A Simple Topology Preserving Max-Flow Algorithm for Graph Cut Based Image Segmentation

Ondřej Daněk and Martin Maška

Centre for Biomedical Image Analysis, Faculty of Informatics
Masaryk University, Brno, Czech Republic
`xdanek2@fi.muni.cz`

Abstract

In this paper, we propose a modification to the Boykov-Kolmogorov maximum flow algorithm [2] in order to make the algorithm preserve the topology of an initial interface. This algorithm is being widely used in computer vision and image processing fields for its efficiency and speed when dealing with problems such as graph cut based image segmentation. Using our modification we are able to incorporate a topology prior into the algorithm allowing us to apply it in situations in which the inherent topological flexibility of graph cuts is inconvenient (e.g., biomedical image segmentation). Our approach exploits the simple point concept from digital geometry and is simpler and more straightforward to implement than previously introduced methods [14]. Due to the NP-completeness of the topology preserving problem our algorithm is only an approximation and is initialization dependent. However, promising results are demonstrated on graph cut based segmentation of both synthetic and real image data.

Keywords and phrases maximum flow, topology preserving, image segmentation, graph cuts

Digital Object Identifier 10.4230/OASIcs.MEMICS.2010.19

1 Introduction

Modern approaches to image segmentation often formulate the problem in terms of minimization of a suitable energy functional. Such methods have many benefits. Most notably, mathematically well-founded algorithms can be used to solve the originally vaguely specified task. Among the most popular energy minimization algorithms for image segmentation belong the level set framework [11] and recently also the graph cut framework [1, 3] both having their pros and cons depending on a particular situation. In this paper, we focus on the latter one.

Graph cuts, originally developed as an elegant tool for interactive object cutout, quickly emerged as a general technique for solving diverse computer vision problems such as image restoration or stereo [3]. In some sense, they can be seen as a combinatorial counterpart of the level set method [1]. Likewise level sets they are applicable to a wide range of energy functions [9], directly extensible to N-dimensional space, topologically flexible and with implicit boundary representation. Moreover, they offer integration of various types of regional or geometric constraints and the ability to reach global optima in polynomial time [2]. In this framework, the input image is converted to a weighted graph with the energy function encoded in the edge weights. Subsequently, a minimum $st-$cut [5] is found, effectively yielding a global minimum of the energy. Typically, maximum flow algorithms are used to find a minimum cut in the graph based on the Ford-Fulkerson max-flow/min-cut duality theorem [5].

The aforementioned topological flexibility of graph cuts is not always desirable. There are situations in which the number of objects in the image and their topology is known in advance, e.g., in biomedical image segmentation only one object topologically equivalent to a sphere should be segmented during brain extraction. Topology is also an important prior for object tracking where objects are not allowed to split or join. The topology preserving problem has been studied extensively in the context of level sets [7]. To the best of our knowledge, the literature is not as rich in the graph cut universe with the work of Zeng et al. [14] being the only relevant. In their work, the topology preservation is embedded into the maximum flow computation. Unfortunately, the algorithm is rather complicated with description missing many important details[1]. They also showed that the topology preserving problem is NP-complete so the devised algorithm no longer guarantees to reach a global minimum. A partially similar problem is addressed also in [12]. Another option of enforcing the topology preservation is through integration of hard constraints into the energy function itself. However, this may involve considerable amount of user interaction.

In this paper, we propose a new topology preserving maximum flow algorithm for graph cut based image segmentation. Similarly to [14] our algorithm is a modification of the Boykov-Kolmogorov algorithm [2] in a way that guarantees that the output of the algorithm conforms (in the topological sense) to a given initial interface. It is achieved by making sure that the topology of the initialization is preserved during label changes throughout the computation. We borrow several ideas from [14], however, our method is simpler and generally less error-prone implementation-wise. Nevertheless, it is still an approximation, i.e., only locally optimal solution is produced. We demonstrate the potential of the proposed method on graph cut based segmentation of both synthetic and real image data using the Chan-Vese segmentation model [4, 13].

