

Higher-Dimensional Segmentation by Minimum-cut Algorithm

Hiroshi Ishikawa

Department of Information & Biological Sciences
Nagoya City University
Yamanohata, Mizuho, Nagoya 467-8501, Japan.
hi@nsc.nagoya-cu.ac.jp

Davi Geiger

Courant Institute of Mathematical Sciences
New York University
251 Mercer St., New York, NY 10012, USA.
geiger@cs.nyu.edu

Abstract

It is important in many applications of 3D and higher dimensional segmentation that the resulting segments of voxels are not required to have only one connected component, as in some of extant methods. Indeed, it is generally necessary to be able to automatically determine the appropriate number of connected components. More generally, for a larger class of applications, the segments should have no topological restrictions at all. For instance, each connected component should be allowed to have as many holes as appropriate to fit the data.

We propose a method based on a graph algorithm to automatically segment 3D and higher-dimensional images into two segments without user intervention, with no topological restriction on the solution, and in such a way that the solution is optimal under a precisely defined optimization criterion.

1 Introduction

A situation often occurs that a multi-dimensional array that assigns data to each multi-dimensional slot (voxel) is given and a need arises to partition the set of voxels into two or more subsets according to the data. For instance, three-dimensional arrays of data are obtained representing one or more physical properties at regular grid positions within the interior of solid bodies. Such data may be obtained by non-intrusive methods such as computed axial tomography (CAT) systems, by magnetic resonance imaging (MRI) systems, or by other non-intrusive mechanisms such as ultrasound, positron emission tomography (PET), emission computed tomography (ECT) and multi-modality imaging (MMI). Each of these techniques produces a planar, grid-like array of values for each of a succession of slices of the solid object, thus providing a three-dimensional array of such values. Typically, the solid object is a human body or a portion thereof, although the method is equally applicable to other natural or artificial bodies. In the case of CAT scanning, the physical value is the coefficient of x-ray absorption. For MRI, the physical values are the spin-spin and the spin-lattice relaxation echoes. In any event, the measured physical values reflect the variations in composition, density or surface characteristics of the underlying physical structures. Likewise, such three-dimensional arrays of interior physical values are utilized to generate visual images of the interior structures within the body. In the case of the human body, the visual images thus produced can be used for medical purposes such as diagnostics or for the planning of surgical procedures. In order to display two-dimensional images of such three-dimensional interior structures, however, it is necessary to locate the position of the boundary of such structure within the array of physical values. A

significant problem in displaying such internal surfaces is, therefore, the need to segment the data samples into the various tissues. This has been accomplished by simply deciding the structure to which each voxel belongs by comparing the data associated to the voxel to a single threshold value, or to a range of threshold values, corresponding to the physical property values associated with each structure or its boundary. Bones or any other tissue, for example, can be characterized by a known range of density values to which the array values can be compared. Such simple thresholding, however, is too susceptible to noise. That is, at the boundary, voxels with values near threshold can be swayed either way by a smallest noise, giving very noisy result. What is needed is to incorporate the tendency of nearby voxels to belong to the same partition. Domains of applications of segmentation other than medical applications include graphics, visualization tools, and reconstruction of 3D objects. In graphics, an object from an image are segmented. When there is a sequence of image (video), it can be considered a 3D image. Thus a segmentation of moving object from a video sequence is an application of 3D segmentation. Also, the data array is not limited to 3D. Higher dimensional applications include four-dimensional segmentation of a temporal sequence of 3D images, such as a 3D image of beating heart. It is important in many applications that the resultant sets of voxels are not restricted in the number of connected component. Indeed, it is generally necessary to be able to automatically choose the appropriate number of connected components. Moreover, for a larger class of applications, the subsets should have no topological restrictions at all. For instance, each connected component should be allowed to have as many holes as appropriate to fit the data.

2 Previous Work

Conventional methods have at least one of the following three shortcomings: they either i) have topological restrictions on the solution, ii) are not guaranteed to reach the optimal solution, or iii) need user help or intervention. Some methods presuppose the nature of the set to be found. For instance, if arteries are expected, some methods try to find one-dimensional object with some thickness, making it difficult to find bifurcating arteries. An algorithm that has desirable topological properties is suggested in [3], based on Level Sets method. Yet, it is a gradient-descent method with no guarantee to reach the optimal. Region Growing methods, similarly, have good topological properties, but require user intervention to select the regions. Moreover, no Region Growing method is an optimization method, that is, they are not guaranteed to give optimum solutions. Another technique described in Shi and Malik [6] uses a graph technique, which approximates the solution (i.e., it is not guar-

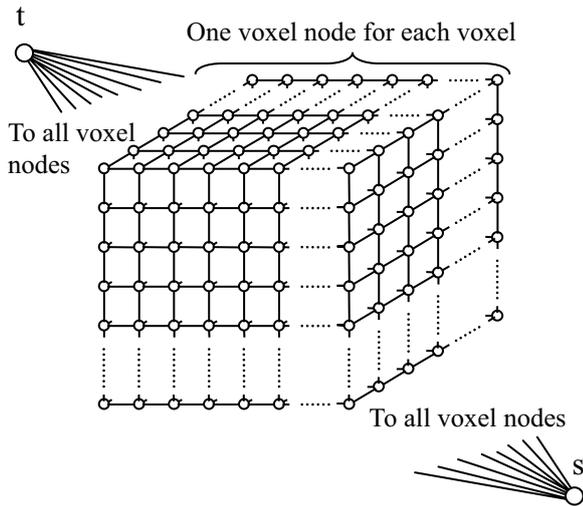


Figure 1. The graph that is used in the method. It has two special vertices s and t , and one vertex for each input voxel. The edges of the graph connect each voxel vertices to s and t , as well as among the voxel vertices, according to the neighborhood structure of the image.

anted), and perhaps does not have the same topological properties. The present method uses similar technique used in other area, 2D image restoration, described in Greig, et. al [5].

3 The Method

In this paper, we describe a method to automatically segment 3D and higher-dimensional images into two subsets without user intervention, with no topological restriction on the solution, and in such a way that the solution is optimal in a precisely defined optimization criterion, including an exactly defined degree of smoothness. The method uses a weighted graph (Figure 1.) The vertices include two special vertices, s , t , and vertices that correspond to input voxels. The edges of the graph connect each voxel vertices to s and t , as well as among the voxel vertices themselves, according to the neighborhood structure of the image. The edges have nonnegative weights. A minimum-cut algorithm is used to cut the graph in two. That is, the set V of vertices are partitioned into two disjoint subsets, one including s and the other t . Then, each of the input voxels are segmented according to the membership of the corresponding vertex to the graph partitions.

Hereafter, the dimension of the input data is denoted by d . In the segmentation problem, a d -dimensional data structure is given as the input to the method. The data structure will be called an “image”, and has voxels and neighborhood structure:

1. Each voxel has associated data, such as a number or a vector. Voxels are conceptually laid out in a d -dimensional configuration. For instance, a 3D ($d = 3$) image can be a simple box of size $L \times N \times M$, or it can also be a subset of such a d -dimensional box.
2. The neighborhood structure is defined by specifying a small set of “neighbor voxels” for each voxel, according to the application. In other words, it is specified,

among all voxels, which voxel is neighboring which other voxels. The neighborhood structure is symmetric in the sense that if a voxel v is a neighbor of another voxel u , then u is also a neighbor of v . The simplest set of “first nearest neighbors” for a voxel includes $2d$ nearest voxels given by increasing or decreasing one of d coordinate entries by 1. The “second nearest neighbors” are those obtained by changing two of the coordinate entries by 1. Similarly, a k -th nearest neighbor of a voxel v has k coordinate entries that are different by 1.

The method partitions the voxels into two complementary subsets S and T , or, equivalently, assigns one of two labels s or t to each voxel. The image will be segmented in the sense that voxels in S , to which label s is assigned, will represent the “interesting” voxels for each particular application, such as voxels corresponding to arteries. It is an advantage of the present method that there is no topological restriction on the resultant subsets. Moreover, our method is completely automatic with no need for user intervention, although it allows the user to intervene as desired in the process to improve or correct the results of the fully automatic system.

The criterion as to how the image should be segmented is given by defining a set of numbers:

- (a) For each voxel v , a number $a(v)$.
- (b) For each neighboring pair v and u , a nonnegative number $b(v, u)$. It is 0 for non-neighboring pairs. Note that $b(v, u)$ and $b(u, v)$ can be different.

Then, the criterion is that the partition shall be given so that the sum

$$\sum_{v \in T} a(v) + \sum_{v \in S, u \in T} b(v, u) \quad (1)$$

is minimum over all possible assignments. The number $a(v)$ represents the likelihood of v to belong to S . If $a(v)$ is positive, v is more likely to belong to S in an assignment with a minimum sum (1). If it is negative, it is more likely to be in T in an assignment with a minimum sum (1). This is usually given by the local evidence from the data that the voxel belongs to S . The number $b(v, u)$ expresses the likelihood of the boundary coming between v and u in such a way that v is in S and u is in T . It shall be larger if such likelihood is smaller. This can reflect, for instance, the gradient of the image at the point, so that the high-gradient points are more likely to be on the boundary, or it can be constant just to realize general cohesiveness. As an example of how these numbers may be selected, suppose that the probabilities

- $P(v)$, of the voxel v belonging to S ; and
- $P(v, u)$, for neighboring voxels v and u , of v belonging to S and u belonging to T ;

are known. Then one possible way is to set

$$a(v) = A \log(2P(v)),$$

$$b(v, u) = -B \log(P(v, u)),$$

where A and B are some positive numbers. In this case we can see this is a first order Markov Random Field optimization problem.

The main idea of the method is to map the voxels to specially interconnected vertices in a graph. A directed graph

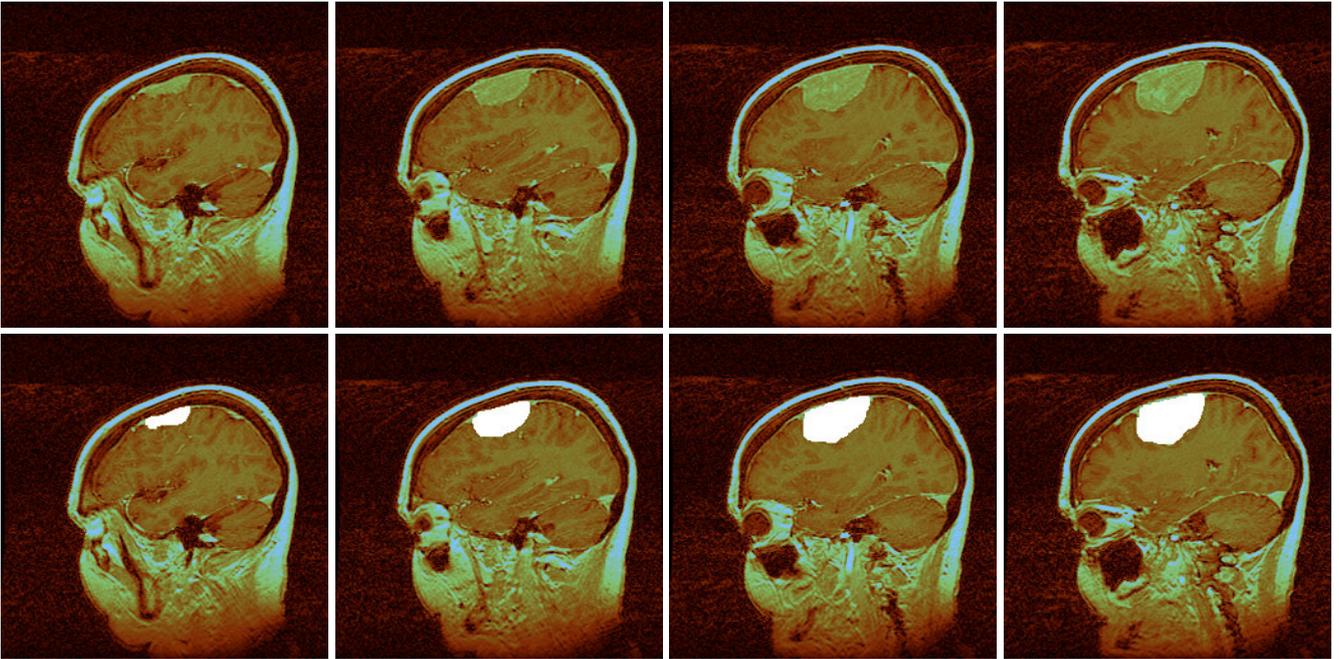


Figure 2. Top: Sample slices of an MRI data. Bottom: The result of segmenting the data; the white part is the segmented tumor.

with edge weights is created. An edge from a vertex v to another vertex u is denoted hereafter by an ordered pair (v, u) . The graph contains the following:

- (a) There is one vertex for each voxel. This type of vertex is hereafter called the voxel vertex corresponding to the voxel, and the voxel vertex corresponding to voxel v is denoted by the same letter v .
- (b) There also are two special vertices s and t that correspond to the two labels s and t , respectively.
- (c) There are edges between voxel vertices. They represent the neighborhood structure between voxels, i.e., voxel vertices corresponding to neighboring voxels are connected by an edge.
- (d) For every voxel vertex v , there is an edge (s, v) from s to v and an edge (v, t) from v to t .

