Proceedings of the "Tenth IEEE International Conference on Computer Vision", ICCV 2005 Volume 1, pp. 259

A Multilevel Banded Graph Cuts Method for Fast Image Segmentation

Herve Lombaert Yiyong Sun Leo Grady Chenyang Xu
Department of Imaging and Visualization
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA

Abstract

In the short time since publication of Boykov and Jolly's seminal paper [3], graph cuts have become well established as a leading method in 2D and 3D semi-automated image segmentation. Although this approach is computationally feasible for many tasks, the memory overhead and supralinear time complexity of leading algorithms results in an excessive computational burden for high-resolution data. In this paper, we introduce a multilevel banded heuristic for computation of graph cuts that is motivated by the well-known narrow band algorithm in level set computation. We perform a number of numerical experiments to show that this heuristic drastically reduces both the running time and the memory consumption of graph cuts while producing nearly the same segmentation result as the conventional graph cuts. Additionally, we are able to characterize the type of segmentation target for which our multilevel banded heuristic will yield different results from the conventional graph cuts. The proposed method has been applied to both 2D and 3D images with promising results.

1. Introduction

The graph cuts technique of Boykov and Jolly [3] has witnessed an explosion of interest in recent years, rising rapidly to become one of the leading algorithms for interactive segmentation in 2D and 3D [12, 9, 11]. Concurrently, the resolution of digital cameras and medical imaging scanners continues to increase at a rapid pace, e.g., it is not uncommon for a present-day Computed Tomography (CT) scanners to produce volume data of more than 100 million voxels. Although the algorithm presented in [4] operates at an interactive speed for lower-resolution digital photographs, its use for interactive segmentation of higher-resolution images and volumes is limited both by the intense memory requirements and the supralinear time complexity.

In the present work, we explore the use of a multilevel heuristic from the level sets literature to produce the high-quality segmentations of graph cuts, while drastically reducing the computational burden. Specifically, we perform graph cuts on a low-resolution image/volume and propagate the solution to the next level by only computing the graph cuts at that level in a *narrow band* surrounding the projected foreground/background interface. Since the algorithm is run only on the subgraph that comprises the narrow band, the additional computation required at the fine resolution is significantly less than running it on the full graph. Additionally, since weights need only be stored for the coarse resolution and the interior of the fine-resolution bands, the memory requirement is also significantly less. For higher-order connectivity of an image graph, this savings can be enormous.

Multilevel approaches have a long history of exploration in the context of Markov Random Fields (MRFs), of which graph cuts is a special case. Although there is a vast literature on this subject (see [10] for a review), the problems remain: Given conditional probabilities between sites at a fine level, how does one construct conditional probabilities at a coarse level? Given a solution at a coarse level, how can this be used to find a solution at the fine level? In the context of graph cuts, the sites are graph nodes (i.e., pixels or voxels) and the between-site probabilities translate to edge weights. In other words, given a fine-level, weighted graph, how does one construct weights for a coarse-level graph? Moreover, once a solution is obtained for a coarse-level graph, how may that solution be used to obtain a fine-level solution? Exact answers to these questions were given by Gidas' work [5] based on renormalization group theory from statistical mechanics. Unfortunately, the coarse-level graphs require enough additional edges that the benefit of reducing to a coarser grid is reduced or eliminated. A different type of approach was taken by Krishnamachari and Chellappa [8] where the coarse-level parameters were estimated by minimizing the Kullback-Leibler distance between local conditional distributions and the solution was propagated from a coarse level to a fine level as an initialization for a local optimization method. However, in the context of graph cuts, which is a global optimization approach, such an initializa-

tion is not relevant. For this reason, we chose to modify the graph by removing all nodes outside of a narrow band instead of using such an initialization scheme. Presegmentation on a fine-level image may also be used by coarse-level graphs. In the recent work from Li, Sun, Tang and Shum [9], the graph cuts technique is used on watershed regions. Hence solving the coarse-level graph will immediatly lead to a fine-level segmentation. However having a fast solving of a coarse-level graph is at the expense of a watershed precomputation that could be very costly. Also, giving that the watershed algorithms result in an unpredictable number of regions, it is not clear what the gain of speed and the reduction of memory consumption could be for large images and volumes.