This paper is organized as follows. In Section 2 a brief review of the Boykov-Kolmogorov maximum flow algorithm and simple point concept from digital geometry is given. The proposed modifications, complexity guarantees and differences from [14] are described in Section 3. Section 4 contains experimental results of the devised algorithm. We conclude the paper in Section 5.

2 Preliminaries

2.1 The Boykov-Kolmogorov Algorithm

The maximum flow algorithm introduced by Boykov and Kolmogorov is one of the most popular when dealing with graph cut based image processing [2].

▶ **Definition 1.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a directed graph with two distinguished nodes s and t, where every edge $(u, v) \in \mathcal{E}$ is assigned a non-negative real valued capacity c_{uv}. A *flow* is a mapping $f : \mathcal{E} \to \mathrm{R}^+$. It is called feasible if:

1. $\forall (u, v) \in \mathcal{E} : f_{uv} \leq c_{uv}$ (capacity constraint)

2. $\forall u \in \mathcal{V} \setminus \{s, t\} : \sum_{(u,v)\in\mathcal{E}} f_{uv} = \sum_{(w,u)\in\mathcal{E}} f_{wu}$ (flow conservation rule)

The flow value $|f|$ is defined as $\sum_{(s,v)\in\mathcal{E}} f_{sv}$ and the maximum flow problem is a problem of finding a feasible flow of a maximum value.

To find a maximum flow the Boykov-Kolmogorov algorithm (BKA) uses the augmenting path strategy [5]. This strategy involves iterative searching of a non-saturated path from s

[1] The author provided implementation has stability issues and the source code seems to perform operations not mentioned in the paper.

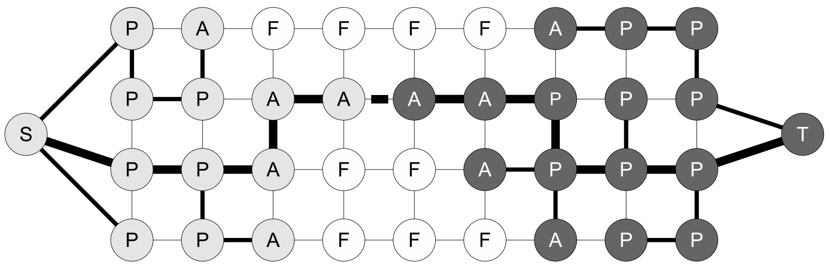

Figure 1 Boykov-Kolmogorov maximum flow algorithm scheme with active, passive and free nodes. An augmenting path (bold) is found when the two dynamic trees touch.

to t in the *residual graph* and pushing as much flow as possible along this path. A residual graph is obtained from $\mathcal{G}$ by taking the *residual capacity* $c^f_{uv} = c_{uv} - f_{uv}$ as the edge capacity for all $(u, v) \in \mathcal{E}$. The popularity and efficiency of BKA stems from the way augmenting paths are searched. It grows two dynamic trees from the terminal nodes s and t and an augmenting path is found when the two trees touch. This stage is called *tree growth*. In the *augmentation* stage the flow is sent along this path. This step may break the trees into forests (edges become saturated). Subsequently, *adoption* stage is performed to restore the two trees and the whole process is repeated. A visualization of BKA is depicted in Fig. 1. For a detailed description, please refer to [2].

Obtaining the requisite minimum cut is straightforward after the maximum flow has been found. Due to the min-cut/max-flow duality [5] the minimum cut is determined by the saturated edges (i.e., edges with zero residual capacity) and the cost of the cut is the same as the maximum flow value. In Section 3 we explain how these principles are exploited in image segmentation.

2.2 Digital Topology and Simple Point Concept

Digital topology is a subfield of digital geometry that deals with topological properties of digital (binary) images/objects, i.e., spatial properties such as connectedness (or number of objects and holes) that are invariant under certain kind of transformations (e.g., continuous deformations involving stretching, etc.) [8]. In this context, a simple point refers to a point whose switching from background to foreground or vice versa does not change the topology of the digital image. It is one of the fundamental ideas allowing topology preserving deformations of digital images. A fast characterization of simple points in 2D has been introduced by Klette and Rosenfeld [8]. Their method considers the number of connected background and foreground components in the 8-neighbourhood of the investigated point and can be efficiently implemented using a look-up table. An extension to 3D employing a breadth-first search in a small graph has been proposed in [10].

