Weights and Topology: A Study of the Effects of Graph Construction on 3D Image Segmentation

Leo Grady and Marie-Pierre Jolly

Siemens Corporate Research — Dept. of Imaging and Visualization
755 College Rd. East, Princeton, NJ 08540

Abstract. Graph-based algorithms have become increasingly popular for medical image segmentation. The fundamental process for each of these algorithms is to use the image content to generate a set of weights for the graph and then set conditions for an optimal partition of the graph with respect to these weights. To date, the heuristics used for generating the weighted graphs from image intensities have largely been ignored, while the primary focus of attention has been on the details of providing the partitioning conditions. In this paper we empirically study the effects of graph connectivity and weighting function on the quality of the segmentation results. To control for algorithm-specific effects, we employ both the Graph Cuts and Random Walker algorithms in our experiments.

1 Introduction

Graph-based algorithms have become well-established tools for general image segmentation problems [1,2,3]. The procedure underlying these algorithms is to: 1) Identify each pixel with a node of the graph, 2) Assign an arbitrary edge set (connectivity), 3) Use the image content to establish a set of weights on the edges, 4) Establish a partitioning criterion that may be optimized to produce a segmentation. Distinctions in the fourth step separate different graph-based segmentation algorithms. Although some attention has been paid to the first step (by associating nodes with presegmented regions), the second and third steps have been almost entirely ignored. Specifically, these steps have typically employed the same set of heuristics. Our goal in this work is to empirically study these common heuristics and determine which, if any, work best on real data.

The seeded user interface employed by many of the graph algorithms supports interactive segmentation for which the segmentation target is chosen by the user and not known *a priori* by the algorithm designer. In these cases, a small number of intensity-based edge weighting functions seem to reoccur throughout very different segmentation algorithms. Since the same weighting functions consistently reappear throughout the graph-based segmentation literature, there appears to be an unwritten assumption that the utility of these weighting functions is *independent* of the specific algorithm in use. This assumption implies that the results of testing different weighting functions with any graph-based segmentation algorithm will support valid conclusions about the weighting function that apply to all graph-based algorithms.

Edge connectivities of the image lattice have been treated similarly in the graph-based literature. In most cases, a 6-connected, 10-connected or 26-connected lattice

D. Metaxas et al. (Eds.): MICCAI 2008, Part I, LNCS 5241, pp. 153–161, 2008.

have been employed, without much discussion of why one choice was made over another. A common feeling in the community seems to be that higher levels of connectivity do, in fact, improve the segmentation results. In rare cases, such as the Graph Cuts algorithm, this assertion is also predicted theoretically [4]. Unfortunately, the additional overhead of more edges usually results in a decrease of performance speed and therefore less-connected edge topologies are sometimes preferred.

In this work, we empirically study the effects of weighting function and graph topology on the performance of segmentation algorithms. We make the simplifying assumption that the performance effects of weighting function and graph topology are independent. Although this assumption is not likely to be strictly true, we are unaware of any claim in the literature that the weighting functions (or even weighting function parameters) should be paired with particular edge topologies. Since the assignments of weightings/topologies reoccur between different graph-based segmentation algorithms, it seems to be assumed that the weighting/topology choice is independent of the specific graph-based algorithm. Although this assumption suggests that the choice of graph-based algorithm should not bias the findings on the utility of a particular weighting function or graph topology, we employ two graph-based algorithms to control for any algorithm-specific bias toward a particular weighting function or edge connectivity. In this work, we have chosen to employ the Graph Cuts [1] and Random Walker segmentation algorithms [2] to perform our tests.

2 Method

We obtained 62 3D medical datasets containing a single segmentation target that were manually segmented by a clinical practitioner. Each volume was also given manually-placed foreground and background seeds by the same clinical practitioner that provided the manual segmentation. The data contained a range of segmentation targets including tumors, lymph nodes, cysts and other lesions. The data was acquired using different Siemens computed tomography (CT) scanners, with different reconstruction kernels and the clinical input (ground truth and seeds) was given by different clinical partners. Therefore, our results should not be biased by the details of a particular acquisition protocol or clinical individual. The datasets we used for segmentation were typically cropped from larger data acquisitions and ranged in size from roughly $40 \times 40 \times 40$ to $128 \times 128 \times 128$. Most of the datasets had different numbers of voxels in each dimension (i.e., they were not cubes) and had a greater spacing between axial slices than within the slice. All of the data acquisitions were axial scans. The XY-plane was chosen to correspond to an axial slice.

