ISSN: 2321-7782 (Online)
Volume 2, Issue 6, June 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Image Segmentation by Graph Cut Method

Roopa Hubballi
Dept. of Computer Science
K.L.E Dr. M.S. Sheshgiri College of Engineering and Technology
Belgaum – India

Abstract: Image segmentation is a difficult problem. Retrieving segments in the image using image content as a key is a challenging and important problem. Graph cut is a graph partitioning method it proposes a novel global criterion for segmenting the graph. Segmentation subdivides an image into its constituent regions or objects. