Descending Variance Graph for Segmenting Neurological Structures

Shailja[1] and George Stetten[2]

***Abstract*— We present a novel and relatively simple method for clustering pixels into homogeneous patches using an undirected graph of edges between neighboring pixels. The initial steps are same as that of segmenting images using descending variance directed graph method for automatic segmentation [1]. For a 2D image, the mean and variance of image intensity are computed within a circular region centered at each pixel. Each pixel stores its circle's mean and variance and forms the node in a graph. With possible edges to its four immediate neighbors (six in a 3D image), undirected edges are formed connecting all the neighbors with the least variance among all. Such connected islands represent the uniform regions. The method works in n-dimensions and requires no parameter. Setting the intensity of all pixels within a given patch to intensity of pixel with minimum variance in the connected region significantly reduces image noise while preserving the anatomical structure, including the location of boundaries. We demonstrate such segmentation in the high-resolution 2D ultrasound images as well as 3D brain magnetic resonance images (MRIs). Also, we have compared the results and statistics of previous and new version of algorithm.**

Keywords-component: image analysis, segmentation, graph theory, noise filter, ultrasound, MRI

I. INTRODUCTION

The main motive of this paper is to automate the segmentation process which can be used to denoise an image. We have discussed one method in "Descending Variance Graph for Segmenting Neurological Structures" [1]. The algorithm presented here is similar to DVG (undirected) except for the graph structure which results in clustering. In this paper, we have also demonstrated the results and compared it with the previous version of algorithm. The previous version is mentioned as "DVG (undirected)" and the newer version is mentioned as "DVG (directed)" during the comparison. The algorithm is non-parameterized. We use Shells and Spheres (SaS) framework [3] to enable a statistical approach to segmentation that scales the sample size for intensity to the particular object being segmented. This is analogous to what Attali describes as $balls$ for extracting skeletons from predetermined shapes [2]. By a sphere, we mean the entire volume in the 3D and not just a spherical surface. In the SaS framework, we define an integer r as the radius in units of inter-pixel distance, assuming an isotropic image grid. A shell of radius $r$centered at pixel location r contains any pixel whose distance from x rounds to r. A sphere $\mathcal{S}(x, r)$ of radius r is the union of all shells with radii less than or equal to r. These definitions extend to n-dimensional shells and spheres. Spheres centered at every voxel may grow or shrink by adding or deleting an outer shell, performing incremental, and thus efficient, computation of mean and variance of pixel intensity of the pixels within a sphere. For spheres that extend beyond the boundaries of the image, only pixels within the image are used to compute mean and variance. Thus, no assumption is made about the value of pixels outside the image.

We have developed the previous version of the algorithm using the SaS framework and applied it to identify fascicles in the median nerve. Similar to it, we present here a simple and faster algorithm based on the same framework, which uses unit radius for all spheres and creates a graph structure based on comparisons of variances between spheres centered on neighboring pixels.

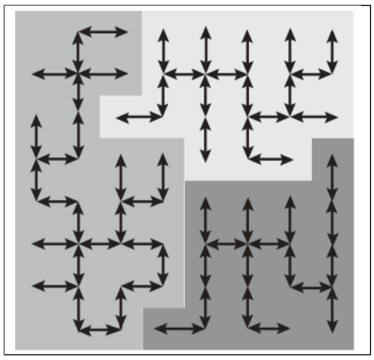


Fig. 1. Undirected (or bi-directed) edges (arrows) between pixels (circles) represent connected pixels. Three disjoint islands form patches (different level of gray).

II. DEFINITION OF ALGORITHM

A. *Creating an undirected graph*

Consider a circle $\mathcal{C}(x, 1)$ with unit radius and centered around every pixel x in the given image. For any pixel, we can compute the mean and variance of the intensity of $\mathcal{C}(x, 1)$ and assign the values to the corresponding pixel x. It is easy to see that the circle around the neighboring pixels will overlap with the circle of radius 1 around the pixel x. This overlap would be considerable if we would instead have circles of radius $r > 1$ around each pixel, $\mathcal{C}(x, r)$.

