Multiscale Graph-Cut for 3D Segmentation of Compact Objects * ** ***

Miroslav Jirik[1,2][0000−0002−8002−2079], Vladimir Lukes[2][0000−0001−6579−1868], Milos Zelezny[2][0000−0003−1695−4370], and Vaclav Liska[3][0000−0002−5226−0280]

[1] Biomedical Center, Faculty of Medicine in Pilsen, Charles University, Czech Republic
[2] NTIS - New Technologies for the Information Society, Faculty of Applied Sciences, Pilsen, Czech Republic
[3] Biomedical Center and Department of Surgery, University Hospital and Faculty of Medicine in Pilsen, Charles University, Czech Republic,

Abstract. The research described in this article focuses on segmentation using the Graph-Cut method. This method is optimized for data in which the target object occupies only a small portion of the total volume. Two step procedure is used. In the first step, the location of the object is roughly localized. In the second step, Graph-Cut segmentation is performed with a special multi-scale chart structure. Two different graph construction methods have been proposed. The calculation time of both variants was compared with the original Graph-Cut. The msgc_lo2hi method has been shown to provide a statistically significant time reduction of computational costs.

Keywords: Graph-Cut· multiscale · medical imaging image segmentation

1 Introduction

The graph-cut algorithm for combinatorial optimization has been used for image segmentation for a long time. It was first described to segment the black and white image [7]. A significant improvement came from research that compared

* This work has been supported by Charles University Research Centre program UNCE/MED/006 "University Center of Clinical and Experimental Liver Surgery" and Ministry of Education project ITI CZ.02.1.01/0.0/0.0/17_048/0007280: Application of modern technologies in medicine and industry and Erasmus+ project MedTrain3DModsim, nr. 2016-1-TR01-KA203-034929 provided by the Turkish National Agency

** This research is supported by the project LO 1506 of the Czech Ministry of Education, Youth and Sports.

*** Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

the algorithms for calculating the graph cut [1, 3]. Main improvements were suggested and used for image segmentation.

Graph-Cut image segmentation is currently a popular computer vision method. Under certain conditions, a global minimum of the critical function can be found. For other segmentation methods, it is often very difficult to determine a critical function, and the results are often very different from the global optimum.

Advantage of the Graph-Cut is the possibility of general use in N-D space. This allows us to use the method for processing data from 3D medical imaging methods [6]. However the size of the data processed in diagnostic procedure is enormous and the computation time can easily exceed several minutes. It can be tolerable for offline computation, but it can not be accepted for interactive tasks required for example for segmentation of parenchymatous abdominal organs (e.g. liver) in CT images. Our goal in this paper is to reduce the time requirements in these tasks.

To reduce computational costs of Graph-Cut segmentation, Kang and Wan proposed segmentation described in [10]. It uses a rectangular area of interest to limit the extent of the calculation. However, this method does not allow the reduction of computational costs for objects with complex shape.

In order to increase the efficiency of the algorithm, we propose multi-scale segmentation using the Graph-Cut method. It is based on detection of discontinuities in the 3D image and unique construction of corresponding graph. We introduced two different methods of graph construction.

2 Methods

Data from three-dimensional medical imagery contain a large number of voxels and their processing requires large amount of memory. Despite the efficiency of the algorithm used to calculate the Graph-Cut segmentation, the calculation time and the memory costs are considerable. The fact that the target volume is rather compact and the segmentation boundary occupies a small area of the total volume of input data can be used for improvement of a segmentation procedure.

Method described in [10] for an application in the bright-field microscope images is based on performing Graph-Cut on subset of input image. In this case, rectangular areas are determined based on the Bhattacharyya distance for exploration on a fine scale. A local calculation is then performed on these areas. Output of the Graph-Cut segmentation is then transformed back to the coordinate system of the original image.

