Canny Edge Detector

Matjaž Zupančič Muc
Biomedical Signal and Image Processing 22/23, FRI, UL
mm1706@student.uni-lj.si

***Abstract*—In this work we implement a Canny Edge Detector. We use the Otsu's method to roughly determine the optimal threshold and display detections on a set of 6 images from the CTMRI database.**

I. Introduction

Edge detection is a fundamental problem in image processing and computer vision. Edge detection is used to reduce the amount of data in the image while preserving the structural properties of the image. In this work we implement a Canny Edge Detector (as proposed in [1]). We implement the Otsu's method (as proposed in [2]) to roughly determine the value of the binarization threshold. Finally we show detections on a set of 6 images from the CTMRI database ([3]).

II. Related Work

[1] presents a algorithm for detecting edges in images that is based on the concept of multi-scale differentiation and is designed to be both efficient and effective. The proposed algorithm consists of several steps which include: noise reduction, gradient computation, non-maximum suppression and hysteresis thresholding.

[2] proposed a method for determining the optimal threshold for segmenting an image. The method is based on the idea of minimizing the within-class variance of the two classes formed by thresholding the image. The within-class variance is a measure of the spread of the pixel intensities within each class.

III. Methodology

We start with the noise reduction stage. A Gaussian kernel is sued to filter the input image, this reduces the noise and minimizes the impact of isolated pixels. The strength of the Gaussian kernel is controlled using the standard deviation σ. Figure 1b shows an example image filtered with a Gaussian kernel.

The filtered image is used to compute the gradient (i.e partial derivative with respect to x and y axis). We compute the partial derivative w.r.t x axis (denoted with g_x) and the partial derivative w.r.t y axis (denoted with g_y) by convolving the image with the Sobel's kernels G_x and G_y respectively. Finally we compute the magnitude image $M = \sqrt{g_x^2 + g_y^2}$ and the angle image $\theta = tan^{-1}(\frac{g_y}{g_x})$. Figure 1c shows the resulting magnitude image M and Figure 1d shows the resulting angle image θ.

The algorithm next performs non-maximum suppression. The method starts by discretizating the angles θ into four directions: direction which is a normal to the vertical edge, $+45°$ edge, horizontal edge and $-45°$ edge. Next the gradient magnitude at each pixel is compared to the gradient magnitudes of its neighbors in the gradient direction (i.e direction that is a normal to the edge). Pixels that are not a local maximum (i.e pixels which have a neighbour with a higher gradient magnitude value) are suppressed, leaving only those pixels that are likely to correspond to edges. Figure 1e shows the resulting image after we apply the non-maxima suppression. We can see that the edges are now much thinner than the ones on Figure 1c.

We compute a rough estimate of the high threshold using the output of the non-maxima supression stage. Estimated high threshold is computed using the Otsu's method. To determine the optimal threshold we iterate over all thresholds τ and select the one which maximizes the between class variance. For each threshold τ we split the pixels into background (background consists of all pixels with intensity smaller than τ) and into foreground (foreground consists of all pixels with intensity greater than or equal to τ). Finally we can compute the between class variance as follows: $w_1.w_2.(\mu_1 - \mu_2)^2$, where w_1 is the portion of pixels that belong to the background, w_2 is the portion of pixels that belong to the foreground, μ_1 is the mean of all background pixels, and μ_2 is the mean of all foreground pixels. The optimal threshold is therefore defined as $\tau^\star = \underset{\tau}{\text{argmax}}(w_1.w_2.(\mu_1 - \mu_2)^2)$.

In the hysteresis thresholding stage we define two thresholds $\tau_h = \alpha\tau^\star$ and $\tau_l = \frac{1}{2}\tau_h$. We label all pixels above τ_h as legitimate edges and remove edges bellow τ_l (as this are definitely not edges). This leaves us with edges that have a intensity value which is above τ_l and bellow τ_h. Note that the estimated threshold $\tau_h = \tau^\star$ works well only on average, therefore we should select a scaling factor α which is unique to our image.