We take the position in this work that the quality of graph cuts segmentation has now been well-established. Therefore, our aim is to give details of our multilevel banded approach and verify empirically that the computational burden is dramatically reduced while maintaining the segmentation quality of graph cuts. Additionally, we seek to provide the reader with an understanding of when to expect our heuristic to fail and when we should expect it to perform well. Section 2 develops our method and provides implementation details. In Section 3 we compare the computational speed and memory requirements of our approach to standard graph cuts. Next, we apply this hierarchical algorithm to a set of shapes with increasingly complex boundary in order to provide the reader with an intuition of when to expect failure of this heuristic. Finally, segmentation results on a set of 2D and 3D images are presented in Section 4. A conclusion follows in Section 5.

2. Multilevel Banded Graph Cuts

2.1. Image Segmentation by Graph Cuts

We begin by briefly summarizing the Boykov and Jolly's graph cuts algorithm to N-D image segmentation [3]. Note that in this paper we use the term "segmentation" for its meaning of delineating a boundary of one or multiple objects from images rather than the meaning of partitioning images into disjoint regions.

An N-D image can be specified by a pair (P, I) consisting of a finite discrete set P of N-D points (pixels in R^2 and voxels in R^3), and a mapping I that maps each point p in P to a value $I(p)$ in some arbitrary value space. From a given image, we can construct a weighted undirected graph $G = (V, E, W)$ that consists of nodes (vertices) $v \in V$, edges $e \in E$, and nonnegative weights (costs) $w \in W$. There are two special nodes in V: a source S node specifying the "object" terminal and a sink T node specifying the "background" terminal. The remaining nodes in V forms a subset $U = V/\{S, T\}$ where each node $u \in U$ uniquely identifies an image point in P. The set of edges E consists of two types of undirected edges: *n-links* (neighborhood links) and *t-links* (terminal links). Each image node $u \in U$ has two t-links $\{u, S\}$ and $\{u, T\}$ directly connected to the terminal S and T, respectively. However, n-links are completely determined by the neighborhood system used (e.g., 4- or 8-neighborhood system in 2-D, and 6-, 18-, or 26-neighborhood system in 3-D[1]).

The segmentation of an image into object and background, known also as the hard segmentation, can be expressed as a binary vector $A = (A_1, \cdots, A_u, \cdots, A_{|U|})$, where the element A_u gives the binary segmentation label of an image point identified by the node u. Boykov and Jolly [3] show a segmentation A can be uniquely determined by a cut C on the graph G, where the cut C is defined as a subset of edges in E such that the terminals become separated on the induced graph $G(C) = (V, E/C)$. Hence, the image segmentation problem can be solved as a minimum graph cut problem on the following energy function

$$\hat{C} = \arg\min_{C \in \mathcal{F}} \sum_{e_{ij} \in C} w_{ij}, \tag{1}$$

where e_{ij} denotes the edge e spanning between the nodes $v_i, v_j \in V$, w_{ij} denotes the weight assigned to the edge e_{ij}, and $\mathcal{F}$ denotes the set of all feasible cuts.

Assume that O and B denote the subsets of image nodes marked as "object" and "background" seeds by the user. Then the weight w_{ij} for the graph is given by [2]

$$w_{ij} = \begin{cases} \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right) / \text{dist}(u_i, u_j) & u_i, u_j \in U, \\ MAX & u_i \in O, u_j = S, \\ MAX & u_i \in B, u_j = T, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where $\text{dist}(u_i, u_j)$ is the Euclidean distance between image points p_i and p_j, identified by nodes u_i and u_j, respectively, $I_i = I(p_i)$, $I_j = I(p_j)$, and MAX is a very large positive constant. This energy penalizes cuts that pass through homogeneous regions and encourages cuts that pass through places where intensity discontinuity is large. The constant parameter σ can be either chosen empirically or estimated as a standard deviation over an image sample.