2.1 Graph-Cut for computer vision

The image segmentation chart described in the article [2] is well known algorithm. It was used in our experiments as a baseline: To distinguish it from our modifications, we call it Single Scale Graph-Cut (SSGC). We use selection of a few foreground and background voxels as an input of the algorithm. Foreground and background intensity models are composed of three-component Gaussian mixtures which is set by using the EM algorithm [4]. This model is used for setting weights in the graph.

Graph contains the same number of nodes as the number of pixels (see Fig 1). Edges connecting neighboring nodes (e.g. p and q) are called N-links. In addition,

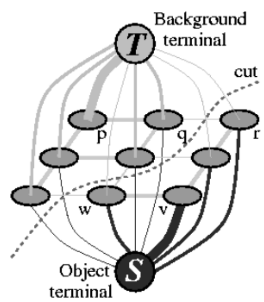

Fig. 1. Simple 2D example of graph. Nodes corresponding to image pixels (p, q, r, v and w) are connected by N-links, terminal nodes S and T are connected to other nodes with T-links. [2]

there are two nodes (terminal nodes) that represent foreground and background. The edges connecting the terminal nodes with the other graph nodes (e.g. S and v) are called T-links. The N-links weights corresponds to local discontinuities in the image. The T-links weights are corresponding to the likelihood with the foreground and background model. An algorithm max-flow/min-cut is run after the graph is constructed.

2.2 Multiscale Graph-Cut segmentation

The main idea of multiscale segmentation is to work with high resolution only close to the border between foreground and background object and in low resolution in other parts of the data, i.e. those that are far from the border. The idea is shown on figure 2. From the algorithm described in [10], the approach suggested here differs in working with the full volume of data. In addition, it is able to simplify the graph structure even inside of bounding box area of the segmented object. The high resolution area may not be rectangular or convex. This makes algorithm more efficient.

All steps of the algorithm are shown in table 1. First step is the feature modeling. Although our implementation allows us to use wider range of feature descriptors (e.g. textures), we use intensity modeling in this paper. As with the Single Scale Graph-Cut, the Gaussian mixture is used to model foreground and background.

The second step in the process is to locate discontinuities in the image. This is ensured by performing gross segmentation on resampled data. The reduced resolution is determined by a block size. Similarly, the seed is over-sampled.

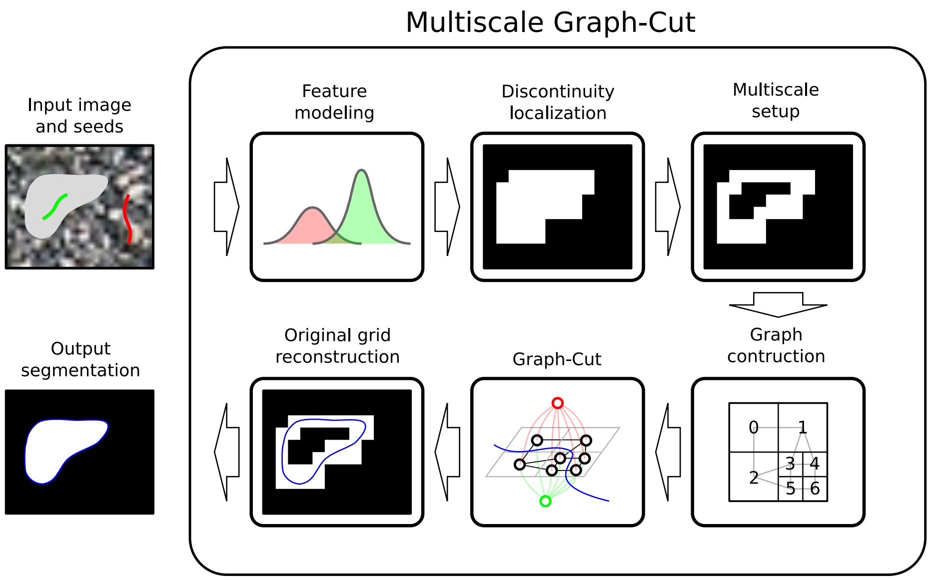

Fig. 2. Multiscale Graph-Cut scheme. Discontinuity localization is used for selection of voxels close to the border of segmentation