We can compare the variance assigned to each pixel x with those of the neighboring pixels (4 pixels are connected in 2D and 6 in 3D). Edges are drawn to all the neighboring pixels which has the minimum variance value among the four of its neighbors. Since, each pixel connects to at least one of its

*This work was supported by Bioengineering Department, University of Pittsburgh

[1]Shailja, Visiting Research Scholar, Bioengineering Department, University of Pittsburgh [2]George Stetten, M.D., Ph.D., Professor, Bioengineering Department, U. Pitt, Research Professor, Robotics Institute, CMU

neighbors, we can conclude that there will be no one-pixel size patch. We can use this to reduce the noise in the image.

B. *Patch Formation*

Once the graph is constructed, we can follow either Depth First Search (DFS) or Breadth First Search (BFS) to connect each pixel to all the nodes of its components. The problem statement diminishes to finding the connected component in the forest of disjointed graphs. Depth-first search, or DFS, is a way to traverse such a graph. Once the connected regions are found, we assign the final intensity of the patch as the intensity of pixel with minimum variance in the connected region (if there are multiple such pixels, we can choose any one of them).

The situation is depicted in Fig. 1, which shows the pixels in an 8x8 two-dimensional image as small circles (either filled or empty). Edges are present between neighbors, such that each pixel is connected to one or more of its neighbors. Using the terminology of graph theory [4], we have created a graph made up of nodes (pixels) connected by edges (the lines between neighboring pixels). Our graph, which we call the Descending Variance Graph (DVG), is a simple undirected graph. It is simple because there are no loops (edges between a node and itself). Entire image will be split into disjointed graphs. Since the graphs are disjoint (their nodes do not overlap), the DVG for the whole image is, itself, called a forest. We will apply Depth-First Search to find the connected components. Looking back at Fig. 1, we can see that this particular forest contains three disjointed graphs, each representing a region or ”patch” depicted by a different level of grayscale intensity.

The resulting components represent relatively homogeneous regions. Each patch is a local minimum of variance, whose sphere is generally within its own relatively homogeneous patch, surrounded by pixels with higher variances that drain into it. Each patch is separated from its neighboring patches by a ridge in variance, whose spheres overlap boundaries between one relatively homogeneous region and another. The system is self-normalizing and parameter-free.

We next present some preliminary results showing the comparison between the previous and new version of the algorithm.

III. Overview of DVG (directed) algorithm

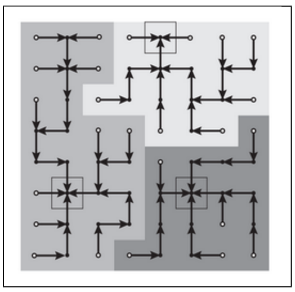


Fig. 2. Directed edges (arrows) between pixels (circles) represent connected pixels. Three disjoint islands form patches (different level of gray).

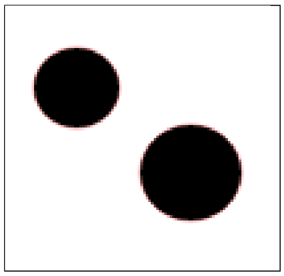
Method:

- Given the integer radius r of spheres $S(x, r)$ centered at every pixel in the image, mean and variance are computed for intensity within each sphere.
- Each pixel stores its sphere's mean and variance, and forms the node in a graph, with possible edges to its 4 immediate neighbors.
- If at least one of those neighbors has a lower variance than itself, a directed edge is formed, pointing to the neighbor with the lowest variance.
- The nodes and edges form a simple directed acyclic graph, which we call Descending Variance Graph (DVG).
- Local minima in variance thus form the roots of disjoint trees, representing patches of relative homogeneity.
- Each patch is separated from neighboring patches by a ridge of variance.