One of the most desirable properties of the graph cut algorithm compared to other energy minimization techniques is that the global minimum of the above energy function can be computed efficiently using a polynomial complexity algorithm. In this paper, we use the recent max-flow implementation proposed by Boykov and Kolmogorov [4][3]

[1] Larger neighborhood systems typically yield better image segmentation results but at the expense of both increased running time and memory consumption.

[2] Here, we used a simplified form that contains the boundary term only.

[3] Boykov and Kolmogorov's max-flow implementation is publicly available at http://www.cs.cornell.edu/People/vnk/software.html.

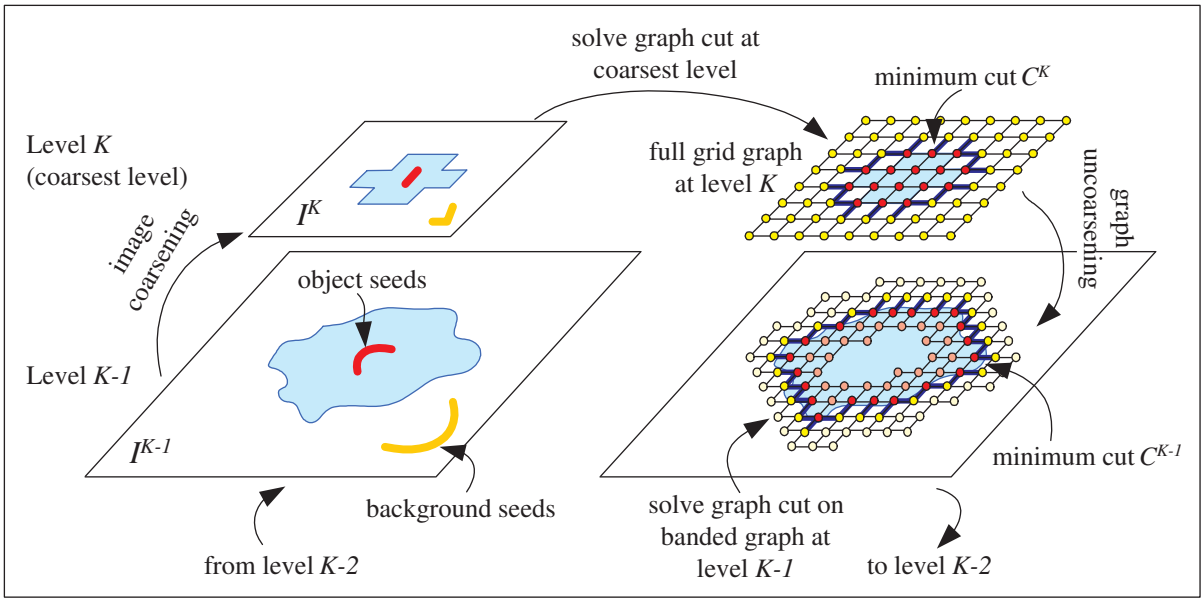


Figure 1. Multilevel banded graph cuts algorithm

that has been shown to perform several times faster than standardmax-flow implementations for the task of image segmentation.

2.2. Multilevel Banded Approach

Despite the fact that the graph cuts technique provides a powerful tool for image segmentation, the speed and memory consumption constrain its feasibility in many applications where large data sets need to be processed. For example, the memory allocation for the graph construction in Boykov and Kolmogorov's max-flow implementation needs $24|V| + 14|E|$ bytes. To segment a typical CT volume of size 512^3 in a medical application, the memory consumption is more than 8GB, obviously impractical for today's clinical computers. Moreover, finding the minimum cut for a graph of such large size is prohibitive due to the polynomial worst case complexity.