Table 1. Algorithm of Multiscale Graph-Cut segmentation

Step	Name	Description
1	Feature modeling	Gaussian mixture intensity model
2	Discontinuity localization	Low resolution segmentation
3	Multiscale setup	Set the distance of high resolution area from the border
4	Graph construction	Low to high algorithm or High to low algorithm
5	Graph-Cut	Perform max-flow/min-cut algorithm
6	High resolution grid reconstruction	Extract segmentation from the graph

In this case, interpolation by a nearest neighbor is used. In places near the border, controversial cases may arise, where pixels representing foreground and background segments may be within the block. In our implementation, we have left decision-making on the rule of the closest neighbor. An alternative approach is the removal of the inbred seed. Subsequently, the segmentation is performed in a conventional manner. Because of the low resolution of image data, this process is very fast.

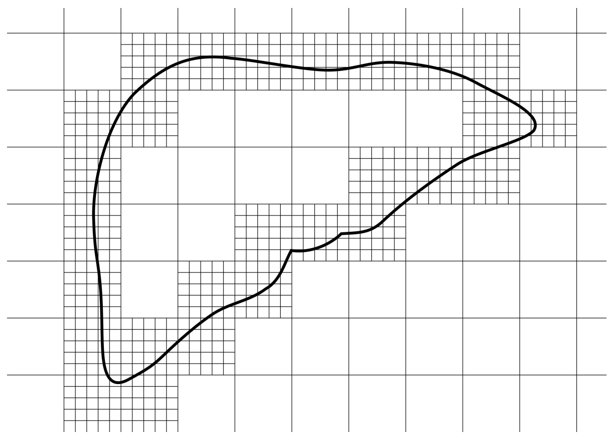

Fig. 3. Multiscale pixel schema. Discontinuity localization is used for selection of voxels close to the border of segmentation

The next step is to identify the areas that will be calculated with the fine resolution. The multiscale idea is shown in figure 3 Fine resolution area it along the edge of the rough segmentation from the previous step. One of the parameters of our method is then the width of this edge. This may include blocks that are close to the boundary but do not touch it directly. This parameter is then chosen taking into account the expected shape richness of the segmented object. In the example in the figure 3 the border width is set to 1. It means that only blocks located on the border are processed on high resolution.

The graph construction step is the most complex part of the multiscale segmentation procedure. We have proposed two different algorithms. The High to Low (hi2lo) algorithm builds the graph by simplifying the high resolution structure. The other one is called Low to High and it start from low resolution structure. Both of them would be discussed later in 2.3 and 2.4. Main goal of this step is to build the graph structure with all the weights. The figure 4 show small example with multiscale structure. There are three types of N-links. Weight of an edge n in the high resolution grid edge is set to w_n according to N-link rule from the SSGC. The weight w_o between high resolution pixel and the low resolution pixel is set the same as w_n. The weight m between two low resolution pixels is set to $b \cdot w_n$.

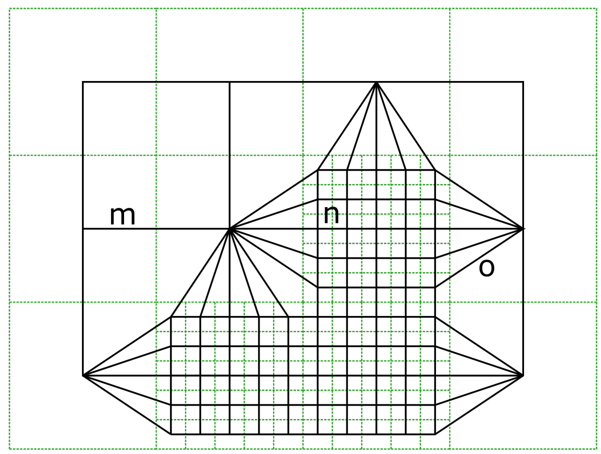

Fig. 4. 2D multiscale graph example with 3×4 blocks and 15×20 pixels. Green color is used for blocks and high resolution pixels

The final step is a high resolution grid reconstruction. In this step the graph structure and the Graph-Cut output is transformed into the original resolution.

