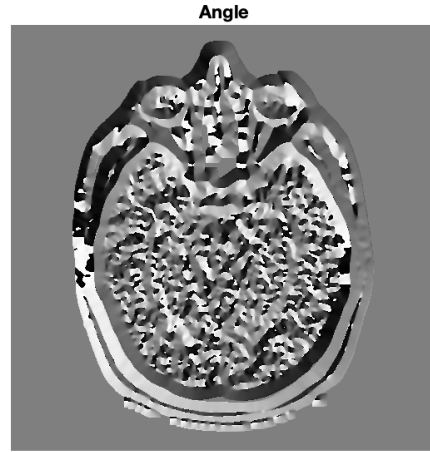


Figure 7: Angle visualization.

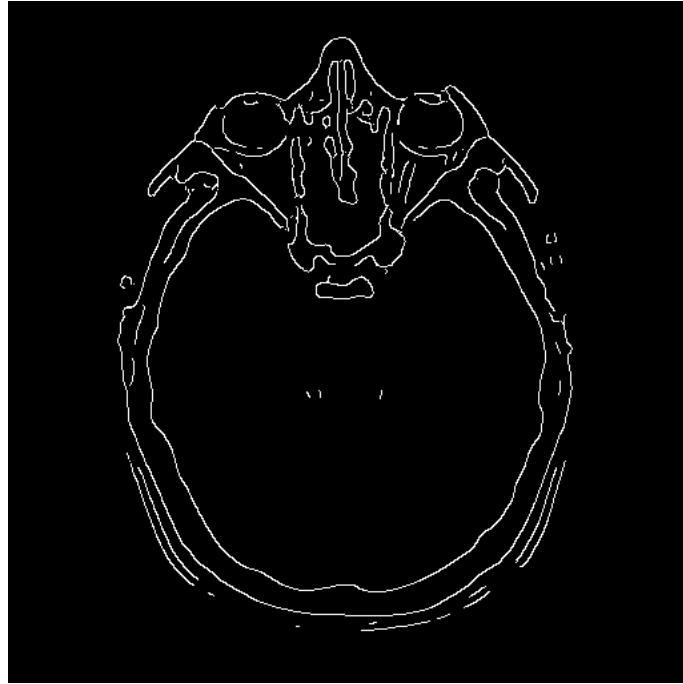


Figure 8: Image after canny edge detection.

For further analysis, or rather, for an attempt to improve the result of the algorithm, we try the Poisson smoothing filter (source [2]) that generates the Poisson noise from the data. The result of edge detecting can be better, but we must set the TH and TL by hand empirically. For the same image as above, the TH and TL were calculated as (we also increase TL so the relationship between TH and TL remains the same):

- TH is set as 28% of maximal magnitude intensity,
- TL is set as 18% of maximal magnitude intensity.

Take a look at the image after canny edge detection with Poisson smoothing filter:

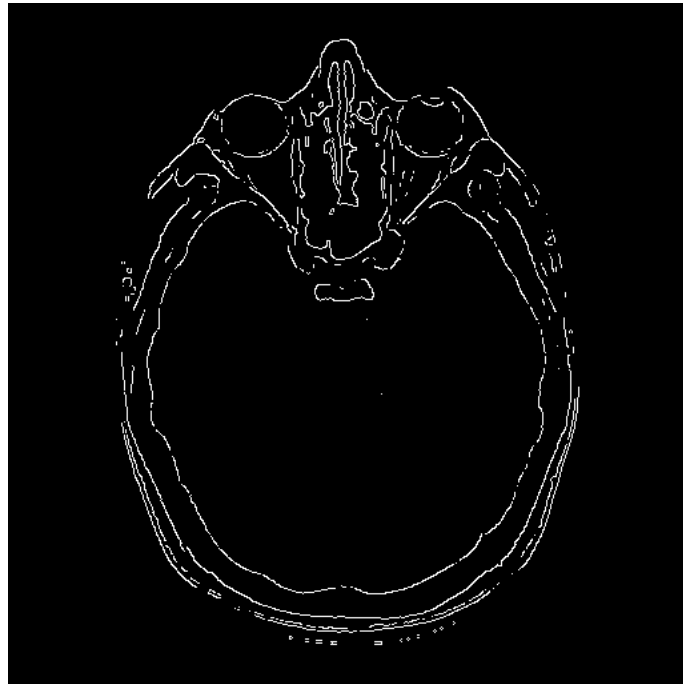


Figure 9: Image after canny edge detection using Poisson smoothing filter.

### 3.2 Other examples

In this section are presented some other examples (original image and image after canny edge detection). In canny edge algorithm we used different parameters (starting with our standard 20% and 10% for TH and TL and then change them manually) and different smoothing filters – depends on which parameters and filter achieved better results.

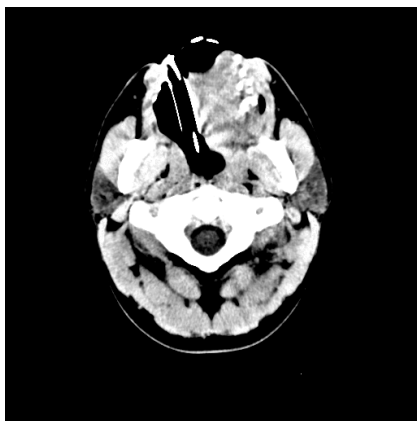


Figure 10: Original image.

