Detecting Contours of Human Organs in 3D CT Images using the Canny Edge Detector

Jernej Vivod (63160328)

January 13, 2020

1 Introduction

For the second assignment for the Biomedical signal and image processing class, we implemented a version of the Canny edge detector that is suitable for processing three-dimensional images specified in the form of rank-3 tensors. The images comprising the sequence are first individually processed using the Canny edge detector and are then linked using the method of 24-connectivity. In this report, we describe the methodology used to implement the functionality needed to complete this assignment as well as the obtained results.

2 Methods

2.1 The Canny Edge Detector

The Canny edge detector is an edge detection algorithm that uses multiple stages to detect a wide range of edges in images. It makes use of Gaussian smoothing, image gradient information and thresholding to infer the locations of edges.

Before the gradient is computed, the image is smoothed using a Gaussian kernel to suppress noise that would otherwise be amplified by the derivations. Then, the gradient magnitude and orientation are computed for every point in the image. Figure 1 shows the visualization of image gradient magnitude and image gradient orientation for a sample image.

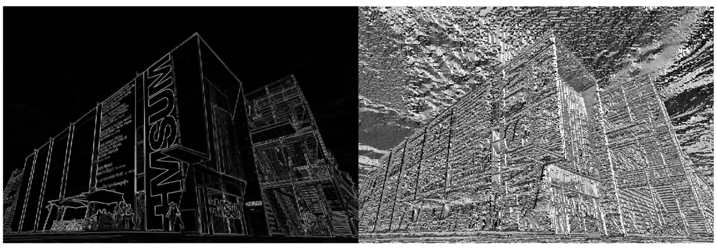

Figure 1: Visualization of image gradient magnitude (left) and image gradient orientation (right).

Non-maxima thinning is performed to thin the edges to a single-point width. This is performed by taking the maximum along lines perpendicular to the edges with the gradient as their directional vector. This result is further processed to discard potential false detections. For this, hysteresis thresholding is used. Figure 2 shows the image after non-maxima thinning.

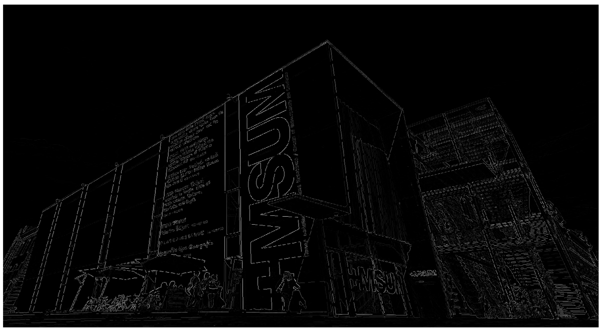

Figure 2: Image after non-maxima thinning.

Hysteresis thresholding uses two thresholds to trace the edges. The tracing starts at points with values above the first (high) threshold and continues over all points connected to this area, that pass the second (low) threshold. This produces the final binary image with single-point edge widths. The image after hysteresis thresholding is shown on figure 3

Figure 3: Image after hysteresis thresholding.

2.2 24-connectivity

We implemented the method of 24-connectivity to link images in the sequence comprising the three-dimensional image. This method links points with value 1 in the n-th image with points with value 1 in the neighborhood of the position of the point in the first image. A detailed algorithm was provided in the instructions. Figure 4 shows a neighborhood on the (n+1)-th image of a position of point with a value 1 on the n-th image before and after the linking procedure.

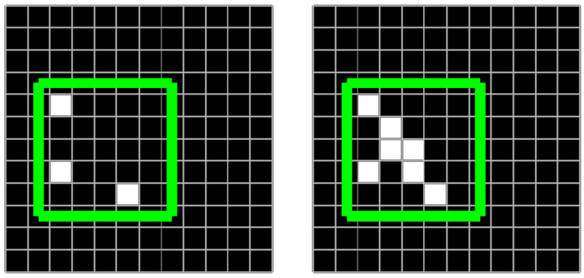

Figure 4: Neighborhood on (n+1)-th image of position of point on n-th image before (left) and after (right) linking using 24-connectivity.

3 Results

We implemented a script called *main.m* that reads a path to a dataset folder and processes, visualizes and finally saves images to a results folder in the dataset folder. The project folder contains one such example dataset for which this procedure can automatically be run by running the *example.m* file in the root of the project folder. The results can be visualized using an interactive figure automatically initialized by the *main.m* script. Figure 5 shows the interactive figures used to visualize the processed images for the provided sample dataset.

Figure 5: Interactive figures used to visualize processed images.