Then, nonnegative edge weights are assigned. For each voxel vertex v , the edge (s, v) is given a nonnegative weight $w(s, v)$ and the edge (v, t) a nonnegative weight $w(v, t)$. These weights are selected so that the following holds:

$$w(s, v) - w(v, t) = a(v).$$

Each voxel vertex v is also connected to its neighbors. For each neighbor u of v , there are edges (v, u) and (u, v) . The edge (v, u) is assigned a weight $w(v, u) = b(v, u)$ and the edge (u, v) is assigned a weight $w(u, v) = b(u, v)$. These weights are chosen so that the segmentation criterion exactly corresponds to a condition on a cut of the graph. Here, a cut is a partition of the graph into two parts, one including s and the other t . Then, each voxel vertex belongs to one of the parts, either including s or t . This defines a segmentation of the image: a vertex that belongs to the same partition as vertex s is assigned the label s , and a vertex that belongs to the same partition as vertex t is assigned the label t . If

an edge goes out from the part including s to the one including t , the edge is said to be “in the cut.” This gives the method an ability to take neighbor interaction into account. There is one-to-one correspondence between partitions of vertices and voxels. A score of the assignment (segmentation) is given by the sum of the edge weights that are in the cut. The segmentation problem is thus mapped to a problem of finding the “minimum cut”, that is, a cut with the minimum score. Thus, if a minimum-cut algorithm is applied to the graph, the minimum cut found corresponds to the optimal segmentation that minimizes the sum (1), because of the way that the edge weights are defined. Any variant of minimum-cut algorithms would suffice, which are known to solve this problem in polynomial time in terms of the number of vertices and edges in the graph. The method possesses all topological properties as described/required above and can be applied to graphs embedded in any dimension, not only 3D.

Finally, voxels are segmented according to the cut of the graph. If a voxel vertex belongs to the same partition as s , the voxel to which it corresponds is assigned the label s and belongs to S . Otherwise, it is assigned the label t and belongs to T . Thus, the method partitions the voxels into two complementary subsets S and T , or, equivalently, assigns one of two labels s or t to each voxel, according to the criterion stated above.

4 Experiments

We applied the method to 3D data. Figure 2 shows the result of segmenting an MRI data on the top row. The data is 124 slices of 256×256 array of 10-bit numbers (total of 8,126,464 voxels,) each voxel representing a physical dimensions of $0.9375\text{mm} \times 0.9375\text{mm} \times 1.5\text{mm}$ slice thickness. The target area was the approximate brain part of the image. A tumor (meningioma in left frontal lobe) is segmented and shown as the white region in the images on the bottom row.