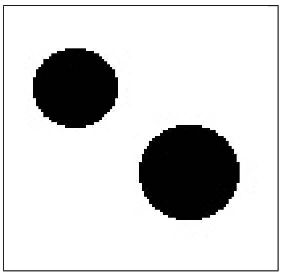
IV. Results

Here, we are going to show results from three different types of images using DVG (directed) and DVG (undirected)

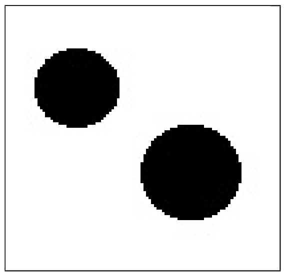
A. *Using 2D Homogeneous Binary Images*



(a) Input Image



(a) Output Image from DVG (directed)



(b) Output Image from DVG(undirected)

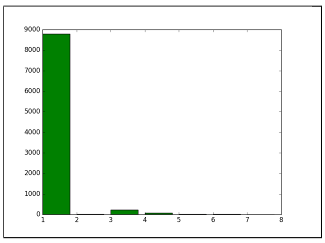


Fig. 3. Histogram of patch sizes using DVG (directed) algorithm

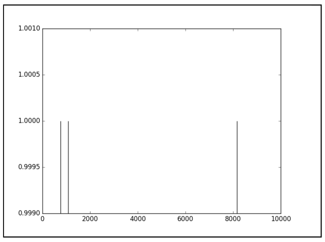


Fig. 4. Histogram of patch sizes using DVG (undirected) algorithm

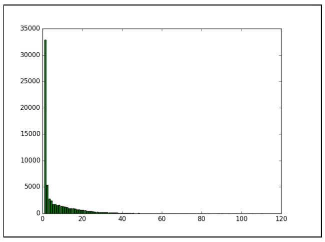


Fig. 5. Histogram of patch sizes using DVG (directed) algorithm

TABLE I
COMPARISON OF PARAMETERS FROM DVG (DIRECTED) ALGORITHM AND DVG (UNDIRECTED) ALGORITHM

Parameters	DVG (directed)	DVG (undirected)
Total patches created	9139	3
Max patch size	7	8158
Min patch size	1	758
Mode patch size	1	758
No. of patches of min patch size	8795	1

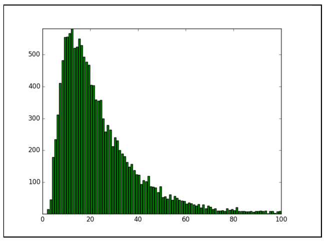
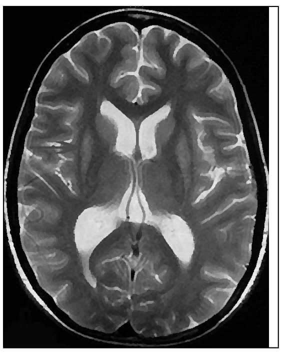


Fig. 6. Histogram of patch sizes using DVG (undirected) algorithm

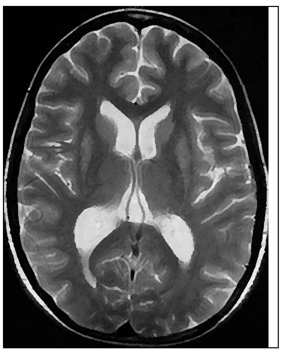
We can clearly see from the above data that the algorithm was successfully able to segment the homogeneous regions clustering them to three different patches.

B. Using 2D brain MRI

(a) Input Image



(a) Output Image from DVG (directed)



(b) Output Image from DVG (undirected)

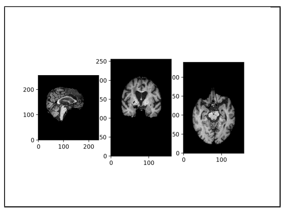
TABLE II
COMPARISON OF PARAMETERS FROM DVG (DIRECTED) ALGORITHM AND DVG (UNDIRECTED) ALGORITHM

Parameters	DVG (directed)	DVG (undirected)
Total patches created	8644955	19761
Max patch size	397	8741843
Min patch size	1	2
Mode patch size	1	17
No. of patches of min patch size	8560497	20

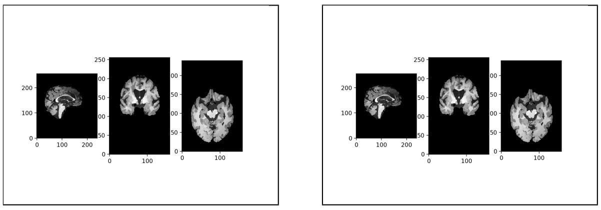
Observations:

- Total number of patches is reduced by around 78%
- Maximum patch size is increased by 200 times
- Minimum patch size is increased to 2 with only 15 such patches as compared to previous results where minimum patch size is 1 with 32861 such patches.
- Mode patch size was 1 in the previous version which is increased to 12.