3 Topology Preserving Algorithm

In traditional graph cut based image segmentation a graph is created from the image where each node in the graph corresponds to an image voxel (plus the two terminal nodes s and t connected to all non-terminal nodes are added) and with the energy encoded in the edge weights [1]. A maximum flow algorithm is then used to find the minimum st-cut effectively yielding a global minimum of the energy. The final node labels are determined by the minimum cut partitioning. When using BKA this is equivalent to a so-called *tree membership*, i.e., the node/pixel is implicitly labelled as background or foreground depending on whether it lies in the s or t tree, respectively, after the computation has finished.

The described method does not impose any topology constraints on the result. In general, the final segmentation may contain arbitrary number of objects, holes, etc. To avoid this (for the reasons given in the introduction) we modify BKA to handle labels and topology changes explicitly. The modifications to particular stages of BKA are presented in the following subsections.

3.1 Initialization

Node labels are initialized using a user supplied mask (interface). The algorithm ensures that the final segmentation conforms to this initialization in the topological sense, e.g., if the initial mask contains one object with a hole there will be a single object with a hole on the output. In object tracking the initial mask may typically correspond to the segmentation from the previous time point.

The algorithm starts with a zero flow. Instead of two trees, four trees are maintained during the computation. We will denote them S_F, S_B, T_F and T_B. Initially, S_F tree contains nodes labelled as foreground and connected to s through an edge of non-zero residual capacity. Similarly, T_B contains nodes labelled as background and connected to t through an edge of non-zero residual capacity. Analogously for S_B and T_F.

3.2 Tree Growth

The tree nodes are called *active* if they are on the border of the tree (the tree can grow from them) otherwise they are *passive*. Nodes that do not belong to any of the trees are *free*. See Fig. 1 for illustration. Initially, all tree nodes are active. In this stage the four trees grow by acquiring new children for their active nodes. An active node p is chosen and its neighbours connected through an edge of non-zero residual capacity are considered for growth. Let $l(p)$ denote the label of p and $t(p)$ the associated tree. Following situations may arise when considering neighbouring node q:

- q is free: If $l(p) = l(q)$ then q is recruited as a child of p. If $l(p) \neq l(q)$ and q is simple then it is also recruited as a child of p and relabelled to $l(p)$. If q is recruited it becomes active. Nothing is done otherwise.
- $t(p) \neq t(q)$: An augmenting path has been found (irrespective of the node labels). The algorithm continues with the augmentation stage.
- $t(p) = t(q)$ and $l(p) \neq l(q)$: q is recruited as a child of p if it is simple and its label is associated to the opposite tree (recall that s is associated with the background and t with the foreground), i.e., if either (1) $t(q) = t$ and $l(q) = b$ or (2) $t(q) = s$ and $l(q) = f$. If q is recruited it is relabelled to $l(p)$ and becomes active.

As soon as all neighbours of p are explored the node becomes passive and a new active node is picked. When there are no active nodes the computation ends.

3.3 Augmentation and Adoption

These two stages remain the same as in the original algorithm. During augmentation nodes behind saturated edges and all their descendants became orphans. During the adoption stage new parents are searched for the orphans among their neighbouring nodes (in the same subtree). If no admissible parent is found the node becomes free.

3.4 Active Node Selection Rules

To ensure homogeneous propagation of the segmentation boundary (as in the level set algorithms) active nodes closest to the frontier between the foreground and background should be handled first in the tree growth stage. Various methods can be used to efficiently extract nodes closest to the frontier. The bucket priority queue approach introduced in [14] is not correct according to us and caused buffer underflows in both the original and our implementation. Instead we store queues of active nodes with the same distance from the separating boundary in an associative array. This approach has a logarithmic time complexity (in the number of graph nodes), however, it is correct and in practice as efficient (in both memory and time) as the constant-time method of [14]. To initialize the distance attribute of each node an L1 metric distance transform is employed in the beginning of the computation. Subsequently, this attribute is updated whenever a node label changes.