Data from CT acquisitions is sometimes considered to be easier to work with, due to the reliability of the output intensities, than other imaging modalities (e.g., ultrasound, magnetic resonance). However, our purpose in this work is not to examine the absolute performance of an algorithm to the segmentation of these targets. Instead, our goal is to compare the relative segmentations obtained through the use of different weighting functions and graph topologies in otherwise controlled conditions. The choice to use this series of CT data was made primarily due to the availability of this data in sufficient quantity to produce meaningful results.

Due to the good segmentation performance and widespread usage, we chose to employ the Graph Cuts algorithm of [1] and the Random Walker segmentation algorithm of [2]. These algorithms are representative examples of the set of modern, graph-based segmentation algorithms that input foreground/background seeds and output a label for each voxel. In order to assign a label to an unseeded voxel, the Graph Cuts algorithm computes the minimum cut separating the foreground from background seeds using a max-flow/min-cut computation [5]. In contrast, the Random Walker algorithm computes the probability that a random walker initiating its walk at each voxel first arrives at a foreground seed before arriving at a background seed. If that probability is larger than 0.5, then the voxel is labeled as foreground (otherwise it is labeled as background). It was shown in [2] that these probabilities could be efficiently computed by solving a sparse system of linear equations. In these experiments, the system of linear equations was solved iteratively using the preconditioned conjugate gradient method with a Jacobi preconditioner and solved to the same level for all examples.

Evaluation of segmentation quality with respect to ground truth is a delicate problem. In this study, we chose to employ the volume overlap and the normalized volume difference [6]. The volume overlap is generally more meaningful as a metric of segmentation quality, since it takes into account the relative position of the ground truth and the computed segmentation.

2.1 Weighting Functions

Weighting functions have been used to map intensity gradients to graph weights since at least as early as the influential work of Perona and Malik on anisotropic diffusion for image smoothing [7], and earlier image reconstruction efforts [8]. When introducing anisotropic diffusion, the authors suggested two functions used to map intensity changes to diffusion constants. These two functions were subsequently studied and determined to reflect differing models of image formation [9]. Since this time, these two functions have been employed in the Normalized Cuts algorithm [10] and subsequent graph-based segmentation algorithms [1,2,3,11]. Although these functions have generally yielded good segmentation performance, we are aware of no attempt to carefully compare the quality of results obtained with these functions. Since natural images are known to contain significant structure (and medical images presumably even more), it is not unreasonable to think that one of these weighting functions better models how to convert the image inputs into graph weights. The weighting functions initially proposed in [7] but subsequently utilized throughout the segmentation literature are

$$\text{Gaussian}: \; w_{ij} = \frac{1}{\text{dist}(v_i, v_j)} \exp(-\beta(g(v_i) - g(v_j))^2), \tag{1}$$

$$\text{Reciprocal}: \; w_{ij} = \frac{1}{\text{dist}(v_i, v_j)} \frac{1}{1 + \beta(g(v_i) - g(v_j))^2}, \tag{2}$$

where $g(v_i)$ indicates the image intensity at voxel v_i and β represents a free parameter. The function $\text{dist}(\cdot)$ accounts for differences in spacing and edge length and is computed as the Euclidean distance between voxels, taking into account voxel spacing.