Inspired by both the multilevel graph partition method [6] and the well-known narrow band algorithm in level set methods [1], we propose a new segmentation algorithm that first solves the graph cut on the coarsest level graph and then solves the graph cut at successive higher resolution but on a narrow banded graph derived from the minimal cut found at the previous coarser graph. The algorithm is illustrated in Figure 1. This multilevel banded approach makes it possible to achieve high quality segmentation results on large data sets with faster speed and less memory consumption, thus allow it to be used in a wider range of medical applications as well as in other practical applications where high performance segmentation of large image data sets is crucial.

Our proposed multilevel banded graph cuts method consists of three stages: *coarsening*, *initial segmentation*, and *uncoarsening* similar to those existing multilevel graph partition methods (cf. [6, 2]). However, if the same strategy of multilevel graph construction is applied to design a multilevel graph cut algorithm, the memory consumption is not reduced because the original graph is still needed to start the coarsening process and refine the segmentation in the end. Therefore, we favor the strategy of coarsening directly on images using a standard multiresolution image technique as the original image is almost always needed in the memory for practical applications, hence its memory consumption is typically not considered part of the segmentation algorithm overhead.

During the coarsening stage, a sequence of smaller images $\{I^0, I^1, ..., I^K\}$, are constructed from the original image I^0 such that the size constraint $M_n^k \leq M_n^{k-1}$ is satisfied for each dimension $n = 1, \cdots, N$ and each level $k = 1, \cdots, K$, respectively. Note that our constraint does not require the size in each dimension to be reduced simultaneously. In additional to image coarsening, the location of the object and background seeds identified by O and B are also coarsened as well. However, extra care is required when coarsening the seeds, the seed coarsening operator must satisfy the *causality* constraint that the discrete topology [7] of both object and background seed regions is preserved throughout all levels, i.e., the number of connected object and background seed regions must be preserved. As a result, different coarsening operators should be separately chosen for coarsening image and seeds, re-

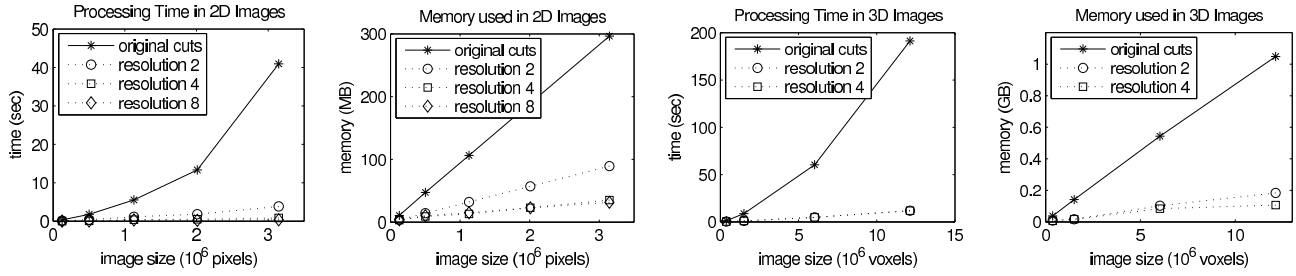


Figure 2. Speed and memory usage compared to standard graph cuts. The processing time and memory usage are shown in (a) and (b) for 2D images, and (c) and (d) for 3D volumes. Our approach significantly reduces the processing time and memory consumption.

spectively. In this paper, we coarsen an image using either a simple weighted mean filter followed by a downsampling of 2 operation or simply a downsampling of 2 operation, which has been found to yield good empirical results. An ad-hoc seed location coarsening operator is chosen such that the causality constraint is satisfied.

The second stage is the initial segmentation of the coarsest image I^K. We first construct a coarse graph $G^K = (V^K, E^K, W^K)$ for I^K according to Section 2.1 and then solve the minimum cut C^K quickly on the coarse graph G^K. This minimum cut yields a segmentation on the image I^K.