3.5 Complexity Guarantees and Discussion

The modified algorithm still runs in polynomial time. However, note that it is no longer guaranteed to reach a global minimum of the corresponding energy functional. Similarly to the level set based algorithms [7], only a locally optimal solution is obtained that may strongly depend on the initialization. As proved in [14] minimizing the original energy with the topology preserving constraint is an NP-complete problem. The output segmentation is given by the explicit node labels handled in the algorithm (i.e., not by the final tree membership of nodes). Because simple point check is always performed before node relabelling in the tree growth stage the solution topology has to conform to the initial mask. Finally, the main difference between our algorithm and [14] is the elimination of the overly complex inter- and intra-label steps. We treat all augmenting paths equally irrespective of the labelling.

4 Experimental Results

In this section we present the results of the proposed algorithm at segmentation of both real and synthetic image data. The Chan-Vese segmentation model [4, 13] is used as the energy functional being minimized. This model aims for partitioning the input image into two possibly disconnected regions (i.e., foreground and background) minimizing their intensity variance and the length of the separating boundary. We compare the results of the topology preserving algorithm with those obtained using the conventional Boykov-Kolmogorov algorithm. Three images were used for the comparison as depicted in Fig. 2.

The first experiment consisted of brain MRI image segmentation. Undesirable results are produced using the conventional topology-free algorithm where bright parts of the image are segmented. On the other hand, a single object corresponding to the brain is extracted using the topology preserving algorithm with the depicted initialization. Fluorescently stained cell nuclei are segmented in the second experiment. Using the standard algorithm all three nuclei merge into one object. This can be avoided using the topology preserving algorithm with initialization containing exactly three seeds as illustrated on the second row in Fig. 2. In the last experiment, topology preserving algorithm is used to keep the middle and ring fingers separated in the final segmentation. On a side note, despite our algorithm is different, we are also able to reproduce the results presented in [14].

Even though the results of the traditional and topology-preserving algorithms vary we have also conducted a comparison of running times of both methods to illustrate the performance penalty incurred by the additional simple point inspections. The test was performed on