The algorithms under consideration contain additional information in the form of the intensity distribution at the seeds. A natural idea is to make an assumption that the

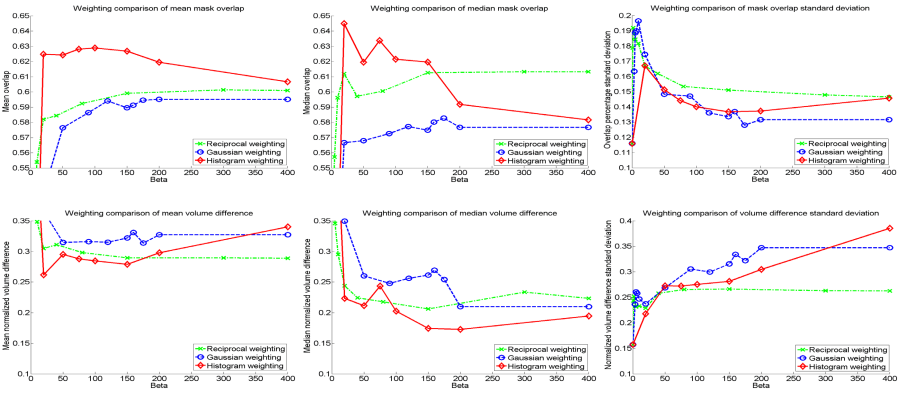

Fig. 1. Graph Cuts: Empirical comparison of weighting functions over a range of β values using the same connectivity. Marker positions indicate measured data points. Blue/dashed line with circles: Gaussian function (1), Green/dash-dot line with crosses: Reciprocal function (2), Red/solid line with diamonds: Histogram-based function (3). Top row: Comparison of mean, median and standard deviation for mask overlap measure. Bottom row: Comparison of mean, median and standard deviation for normalized volume difference.

intensities in the foreground voxels are all drawn from the same intensity distribution. This intensity distribution may be estimated using a Parzen window on the histogram of intensities contained in the foreground seeds. Given this estimation of the foreground intensities, we can look for boundaries of the foreground object with the function

$$\text{Histogram}: \; w_{ij} = \frac{1}{\text{dist}(v_i, v_j)} \exp(-\beta(H(g(v_i)) - H(g(v_j)))^2), \tag{3}$$

where $H(g(v_i))$ denotes the probability that image intensity $g(v_i)$ at voxel v_i is drawn from the foreground object.

For each weighting function, a range of βs was tested. In order to remove any bias for the absolute intensities of the image, the gradients were normalized by the largest gradient (in each dataset) to lie in the interval $[0, 1]$ before applying the above weighting functions. Due to numerical precision or choice of parameter, it might be possible to assign a weight to be exactly zero. To account for this possibility, a small additive constant (equal to $1e^{-6}$) was added to each weight. All experiments comparing the weighting functions were conducted using a 6-connected lattice.

2.2 Graph Topology

In 3D computer vision, the standard graph connectivities are: 6-connected, 10-connected and 26-connected. The edge set of each connectivity is defined as

$$6-\text{connected}: \quad E = \{i, j| \;\; ||C(v_i) - C(v_j)|| \leq 1\}, \tag{4}$$

$$26-\text{connected}: \quad E = \{i, j| \;\; ||C(v_i) - C(v_j)|| \leq \sqrt{3}\}, \tag{5}$$

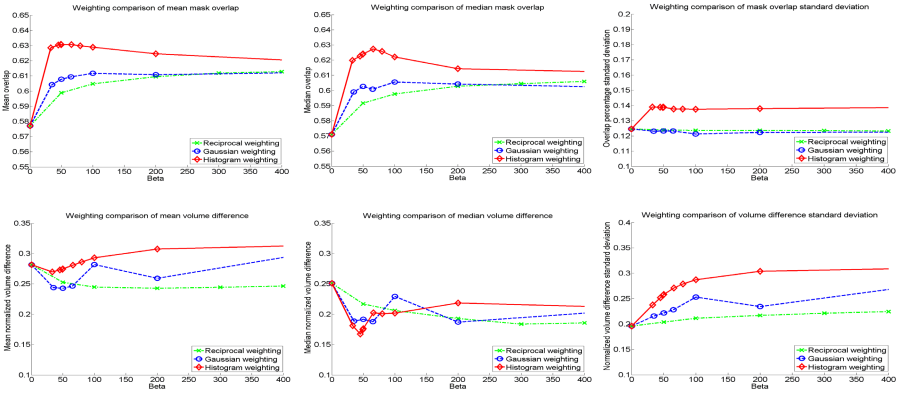

Fig. 2. Random Walker: Empirical comparison of weighting functions over a range of β values using the same connectivity. Marker positions indicate measured data points. Blue/dashed line with circles: Gaussian function (1), Green/dash-dot line with crosses: Reciprocal function (2), Red/solid line with diamonds: Histogram-based function (3). Top row: Comparison of mean, median and standard deviation for mask overlap measure. Bottom row: Comparison of mean, median and standard deviation for normalized volume difference.

