BSIP seminar 2 report: Detecting contours of human organs in 3D CT images using Marr-Hildreth edge detector

Luka Boljević
BSIP 22/23, UL FRI
lb7093@student.uni-lj.si

***Abstract*—In this seminar, we implemented the Marr-Hildreth edge detector, in order to detect contours i.e. edges in 3D CT images from CT-MRI DB. We first needed to implement the 2D version of Marr-Hildreth, so it can be used for the 3D version. We implemented the original and improved Marr-Hildreth algorithm, with the improvements described in "Methods" section. In this paper we show results of 2D Marr-Hildreth on one CT image, with more obtainable from the source code. We also show the result of 3D Marr-Hildreth on 211 head CT slices.**

I. Introduction

The goal of this seminar is to implement an edge detection algorithm, to detect contours in 2D and 3D CT images, originating from CT-MRI DB [1]. CT-MRI DB stands for Computer Tomography-Magnetic Imaging Resonance database, and as it suggests, it contains CT and MRI images from a group of patients.

For edge detection in those images, we chose the popular Marr-Hildreth (MH for short) algorithm [2]. To test our implementation, we chose a few CT images from 3 different patients. The CT images include a patient's head from above or the side, or a patient's torso. To reduce paper length, we will show results of 2D MH for only for one of those images (head slice from above). The rest is easily obtainable from the source code [3]. To test the 3D version, we chose a group of 211 head slices from above, originating from the same patient. One of those head slices, slice 0101, is the one we will be showing results for using 2D MH. We will also show how that slice looks like in 3D.

II. Methods

We first implemented the original MH algorithm - calculate Laplacian of Gauss (LoG) of input CT image, and perform zero crossing detection to detect the edges. To check whether a pixel p in LoG image is an edge pixel, we check its 3x3 neighborhood. To be classified as an edge pixel, the signs of opposing neighboring pixels of p must differ (there are four cases: left/right, up/down, and the two diagonals), and the absolute value of the difference between their values in LoG image must exceed a threshold T, in at least two cases. The result is a binary image, with 1 being an edge, and 0 not being an edge.

We then implemented 2 connected improvements. The first improvement is in the zero crossing detection stage. Instead of using a single threshold, we use two - a high (TH) and low (TL) one. Zero crossing detection is done in the same way as described, with the difference being that we now compare the previously described absolute difference against two thresholds, TL and TH, instead of just one. Any pixel with the absolute difference higher than TH, will be considered a strong pixel - a definite edge pixel. Those with the absolute difference between TL and TH are weak pixels - potential edge pixels. Those with the absolute difference below TL are definitely *not* edge pixels.

The second improvement is edge linking using 8-connectivity. Using these strong and weak pixels, we can check if any weak pixel is connected to a strong one, based on 8-connectivity. If yes, we mark the weak as a strong pixel. Otherwise, we disregard it. We carry out this process of edge linking until there exist weak pixels that can connect to a strong one. The resulting edges can then optionally be thinned, using morphological image processing.

The 3D MH version involves applying the improved MH algorithm to all CT slices corresponding to one organ, or body part, and then linking those slices using 24-connectivity. The resulting (linked) slices can optionally have their edges thinned using morphological image processing.

III. Results

There are two parameters for MH: (1) a single threshold T for the original, or two thresholds (low TL, and high TH) for improved MH, and (2) how much percentage of the minimum of the two image dimensions in pixels we take to be the σ parameter for the Gauss filter. Usually, we take 0.5%, but in the case of patient 000302-01, we got nicer results when we took $\sigma = 0.6\%$. The threshold values had to be set manually for each subject - there was no one threshold (or pair of high and low thresholds) that worked best for all tested CT images.

As discussed in the introduction, we will only show results of 2D MH on just one CT image, to reduce paper length. We will also show the entire 3D result, and how one slice from it looks like. Remember, one of the slices in the 3D image (slice 0101) is the same one as for the 2D case.

IV. Discussion

In Figure 2, we see that the original MH detector does a good job of detecting the majority of edges from the

Fig. 1. Original CT image. Exact image path on CT-MRI DB: **./Images-Patient-240163-01/<first_folder>/4/0101.png**