C. Using 3D Brain MRIs



(a) Input Image



(a) Output Image from DVG (directed)

(b) Output Image from DVG (undirected)

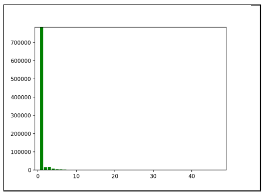


Fig. 7. Histogram of patch sizes using DVG (directed) algorithm

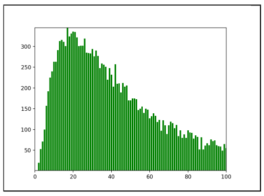


Fig. 8. Histogram of patch sizes using DVG (undirected) algorithm

TABLE III

COMPARISON OF PARAMETERS FROM DVG (DIRECTED) ALGORITHM AND DVG (UNDIRECTED) ALGORITHM

Parameters	DVG (directed)	DVG (undirected)
Total patches created	67289	14750
Max patch size	110	21270
Min patch size	1	2
Mode patch size	1	12
No. of patches of min patch size	32861	15

V. DISCUSSION

The aim of our method is to improve the previous version of the algorithm, thereby representing homogeneous regions in form of larger patches. This can be considered as a second step of clustering the patches formed by DVG (undirected) algorithm. From the above results, we have demonstrated that the method is successful in segmenting binary images as well as. Also, the algorithm works absolutely in both 2D and 3D. Our method has some similarities to other graph-based methods for image segmentation. These methods generally treat the entire image as a graph, with each pixel as a node connected to its neighbors by edges (either a 4-connected or 8-connected neighborhood).

VI. CONCLUSION

The contribution of our work, we believe, is to provide a simple and rapid method to reduce the noise while preserving edges in n-dimensional images. The algorithm is nonparametric. In comparison to the previous version of the algorithm, we get larger patches. Patches represent a compression of the useful information in the image.

The subsequent step for our research is expected to be utilizing patches for shape analysis. Examining lines and circles in 2D could be the most useful first step. This can be extended to classifying cylinders and slabs in 3D. We are working towards finding medial-ness in anatomical structures using the segmented patches. Since the number of patches formed is much less than the total number of pixels. Patch-based computation would have lower time complexity than pixel-based computation. We also propose to apply the DVG algorithm to 3D images for skull stripping.

ACKNOWLEDGMENT

The algorithm was experimented using 7T MRI data provided by University of Pittsburgh Medical College. We would like to thank Professor Howard J. Aizenstein and Professor Minjie Wu for their support.

REFERENCES

[1] George Stetten, Cindy Wong, Vikas Shivaprabhu, Ada Zhang, Samantha Horvath,Jihang Wang, John Galeotti, Vijay Gorantla, Howard Aizenstein, University of Pittsburgh, Departments of Bioengineering, Psychiatry, and Surgery, Carnegie Mellon University, Robotics Institute.:Descending Variance Graphs for Segmenting Neurological Structures

[2] Attali, D.: Computing and Simplifying 2D and 3D Continuous Skeletons, Computer Vision and Image Understanding, Vol. 67, No. 3, pp. 261–273. (1997)

[3] Cois, A., Rockot, K., Galeotti, J., Tamburo, R., Gottlieb, D., Mayer, J., Powell, A., Sacks, M., Stetten, G.: Automated segmentation of the right heart using an optimized shells and spheres algorithm, International Society of Biomedical Imaging, Arlington VA. (2007)

[4] West, D.B.: Introduction to Graph Theory, 2 nd ed. Prentice Hall, Upper Saddle River, NJ. (2001)