where $||\cdot||$ is used to denote the standard Euclidean norm and $C\left(v_i\right)$ maps voxel v_i to its coordinates in 3D. The 10-connected case has a somewhat more complicated definition, since it gives preferential treatment to the within-slice dimensions (taken here to be the XY-plane). We may define a 10-connected lattice to be

$$10-\text{connected}: \quad E = \{i,j| \ ||C(v_i) - C(v_j)|| \leq 1\} \cup \\ \{i,j| \ ||C(v_i) - C(v_j)|| \leq \sqrt{2}, \forall C(v_i)_z = C(v_j)_z\}. \quad (6)$$

We make the assumption that edge connectivity and weighting function are independent design choices, i.e., if a particular connectivity improves the results, we assume that the performance increase will persist even if another weighting function is employed. Since the histogram-based weighting function (3) performed better than those based purely on image intensity (see Section 3), the same histogram-based weighting function was employed across all connectivity experiments.

3 Results

3.1 Weighting Functions

Our comparison of graph weighting functions is displayed in Figure 1 for Graph Cuts and in Figure 2 for the Random Walker algorithm. These Figures plot the mean, median and standard deviation values of the two segmentation measures across β values.

Graph Cuts responded to a different parameter range for β in the two intensity-based weighting functions than the Random Walker algorithm. For better display of Figure 1,

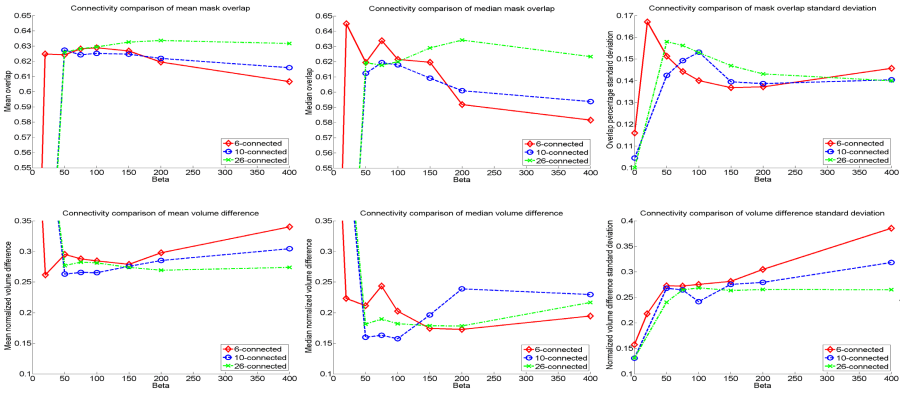

Fig. 3. Graph Cuts: Empirical comparison of graph connectivities over a range of β values using the same weighting function. Marker positions indicate measured data points. Red/solid line with diamonds: 6-connected lattice. Blue/dashed line with circles: 10-connected lattice, Green/dash-dot line with crosses: 26-connected lattice. Top row: Comparison of mean, median and standard deviation for mask overlap measure. Bottom row: Comparison of mean, median and standard deviation for normalized volume difference.

the β value used in the Gaussian function is $10\times$ the number shown on the axis, and the β value used for the Reciprocal function is $100\times$ the axis value.