During the uncoarsening stage, we construct a binary boundary image J^k to represent all the image points that are identified by the nodes in the cut $C^k, k \in \{1, \cdots, K\}$, and project them onto a higher resolution boundary image J^{k-1} at level $k-1$ using an image uncoarsening operator. It is worth noting that the uncoarsening operator need not be the dual operator of the image coarsening operator used in the first stage due to the binary nature of the boundary image. In this paper, we use the uncoarsening operator defined as follows:

$$J^{k-1}(p) = J^k(\alpha(p)), \tag{3}$$

where $p = (p_1, p_2, \cdots, p_N)$ is an N-D point and $\alpha(p) = (\alpha_1(p_1), \alpha_2(p_2), \cdots, \alpha_N(p_N))$ is the reduction mapping used in the coarsening phase to reduce the dimension size under the size constraint.

The resulting boundary image J^{k-1} contains a narrow band that bounds the candidate boundaries of objects to be extracted from I^{k-1}. The width of the band could be controlled by an optional dilation of the band by a distance $d \geq 0$. The dilation distance parameter plays an important role in practice. If d is small, the algorithm may not be able to recover the full details of objects with high shape complexity or high curvature. On the other hand, if d is large, the computational benefits of banded graph cuts will be reduced and the wider band may also introduce potential outliers far away from the desired object boundaries. In our implementation, we found that choosing $d = 1$ gives good compromise between accuracy and performance for most of the real-world 2D and 3D images. The graph G^{k-1} is constructed as follows:

- Construct a banded graph G^{k-1} using only nodes inside the band from the boundary image J^{k-1},
- Use the band's outer layer as the new background seeds B and the band's inner layer as the new object seeds O. In the degenerated case, where the band contains no inner layer due to either segmenting small objects or using large band width, we choose to use the coarsened object seeds at level $k-1$ as the object seeds O. It can be shown that coarsened object seeds are guaranteed to lie inside objects to be segmented due to the way our algorithm is constructed, and
- Assign the weights of all edges according to (2).

Once the graph G^{k-1} is constructed, we can solve the minimum cut C^{k-1} on G_{k-1} and then repeat the same uncoarsening procedure recursively at the next level until the minimum cut C^0 is solved on the banded graph G^0, yielding the final segmentation result.

Note that all other graphs at levels $k = 0, \cdots, K-1$ have banded graph structure except the graph G^K, which is significantly smaller than the full grid graph constructed for the image at the same level. Because we use a much smaller graph in all resolutions, both the run time and the memory consumption of the algorithm is considerably reduced compared to the single graph cut algorithm, resulting in a significantly accelerated segmentation process (typically on the order of a magnitude in our experiments).

3. Banded vs. Conventional Graph Cuts

This section compares the performance of multilevel banded graph cuts and conventional graph cuts [3] in terms

of speed, memory usage, and segmentation accuracy. The ability of capturing high frequency structures is showed through an experiment.

3.1. Improvement on Speed and Memory Usage

The experiments are performed by segmenting a set of 2D images and 3D volumes of different sizes using the multilevel banded graph cuts and the conventional graph cuts. The tests are performed on a computer with a Pentium 4 2.4GHz CPU with 2GB RAM. In two level banded graph cuts, we tested different downscaling factors of 2, 4, and 8. We use the same foreground and background seed positions by projecting them from the original image to the low resolution image. The graph connectivity in this experiment is 4 for a 2D image and 6 for a 3D volume.

Figure 2(a) shows that our approach is 8 times as fast as the original graph cuts algorithm using a downscaling factor of 2. Figure 2(b) shows that it consumes 4 times less memory. This result is expected because the memory allocated for a low resolution graph is 4 times as small as the original graph and the memory is freed before new memory is allocated for the banded graph. The processing time of the multilevel banded graph cuts includes standard graph cuts applied on the low resolution level and the banded graph cuts. Figures 2(c,d) show similar result performed on 3D volumes. At a downscaling factor of 2, our approach consumes 8 times less memory than standard graph cuts.