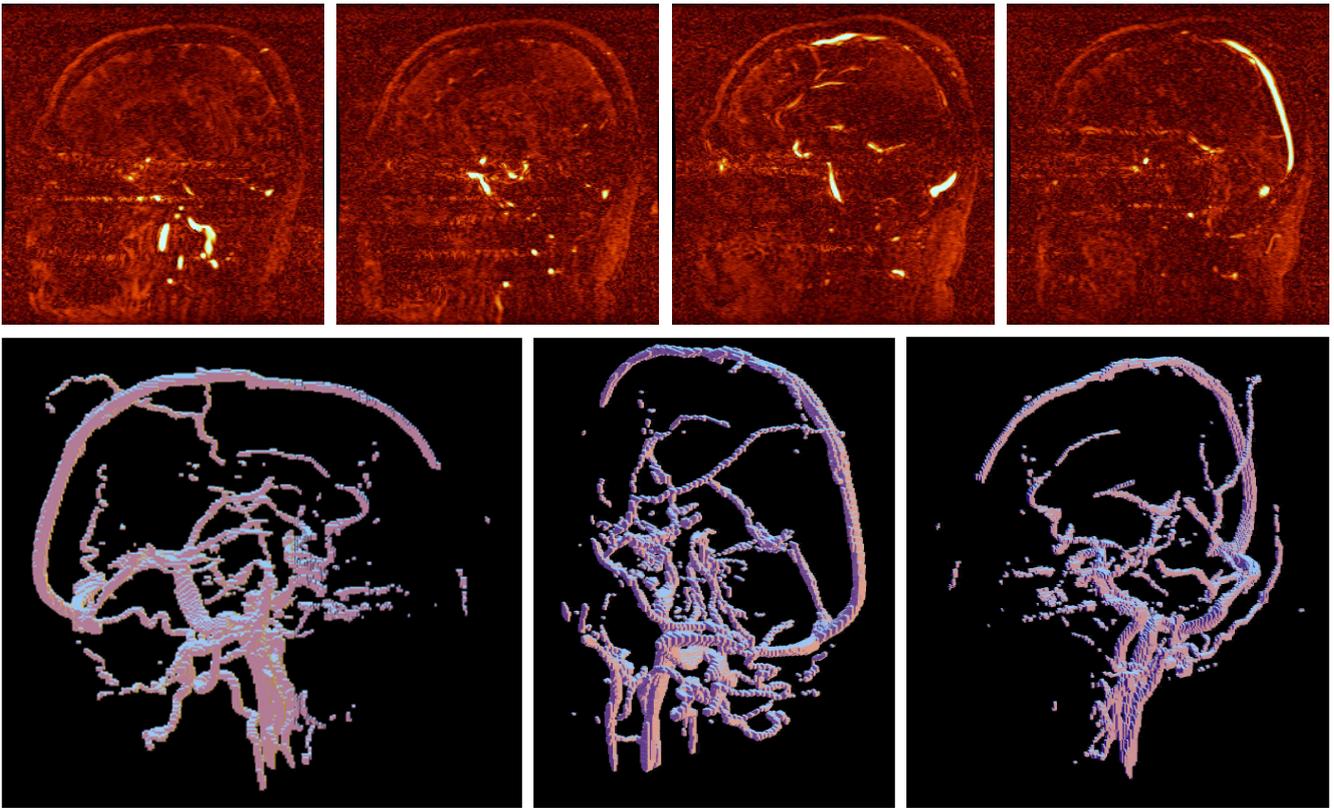


Figure 3. Top: Sample slices of an angiogram data. Bottom: The result of segmenting the data, rendered as seen from three points of view.

We also applied the method to a cerebral angiogram data. The data is 60 slices (half a head) of 256×256 array of 10-bit numbers (total of 3,932,160 voxels.) The result is shown in Figure 3 in three angles. The full picture of network of blood vessels can be seen. As the advantage of this method is that it is not constrained topologically, disconnected components or bifurcating vessels are no problem.

In the experiments, first nearest neighbors are used. The coefficient $a(v)$ at voxel v was given by $a(v) = c \cdot (I(v) - r)^2 - q$, where $I(v)$ is the 10-bit data value at v , and c, q, r are positive constants; for the edge coefficients $b(v, u)$, a constant was used. The method is not too sensitive to small differences in parameters c, q , and r : these parameters can be easily determined to give a good segmentation in a few trials. For minimum-cut algorithm, we used the push-relabel method for the maximum-flow problem (which is well-known to be equivalent to the minimum-cut problem) that is described in [2]. It took 3 to 7 minutes to compute these particular examples on a 3.2GHz machine, utilizing up to 500MB of memory.

5 Conclusions

We have described a method for segmenting 3D and higher dimensional images into two segments, without any topological restriction on the resultant segments. The method is based on a graph algorithm, and it achieves a solution that is optimal under a precisely defined optimiza-

tion criterion. We have also demonstrated the utility of the method by experiments.

References

- [1] Y. Boykov and M-P. Jolly. "Interactive Organ Segmentation using Graph Cuts." In *Proceedings of "MICCAI"*, LNCS 1935, pp. 276–286, 2000.
- [2] B. V. Cherkassky and A. V. Goldberg. "On Implementing Push-Relabel Method for the Maximum Flow Problem." In *Proc. 4th Int. Programming and Combinatorial Optimization Conf.*, pages 157–171, 1995.
- [3] O. Faugeras and R. Keriven. "Complete Dense Stereovision Using Level Set Methods." *Proceedings of the 5th European Conference on Computer Vision*, Springer-Verlag, LNCS 1406, pp. 379–393, 1998.
- [4] C. Fox and G. K. Nicholls. "Exact MAP States and Perfect Expectations: Greig, Porteous and Seheult revisited." In *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 20th International Workshop*. American Institute of Physics, 2001., pp.252–263.
- [5] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact maximum a posteriori estimation for binary images." *Journal of the Royal Statistical Society, Series B*, **51**:271–279, 1989.
- [6] J. Shi and J. Malik. "Normalized cuts and image segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8):888–905, 2000.