Despite some differences in the behavior of both algorithms in the presence of different weighting functions, the overall behavior response is similar. Not surprisingly, the histogram-based weighting function outperforms both of the weighting functions based on intensity difference in both algorithms. However, the standard deviations of the histogram-based results were higher than the standard deviations from the intensity-based weighting functions for Random Walker, although the standard deviations across all three weighting functions were similar for Graph Cuts. All of the weighting functions appear to take a single peak at a certain β value, although the location of this peak is function-dependent. The comparison of the Gaussian and Reciprocal weighting functions, both based purely on intensity differences, is revealing. Although the Gaussian weighting function appears to be more prevalent in recent graph-based segmentation literature, the Reciprocal weighting function appears to outperform the Gaussian weighting function in two respects. For both algorithms (although more dramatically for Graph Cuts), the Reciprocal weighting function achieves an absolute higher performance with respect to both measures and a substantially lower standard deviation with respect to the volume difference measure. Additionally, the performance of the Gaussian weighting function behaves more erratically than the Reciprocal weighting function with respect to changes in β. Moreover, the peak performance of the Gaussian weighting function corresponds to a narrow range of β values, while the peak performance of the Reciprocal weighting function persists over a much broader range of β values. This finding suggests that a designer must be more selective with their choice of β when using the Gaussian weighting function than when employing the Reciprocal weighting function.

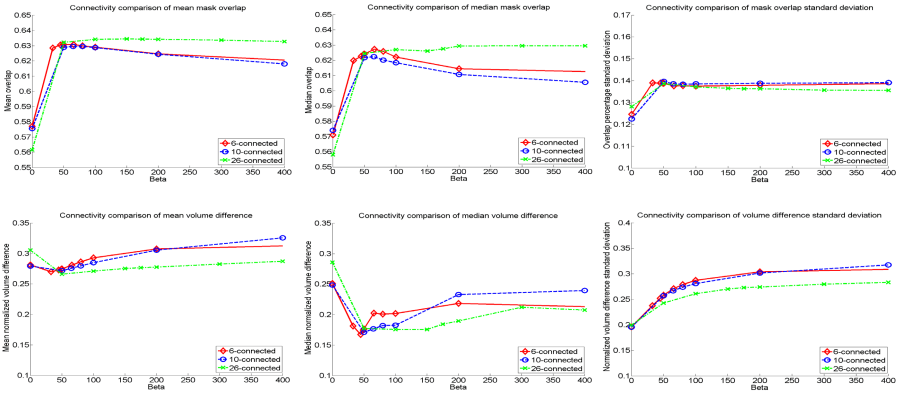

Fig. 4. Random Walker: Empirical comparison of graph connectivities over a range of β values using the same weighting function. Marker positions indicate measured data points. Red/solid line with diamonds: 6-connected lattice. Blue/dash line with circles: 10-connected lattice, Green/dash-dot line with crosses: 26-connected lattice. Top row: Comparison of mean, median and standard deviation for mask overlap measure. Bottom row: Comparison of mean, median and standard deviation for normalized volume difference.

The intensity-based weighting functions were outperformed by the histogram-based weighting function. The form of the histogram-based function was the same as the Gaussian weighting function, except that differences in "foreground probability" were used rather than the raw intensities. It is notable that the response of the histogram-based function and the Gaussian function with respect to β followed a similar evolution, except that the histogram-based function produced consistently better performance.

3.2 Graph Topology

The segmentation performance with respect to graph topology followed a similar pattern with both graph-based algorithms. Overall, the connectivity level seemed to have a stronger effect on the Graph Cuts results than the Random Walker results.

Conventional wisdom about graph-based algorithms tends to support the notion that more edges (stronger connectivity) produce better results. Figure 4 illustrates that this notion is not necessarily correct. Of the three graph topologies, the 10-connected graph appears to exhibit inferior performance to both the 6-connected and 26-connected graphs. One explanation for this phenomenon is that the asymmetry of the 10-connected lattice (i.e., preferential treatment of the XY-plane) introduced a negative bias into its performance. One might assume that it would be appropriate to include more edges on the within-slice plane, due to anisotropic voxel spacing. However, introducing more edges within-slice further reduces the percentage of between-slice edges, which may be responsible for the negative effect.