3.2. Segmentation Accuracy

We used a set of real world pictures (5 sample images of total 15 images are shown in Figure 3) to compare our segmentation results with standard graph cuts. In each picture, we randomly place an object seed in the brighter region and a background seed in the darker region. By brighter region or darker region we mean that the intensity is higher or lower than the average intensity of the picture. Considering the standard graph cuts segmentation results as the ground truth, the multilevel banded graph cuts obtained the average under-segmentation and over-segmentation ratios of 1.60% and 2.64%.

We made an experiment to show how our algorithm performs with different downscaling factors on images with various shape complexities. A flower-like picture is modified so that the amplitude of petals are increased and the original version of the flower is a disk. Multilevel banded graph cuts with downscaling factors of 2, 4, 8, and 16 are applied on these images. Table 1 shows the result of this experiment. The first column of Table 1 is the result of the standard graph cuts algorithm. In subsequent columns, the downscaling factor is increased. Even at the highest complexity, both under-segmentation and over-segmentation ratios are low except when the downscaling factor is 16. From our experiments, a downscaling factor of 2 or 4 is a practical choice for most images.

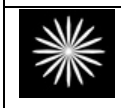
Resolution				
1	2	4	8	16
				
original	0% 0%	0% 0%	0% 0%	0% 0%
				
original	0% 0%	0% 0%	0% 0%	0% 0%
				
original	0% 0%	0% 0%	0% 0%	0.02% 0.05%
				
original	0% 0%	0% 0%	0.01% 0.02%	0.20% 0.33%
				
original	0% 0.05%	0.01% 0.31%	0.07% 1.10%	34.43% 2.43%

Table 1. Robustness to high complexity object. Under-segmentation and over-segmentation ratios are listed. Each column corresponds to a different downscaling factor of the low resolution graph.

4. Segmentation Results

In this section, we show the segmentation results of our approach for different type of images. Figure 3 shows the interactive segmentation results of four 2D pictures using the multilevel banded graph cuts. The first row shows the original images with the superimposed seed points. Yellow strokes are for the object and blue strokes are for the background. The second row shows the segmentation results of standard graph cuts, where the cuts are displayed in white. The third rows shows the segmentation results of multilevel banded graph cuts, which achieves similar results but considerably reduces the processing time and memory usage.