2.3 High resolution to low resolution algorithm

The creation of the N-link multi-scale graph begins with the high resolution grid graph. Nodes are indexed ascendingly in the grid and the edges are represented by a pair of indices of adjacent nodes. Node indices in low resolution block areas are replaced by a new low resolution node index. One low-resolution node replaces multiple high-resolution nodes in the entire block. The edges inside of this block are searched and removed. The two adjacent low resolution blocks then create a series of edges connecting the same nodes. These duplicate edges are also removed. Some nodes are now not used in the edge list anymore. The list of node indexes is rearranged to avoid unused nodes.

2.4 Low resolution to high resolution algorithm

The multidimensional graph is made up of a low resolution grid graph. In each high resolution block an high-resolution grid is created. It is then connected to the surrounding blocks. If a neighboring block is high resolution, high-resolution border nodes are attached directly to each other. Otherwise, all border nodes are connected to one neighboring low-resolution node. During the reconnection, nodes and edges designated for removal are marked. This is then done at the end of the whole process.

3 Results

Experiments were performed on artificial three-dimensional images (see Figure 5). Every image contain a compact object with different brightness and the image is affected by Gaussian noise. This is typical situation for the CT images of parenchymatous organs. The image has cubic shape. The size of the image defined by number of the pixels on the cube edge varies and the volume of the object is constant. Seeds of one type are located within this object, second type segments are located on the outside of the object.

We used 74 images for the experiments and extended dataset with 151 to produce some figures in this paper. We used the compact objects because it is a typical situ The goal of the experiment is to verify the ability of the Multi-Scale Graph-Cut algorithm (MSGC) to reduce computational and memory requirements. Figure shows the comparison of the calculation time of the SSGC

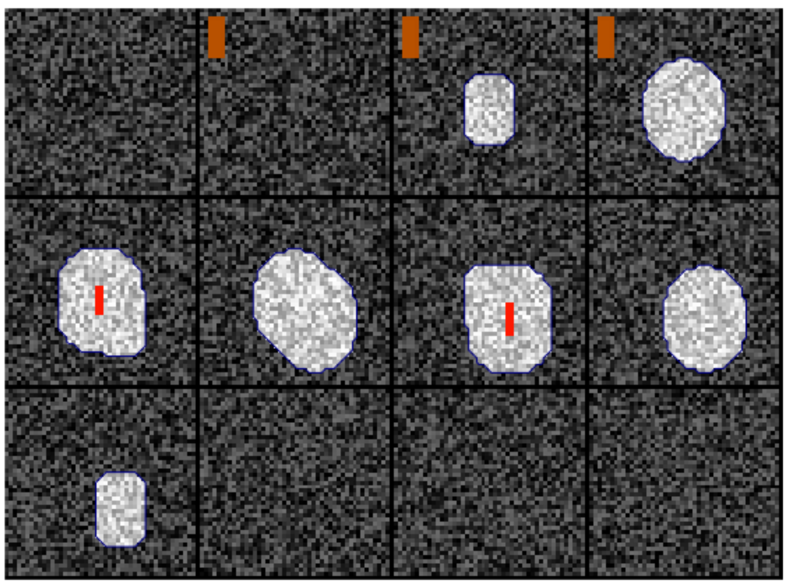

Fig. 5. Sample of test data with seeds

methods.