The 26-connected lattice gave the best performance of all three graph topologies. Use of a more extensive topology leading to better performance of graph algorithms has been previously predicted in the literature for Graph Cuts [4], although the effect

seems to be similar (albeit not as dramatic) for the Random Walker algorithm. Although the peak β value for the 6-connected and 10-connected lattices were roughly equal, the peak β value for the 26-connected lattice was larger. A possible explanation for this phenomenon is given by the fact that gradient normalization was performed over all edges. Since the between-slice diagonal edges present in the 26-connected lattice would be expected to be greater than the other gradients along the other edges, it is not unexpected that a larger β value would be required in the 26-connected case. This explanation suggests controlling for this phenomenon by using the same normalization for each dataset over all graph connectivities.

4 Conclusion

Graph algorithms have become very popular for 3D medical image segmentation. Although the action of each algorithm is different, the same procedures are consistently used to assign graph connectivity and edge weighting. Since these procedures persist across algorithms, there is an implicit assumption in the segmentation community that the algorithms respond similarly to the design specifications of the graph construction. In this work, differences in graph construction were controlled for algorithm-specific effects by employing both the Graph Cuts and Random Walker algorithms.

With respect to different graph weighting functions, it was found that basing the weights on differences in a probability density obtained from the foreground seeds was generally superior to weighting functions based solely on intensity gradients. This finding is not overly surprising, given that more information is being used to build the weight structure. However, it was more surprising to find that the Reciprocal weighting function, which has been employed less in recent years, outperforms the more popular Gaussian weighting function in terms of both absolute performance achieved and stability. This phenomenon was observed with both segmentation algorithms used in the study. Since the Reciprocal weighting function outperformed the Gaussian weighting function, it is possible that it would be more effective in the future to base the form of the histogram-based function on the Reciprocal weighting function.

Although different connectivities have been employed in 3D graph construction, little attention had been previously paid to the effects of topology choice. It has been previously predicted for the Graph Cuts algorithm that higher-order connectivities produce better algorithm performance [4]. Our study confirms this prediction and shows a similar response behavior for the Random Walker algorithm, but with a diminished influence of topology. Specifically, our study comparing graph construction confirms that 26-connected graphs do exhibit better performance than either 6-connected or 10-connected graphs. More surprising is that the 10-connected lattice performed worse than either the 6-connected or the 26-connected lattice. A possible explanation is that the asymmetry of the 10-connected lattice causes an unintended bias in the results.

Our experiments suggest that the best algorithm performance may be gained by using a 26-connected lattice and histogram-based weighting. If histograms are not available (or unreliable, due to small samples), the Reciprocal weighting function outperforms the Gaussian weighting function in both quality and stability.

There is much future work to be done on this topic. Three major questions remain: 1) Do these results persist with other data modalities? 2) Are better, general-purpose,

graph connectivities and weighting functions possible? This paper offers the first look at empirically evaluating the effects of graph construction on algorithm performance.

References

1. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proc. of ICCV 2001, pp. 105–112 (2001)
2. Grady, L.: Random walks for image segmentation. IEEE PAMI 28(11), 1768–1783 (2006)
3. Udupa, J.K., Samarasekera, S.: Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. GMIP 58(3), 246–261 (1996)
4. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proceedings of the International Conference on Computer Vision, vol. 1 (2003)
5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE PAMI 26(9), 1124–1137 (2004)
6. Jolly, M.P., Grady, L.: 3D general segmentation in CT. In: Proc. of ISBI, pp. 796–799 (2008)
7. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE PAMI 12(7), 629–639 (1990)
8. Geman, S., McClure, D.: Statistical methods for tomographic image reconstruction. Proc. 46th Sess. Int. Stat. Inst. Bulletin ISI 52, 4–21 (1987)
9. Black, M.J., Sapiro, G., Marimont, D.H., Heeger, D.: Robust anisotropic diffusion. IEEE TIP 7(3), 421–432 (1998)
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE PAMI, 888–905 (2000)
11. Falcão, A., Udupa, J., Samarasekera, S., Sharma, S., Elliot, B., de A. Lotufo, R.: User-steered image segmentation paradigms: Live wire and live lane. GMIP 60(4), 233–260 (1998)