Figures 4(a,b) show a heart segmentation from a 3D

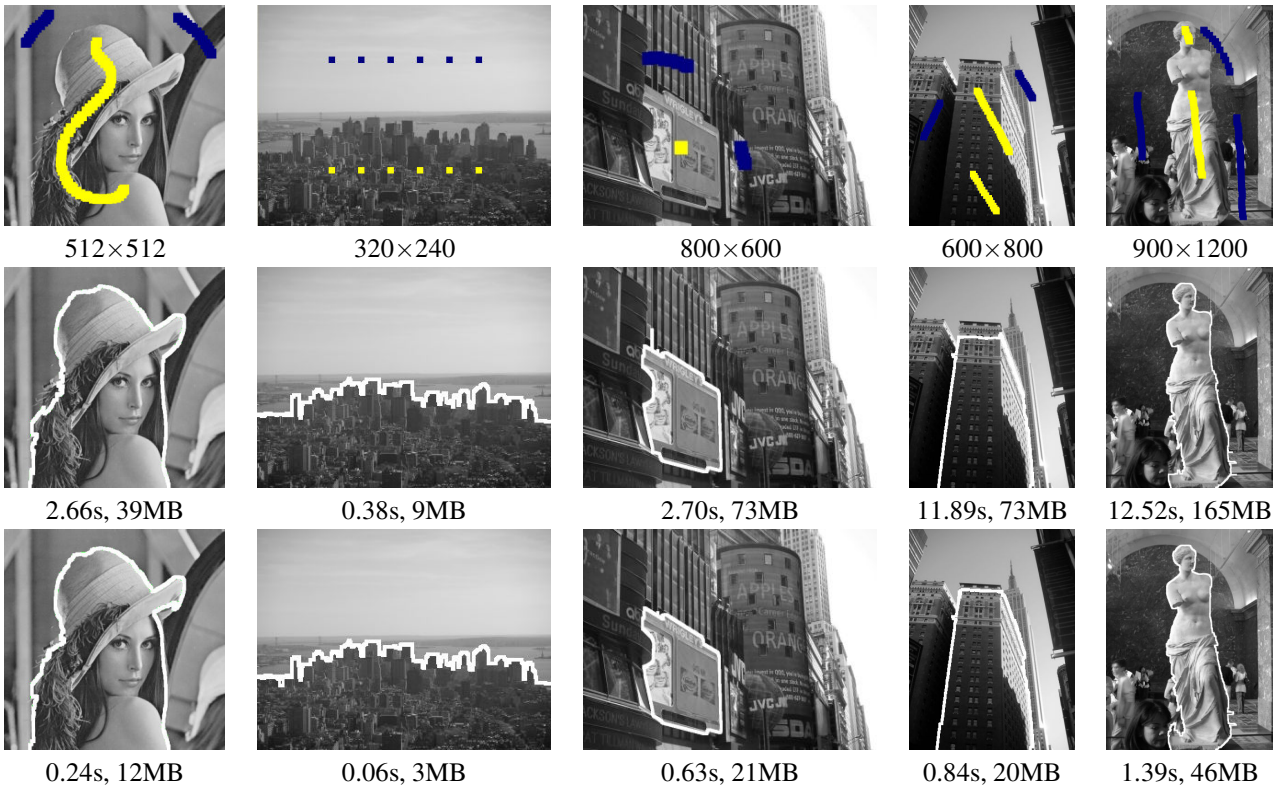


Figure 3. Graph cuts vs. multilevel banded graph cuts on 2D images. The first row shows the original images with superimposed user input (yellow strokes for foreground and blue strokes for background). The second row shows the standard graph cuts results. The third row shows that multilevel banded graph cuts provides similar results but with higher speed and less memory usage.

CT volume of size 256×256×185. The graph cuts algorithm requires 192.27s to segment and consumes 1024MB of memory. With our approach, the segmentation of the same region with the same inputs requires 11.85s and uses 186MB memory. The results are qualitatively similar to the standard graph cuts result. Figures 4(c,d) show a pulmonary artery segmentation from the same CT volume. The graph cuts algorithm requires 63.78s to segment and consumes 1010MB of memory. With our approach, the segmentation of the same region with the same seeds requires 8.63s and uses 165MB memory. Differences occur in high frequency regions as shown in Figure 4(d) where small vessels of the pulmonary artery are under-segmented using the multilevel banded graph cuts. Figures 4(e,f) show a lung segmentation from a CT volume of size 256×256×216. The graph cuts algorithm requires 68.46s to segment and consumes 1236MB of memory. With our approach the segmentation of the same region with the same seeds requires 14.43s and uses 239MB of memory. The results are almost identical in both algorithm results as shown in Figure 4(e) and (f).

5. Conclusions

We have presented a heuristic that provides a fast, memory-efficient, algorithm that produces nearly the same results as the popular graph cuts segmentation technique. Time and memory efficiency were compared with conventional graph cuts, as well as a qualitative and quantitative examination of the segmentation accuracy. We found that use of even a small number of levels in the hierarchy resulted in an increase in time and memory efficiency of several orders of magnitude. Finally, we offer the reader a characterization of the type of segmentation target that we expect the heuristic to perform on. Specifically, we find that our heuristic achieves very similar results compared to conventional graph cuts, unless the boundary of the segmentation target is "spiky" and the number of levels in the hierarchy is large. If either the target has a smooth boundary or a conservative number of levels are used in the hierarchy, a near-perfect result is obtained.