Finally we perform the edge linking using the 8 connectivity method . We iterate over all legitimate edges and look for nearby weak edges (using a window of size 3×3). If a weak edge is connected to the legitimate edge we label the weak edge as legitimate and recursively follow it. The method eventually converges and effectively fills in the gaps between the legitimate edges. Figure 1f shows the detected edges using a scaling factor $\alpha = 0.5$.

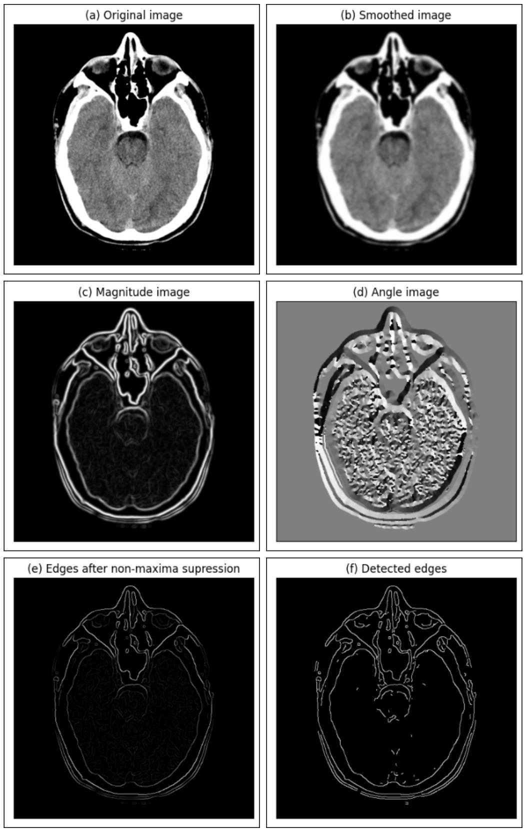

Fig. 1. (1) the original image, (1) image smoothed using a Gaussian kernel with $\sigma = 2$, (3) Gradient magnitude image, (3) Gradient direction image, (4) edges after non-maxima suppression, (5) detected edges.

IV. Experiments

Algorithm was evaluated on a set of 6 images from the CTMRI database ([3]). The following hyper-parameters were used: standard deviation of the Gaussian kernel is fixed at $\sigma = 2$ for all images, the threshold scaling factor is fixed at $\alpha = 0.5$ and adjusted for each image in the second experiment.

V. Results and Discussion

A. Results

B. Discussion

Figure 2 shows the 6 input MRI images, Figure 3 shows detections using a fixed scaling factor of $\alpha = 0.5$. Figure 4 shows detections using a varying scaling factor α (we used $0.5, 0.25, 0.5, 0.4, 0.35, 0.45$). We can see that as the scaling factor α decreases the number of detected edges increases.