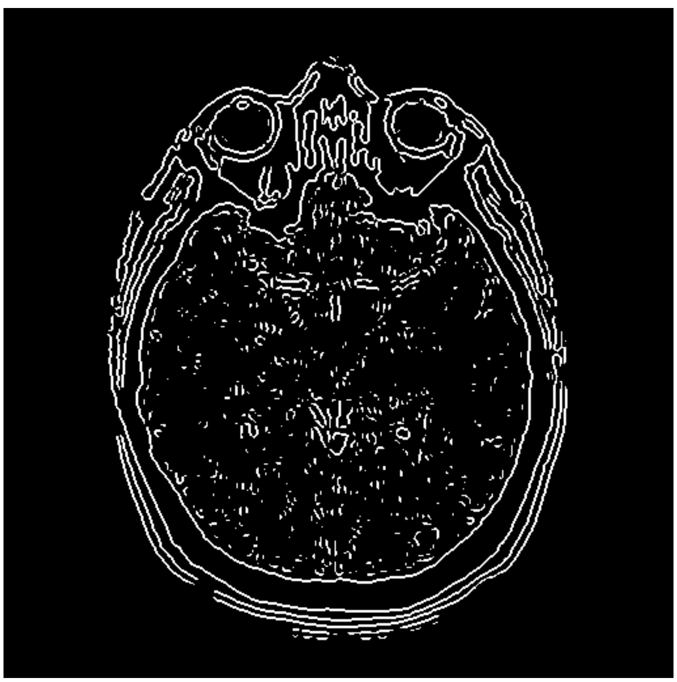

Fig. 2. Result of **original** Marr-Hildreth algorithm. Parameters were $\sigma = 0.005, T = 25$.

original image, especially the slightly thicker and longer ones. However, we immediately see the downside of MH - it does not do a good job of detecting very thin, or short/small edges. We can best see these "artifacts" in the space where brain matter is supposed to be - the MH detector finds something, but it looks more like noise than anything else. In some parts,

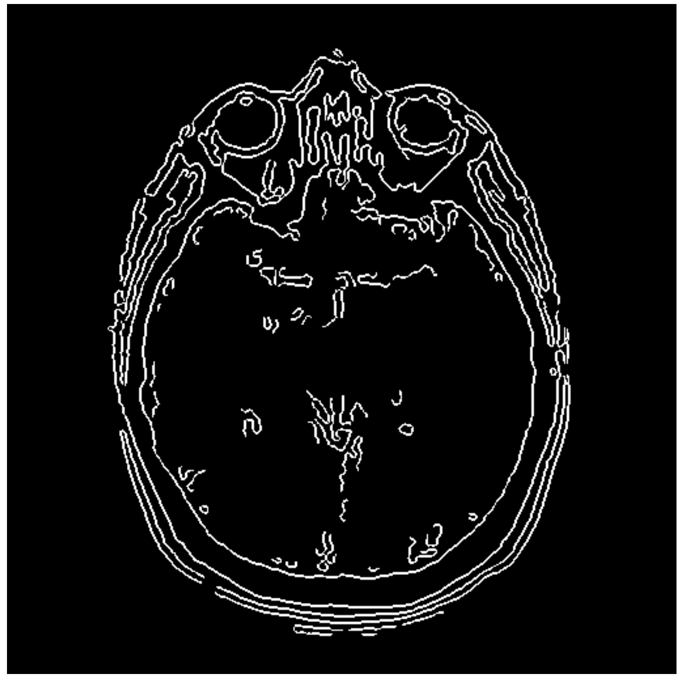

Fig. 3. Result of **improved** Marr-Hildreth algorithm (strong and weak pixels idea + edge linking using 8-connectivity). Parameters were $\sigma = 0.005, TL = 15, TH = 45$.

Fig. 4. Result of **improved** Marr-Hildreth algorithm with additional **edge thinning**. Parameters were $\sigma = 0.005, TL = 15, TH = 45$.

like at the very bottom of Figure 2, MH doesn't manage to connect edges which it should.

On the other hand, the improved MH algorithm produces a nicer result, as seen in Figure 3 - most of the "noise" is removed, and some edges which weren't connected, now are.

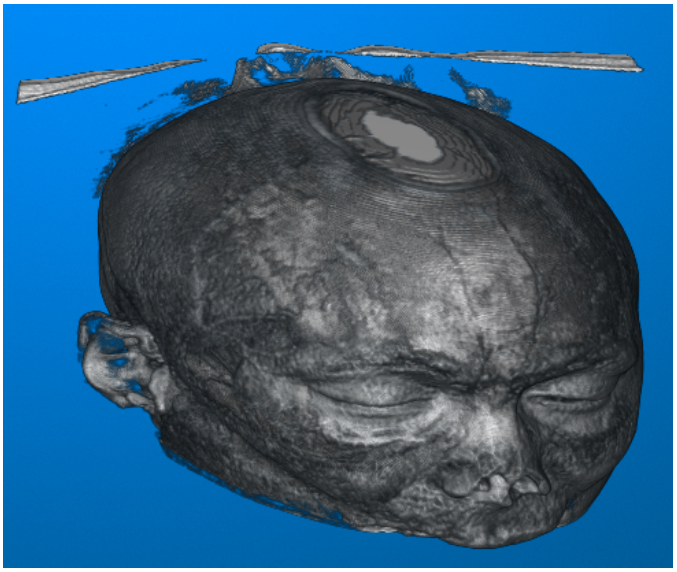

Fig. 5. The 211 CT head slices in 3D. Similar to Figure 1, the slices on CT-MRI DB are found under **./Images-Patient-240163-01/<first_folder>/4/**.