Future work includes an attempt to set coarse-level weights that more accurately reflect the underlying fine-

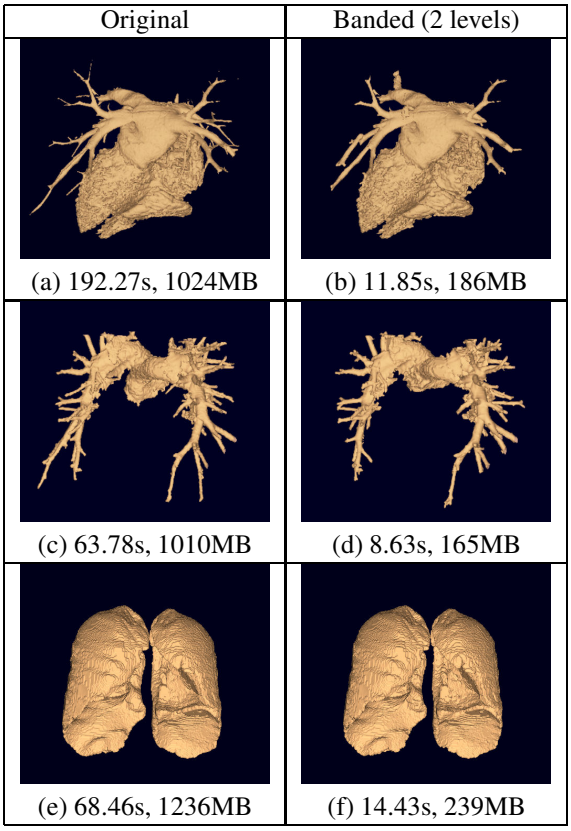


Figure 4. Graph cuts vs. multilevel banded graph cuts on 3D images. Heart segmentation using standard graph cuts (a) and the multilevel banded graph cuts (b). Pulmonary artery segmentation using the graph cuts (c) and the multilevel banded graph cuts (d). Lung segmentation using the graph cuts (e) and the multilevel banded graph cuts (f).

level graph and to explore methods for using the projected fine-level solution as a starting point for finding the global optimal of the graph cuts functional.

Acknowledgment

The authors would like to specially thank both Yuri Boykov and Marie-Pierre Jolly for their helpful discussions concerning this work, and the support from Frank Sauer, Gareth Funka-Lea, and James Williams at Siemens Corporate Research.

References

[1] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1995.

[2] A. Barbu and S.-C. Zhu. Multigrid and multi-level Swendsen-Wang cuts for hierarchic graph partition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 731–738, 2004.

[3] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision*, volume 1, pages 105–112, July 2001.

[4] Y. Boykov and V. 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, Sept. 2004.

[5] B. Gidas. A renormalization group approach to image processing problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):164–180, Feb. 1989.

[6] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998.

[7] T.Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 48:357–393, 1989.

[8] S. Krishnamachari and R. Chellappa. Multiresolution Gauss Markov random field models. *IEEE Transactions on Image Processing*, 6:251–267, February 1997.

[9] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. In *Proceedings of ACM SIGGRAPH 2004*, volume 23, pages 303–308. ACM Press, April 2004.

[10] P. Pérez and F. Heitz. Restriction of a Markov random field on a graph and multiresolution statistical image modeling. *IEEE Transactions on Information Theory*, 42(1):180–190, January 1996.

[11] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut": Interactive foreground extraction using iterated graph cuts. In *Proceedings of ACM SIGGRAPH 2004*, volume 23, pages 309–314. ACM Press, April 2004.

[12] N. Xu, R. Bansal, and N. Ahuja. Object segmentation using graph cuts based active contours. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 46–53, 2003.