In second experiment we used Micro-CT porcine liver corrosion cast data [5]. Segmented veins on image covers whole area of 3D data. Measure times for SSGC, MSGC_hi2lo ad MSGC_lo2hi are 19.18[s], 617.4[s] and 71.48[s]. All experiment scripts are publicly available [8].

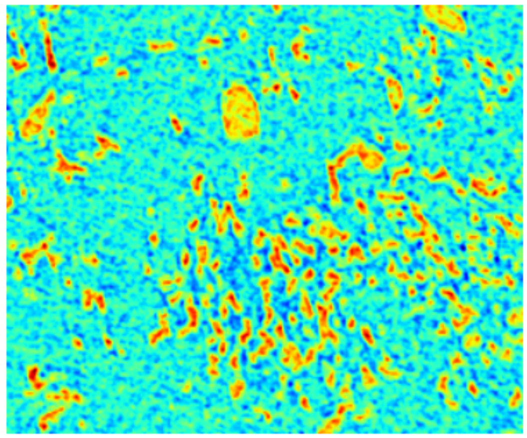

Fig. 6. One slice of porcine liver corrosion cast from Micro-CT

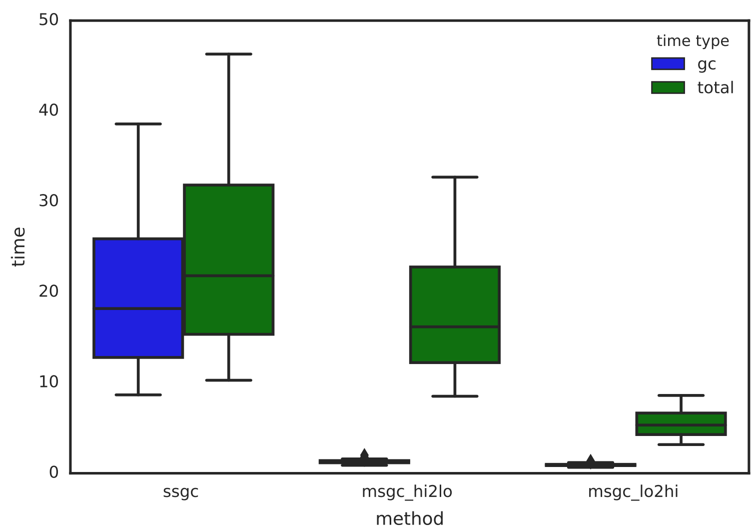

Fig. 7. Comparison of computation time of Single Scale Graph-Cut(SSGC) and Multiscale Graph-Cut (with variants lo2hi and hi2lo). With green color (labeled as "total") the total time of whole segmentation process is visualized. The purple boxes (labeled as "gc") show time for max-flow/min-cut algorithm

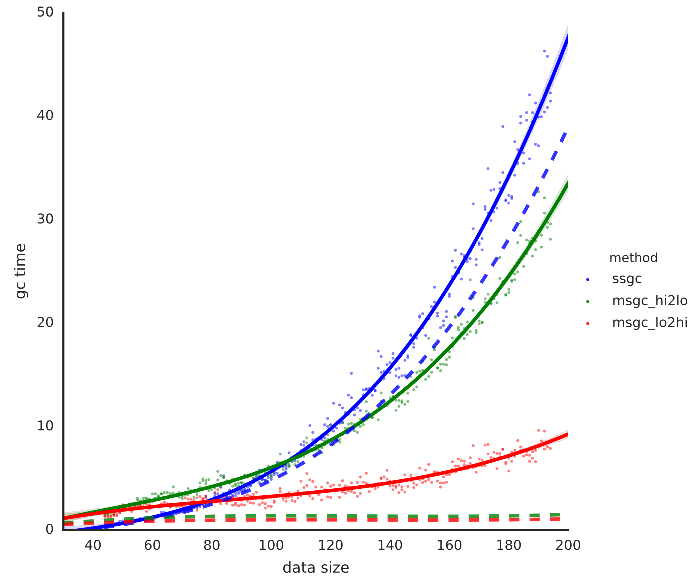

Fig. 8. Data size to time dependency. Time used for max-flow/min-cut algorithm is show by dashed line. Extended dataset was used to produce this figure.