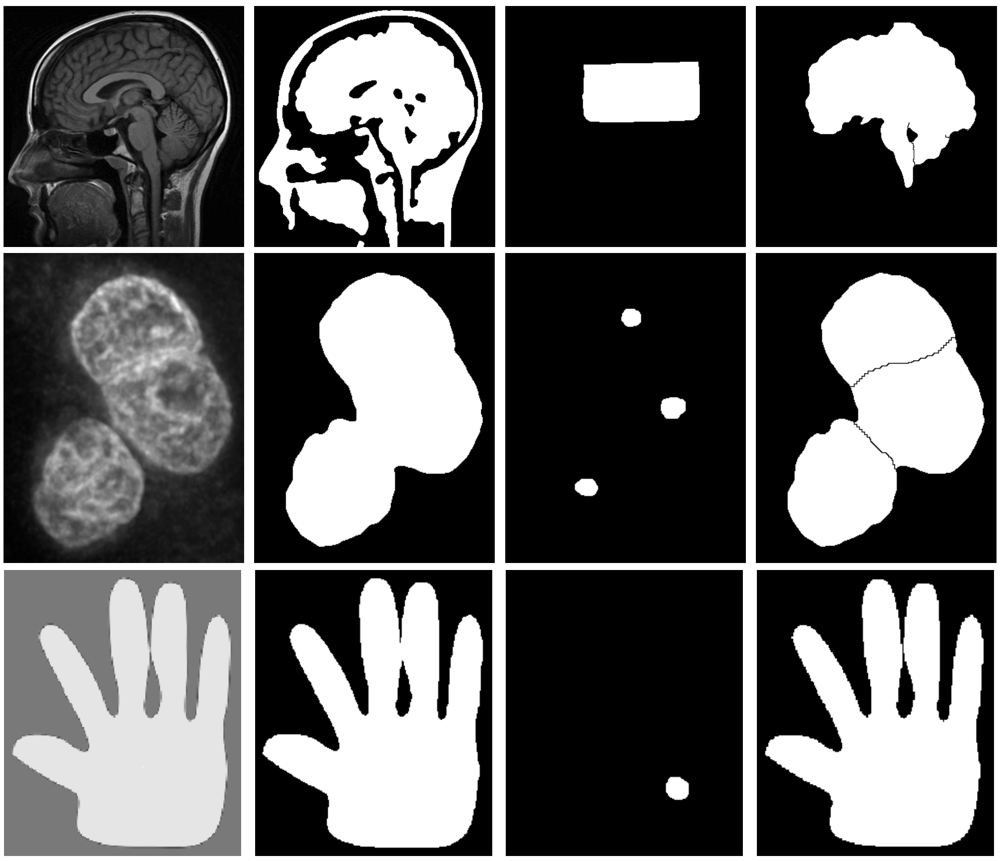

Figure 2 First column (from left): Input image. Second column: Segmentation using the conventional topology-free graph cuts. Third column: Initial mask for the topology preserving algorithm. White markers correspond to the foreground. Fourth column: Results of the topology preserving algorithm.

a common laptop equipped with an Intel Core 2 Duo processor at 2.0 GHz and 4 GB of RAM. As can be seen from the numbers listed in Table 1, the speed penalty is quite low in 2D. However, according to our experiments the topology preserving algorithm may be significantly slower in specific situations. Finally, we have not examined the performance of the algorithm in 3D. In this case a larger performance hit is to be expected due to the more complex simple point inspection routine.

5 Conclusion

A modification of the Boykov-Kolmogorov algorithm allowing topology preserving graph cut based image segmentation has been introduced in this paper. This modification is based on the simple point concept from digital geometry and is simpler than previously proposed algorithms. Despite its relative simplicity, it is able to achieve the same results and is ready for use in situations in which topology preserving is desirable. This was verified on a series of segmentation experiments. In our future work we would like to investigate the possibility of integration of topology preserving constraints also to other maximum flow algorithms such as the Push-Relabel method [6] or even the dynamic maximum flow algorithms. Implementation of the method described in this paper can be downloaded from

Table 1 Comparison of running times of the original Boykov-Kolmogorov algorithm and the proposed topology preserving modification.

Input image	Size	Boykov-Kolmogorov	Topology-preserving
Brain	350 × 350	2.02 s	2.89 s
Cell nuclei	280 × 361	0.49 s	0.51 s
Hand	228 × 275	0.10 s	0.13 s

our website *http://cbia.fi.muni.cz/projects/graph-cut-library.html.*

Acknowledgements This work has been supported by the Ministry of Education of the Czech Republic (Projects No. MSM-0021622419, No. LC535 and No. 2B06052).

References

1 Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation (review). *International Journal of Computer Vision*, 70(2):109–131, 2006.

2 Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

3 Yuri Boykov and Olga Veksler. *Handbook of Mathematical Models in Computer Vision*, chapter Graph cuts in vision and graphics: Theories and Applications, pages 79–96. Springer-Verlag, 2006.

4 Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

5 L. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

6 A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 136–146, 1986.

7 Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:755–768, 2003.

8 Reinhard Klette and Azriel Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

9 Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

10 G. Malandain and G. Bertrand. Fast characterization of 3d simple points. In *11th International Conference on Pattern Recognition*, pages 232–235, 1992.

11 Stanley J. Osher and Ronald P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 1 edition, October 2002.

12 S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR 2008: IEEE Conference on Computer Vision and Pattern Recognition.*, pages 1–8, jun. 2008.

13 Yun Zeng, W. Chen, and Q. Peng. Efficiently solving the piecewise constant mumford-shah model using graph cuts. Technical report, Dept. of Computer Science, Zhejiang University, P.R. China, 2006.

14 Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008.