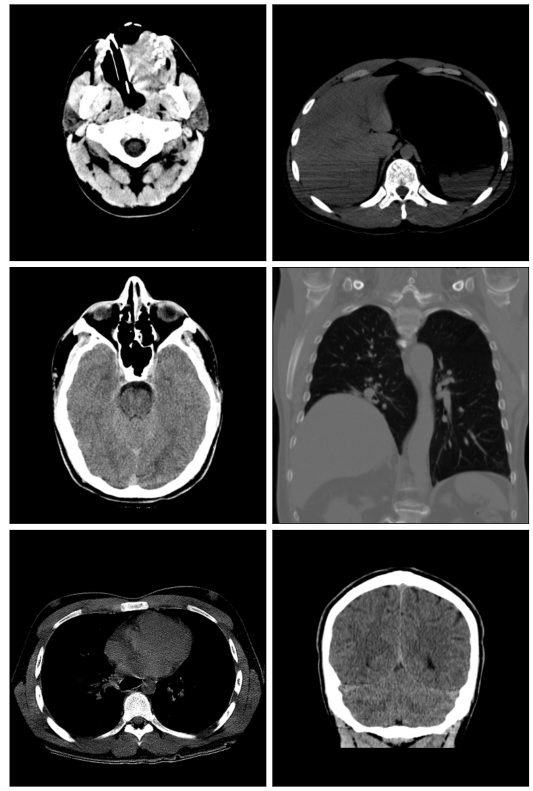

Fig. 2. Selected MRI images

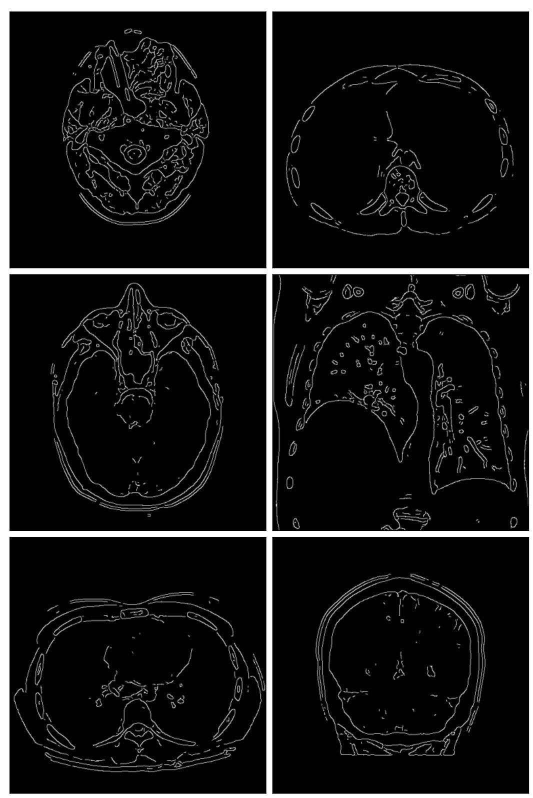

Fig. 3. Detected edges using a fixed scaling factor $\alpha = 0.5$

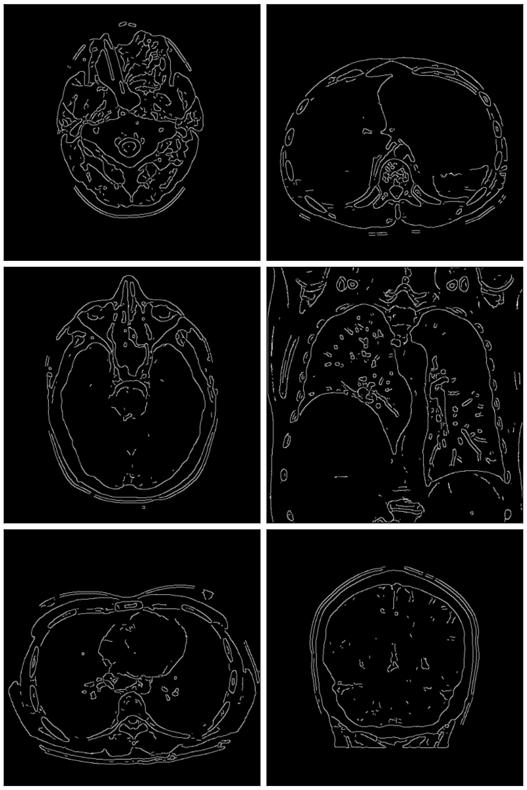

Fig. 4. Detected edges using a varying scaling factor α

VI. Conclusion

We implemented a Canny Edge Detector and used the Otsu's method to estimate the high threshold used in hysteresis thresholding stage. The implemented detector is controlled by a single parameter (threshold scaling factor α). Finally we display detections on a set of images from CTMRI dataset. We display detections for a fixed scaling factor $\alpha = 0.5$ and detections for a varying scaling factor (i.e set individually for each image). We show that as the scaling factor α decreases the number of detected edges increases. Detector could be further improved by implementing a more sophisticated edge linking method (e.g we could take in account more than 8 pixels when performing the edge linking).

References

[1] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[2] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[3] A. Taddei, F. Jager, A. Smrdel, K. Bezlaj, and M. Zadnikar, "Ct-mri database." [Online]. Available: https://lbcsi.fri.uni-lj.si/OBSS/Data/CTMRI/