4 Discussion

Based on the measurements, there is an evidence that after reaching some critical limit of data size the MSGC method starts to be faster than SSGC. The larger the data area, the more times the measured times are in favor of MSGC. The critical limit is affected by size and shape of the object. The smaller the surface, the more complexity reduction by MSGC method.

In our experiments we used 74 images with size higher than 130. On this dataset both MSGS variants are faster than SSGC for larger segmented image sizes. The one sided paired T-test to proof it. MSGV_hi2lo is faster than SSGC with p=1.896×10^{-22} and MSGC_lo2hi with p=4.882×10^{-29}. The lo2hi variant overcomes both SSGC and MSGC_lo2hi (with p=9.359×10^{-32}. This is mainly due to the economical algorithm of chart construction.

5 Conclusion

The Graph-Cut method, when used on large data, has large computational demands. That's why we designed and verified a method to speed it up. We showed on experiment that our Multiscale Graph-Cut method allow to reduce computational cost on data with compact objects. We introduced two graph construction method. The implementation is available as a python module called imcut [8].

The segmentation is used in application for computer aided liver surgery LISA [9].

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(11), 1222–1239 (2001). https://doi.org/10.1109/34.969114, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=969114
2. Boykov, Y., Funka-Lea, G.: Graph Cuts and Efficient N-D Image Segmentation. International Journal of Computer Vision **70**(2), 109–131 (11 2006). https://doi.org/10.1007/s11263-006-7934-5, http://link.springer.com/10.1007/s11263-006-7934-5
3. Boykov, Y., Kolmogorov, V.: An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision. IEEE transactions on pattern analysis and machine intelligence **26**(9), 359–374 (9 2001). https://doi.org/10.1109/TPAMI.2004.60, /home/mjirik/Dropbox/studium/materialy/clanky/An Experimental Comparison of Min-Cut Max-Flow Algorithms for Energy Minimization in Vision.pdf http://www.ncbi.nlm.nih.gov/pubmed/15742889
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. . . . **39**(1), 1–38 (1977), http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract http://www.jstor.org/stable/10.2307/2984875
5. Eberlova, L., Liska, V., Mirka, H., Gregor, T., Tonar, Z., Palek, R., Skala, M., Bruha, J., Vycital, O., Kalusova, K., Haviar, S., Kralickova, M., Lametschwandtner, A.: Porcine liver vascular bed in Biodur E20 corrosion casts. Folia Morphologica **75**(2), 154–161 (2016). https://doi.org/10.5603/FM.a2015.0094
6. Gotra, A., Sivakumaran, L., Chartrand, G., Vu, K.N., Vandenbroucke-Menu, F., Kauffmann, C., Kadoury, S., Gallix, B., de Guise, J.A., Tang, A.: Liver segmentation: indications, techniques and future directions. Insights into Imaging **8**(4), 377–392 (2017). https://doi.org/10.1007/s13244-017-0558-1, http://link.springer.com/10.1007/s13244-017-0558-1
7. Grieg, D.M., Porteous, B.T., Scheult, A.H.: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society **51**(2), 271–279 (1989)
8. Jirik, M., Lukes, V.: imcut - 3D multiscale Graph-Cut segmentation module for python (2018), https://github.com/mjirik/imcut
9. Jirik, M., Lukeš, V.: LISA - Liver Surgery Analyser, https://github.com/mjirik/lisa
10. Kang, S.M., Wan, J.W.L.: A multiscale graph cut approach to bright-field multiple cell image segmentation using a Bhattacharyya measure **8669**, 86693S (2013). https://doi.org/10.1117/12.2007002, http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2007002