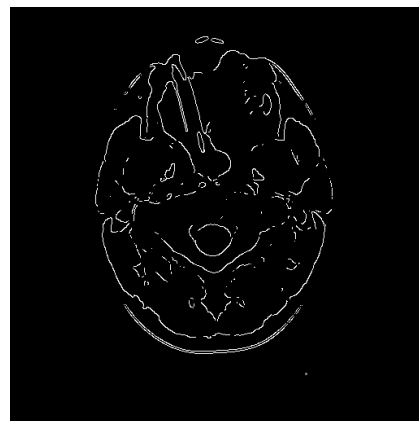


Figure 11: Image after canny edge detection using Poisson smoothing.

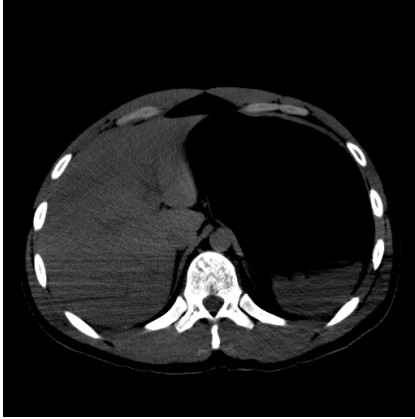


Figure 12: Original image.

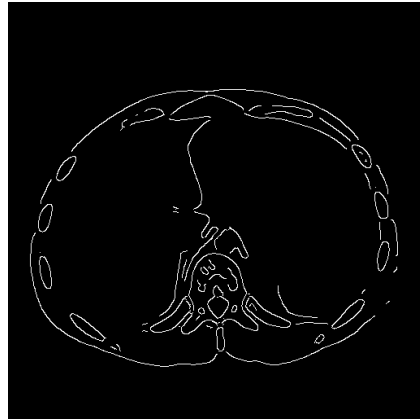


Figure 13: Image after canny edge detection using Gaussian smoothing.

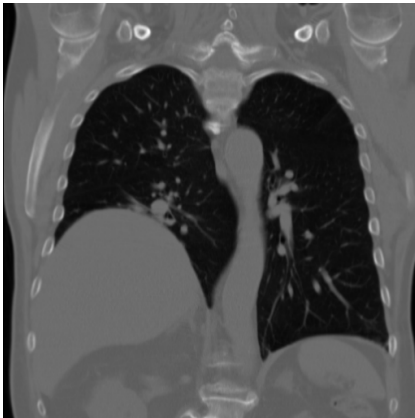


Figure 14: Original image.



Figure 15: Image after canny edge detection using Poisson smoothing.

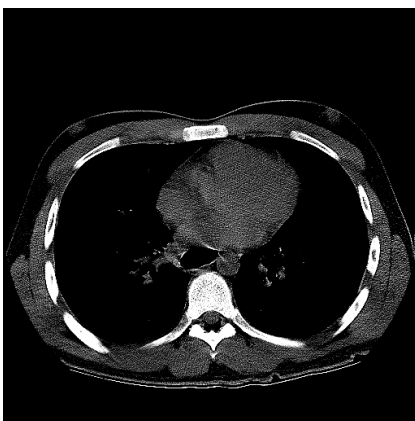


Figure 16: Original image.



Figure 17: Image after canny edge detection using Gaussian smoothing.

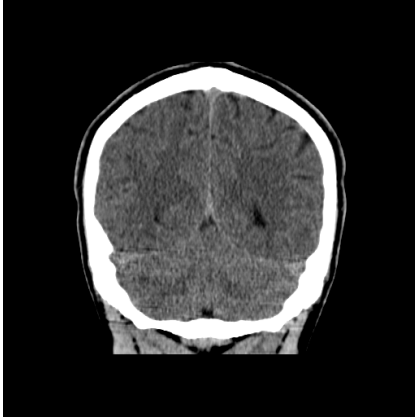


Figure 18: Original image.

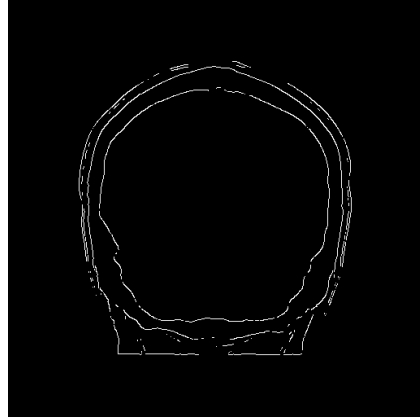


Figure 19: Image after canny edge detection using Poisson smoothing.

## 4 Conclusion

The results of our canny edge algorithm are satisfactory. Our selected threshold (TH as 20% and TL as 10% of maximal magnitude intensity) works fine in general, but to improve results of the algorithm, it is better to calibrate the parameters for each image separately, since they can vary a little. However, it is important that TH and TL remain in the similar relationship, that is, that proportion of intensity for which we cannot determine if the pixel is an edge or not (the intensity between TH and TL) is always about the same. Also, for every image we can determine which smoothing filter (Gaussian or Poisson) works better and perform canny edge with the better filter.

## References

- [1] *DTMRI DB*, [viewed 6. 1. 2021], available at <http://lbcsi.fri.uni-lj.si/OBSS/Data/CTMRI/>.
- [2] *Poisson smoothing filter*, [viewed 6. 1. 2021], <https://www.mathworks.com/help/images/ref/imnoise.html>.