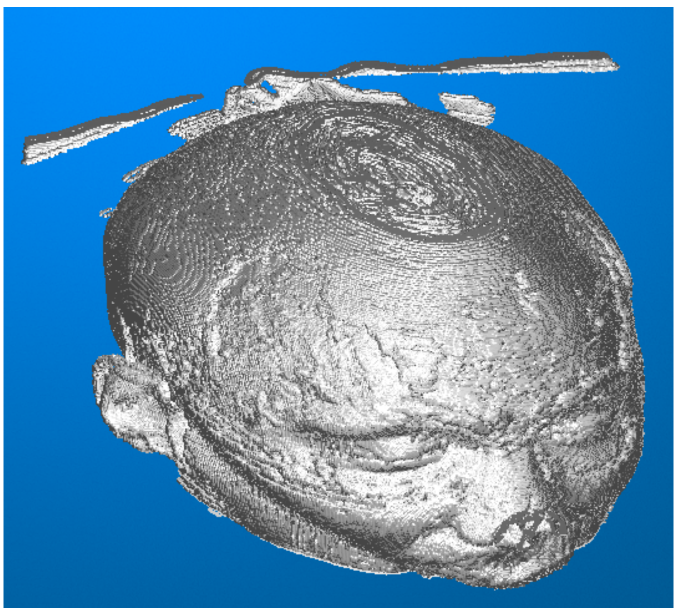

Fig. 6. Result of **3D** Marr-Hildreth algorithm, applied on slices mentioned in Figure 5. The improved Marr-Hildreth algorithm, with parameters $\sigma = 0.005, TL = 15, TH = 45$ and without edge thinning, was applied to each slice.

This result is closer to what we except from an edge detection algorithm. Additionally, we expect the edges to be as thin as possible, so the edge thinning step definitely gives us the cleanest result, as seen in Figure 4.

Figure 6 shows us that MH is capable of reconstructing the head (or any body part or organ for that matter) from Figure 5, in 3D - as seen in Figure 6. We can also see the effect of 24-connectivity used in 3D MH, in Figure 7. Compared to its 2D counterpart (Figure 3), the 3D slice has some additional tiny edges, which look like they are connecting to something above (or below) it. It is not a perfect approach, but we can at least say that the two neighboring slices are not totally disconnected.

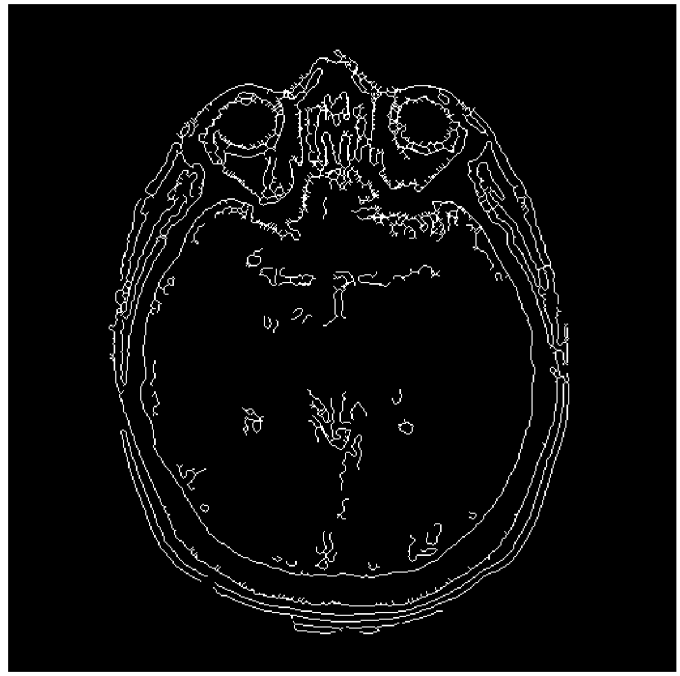

Fig. 7. Slice 0101 from the 3D head image in Figure 6. The slice is the same as the one in Figure 3, with the additional step of 24-connectivity.

V. Conclusion

In this seminar, we implemented a 2D and 3D version of Marr-Hildreth edge detector, and tested it on CT images originating from the CT-MRI database. We implemented the original algorithm, and 2 connected improvements for the 2D case. In 3D, we used the improved 2D MH algorithm on 211 CT head slices, linking them between each other using 24-connectivity.

We discussed that original MH detects the majority of edges from the original image, but fails to detect thin/small ones. The improved MH algorithm produces much cleaner and nicer results, especially if we use edge thinning. In 3D, we were able to reconstruct (almost) the entire head using the 211 slices.

While it still doesn't give as fine edges as Canny's edge detection algorithm, the improved MH algorithm isn't all too bad. However, there are still potential improvements. Just one out of possibly many might be to use Otsu's method on some image property, to "dynamically" set the threshold T, i.e. thresholds TL and TH.

References

[1] 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/

[2] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London Series B*, vol. 207, 1980.

[3] L. Boljević, "Source Code on GitHub," Link.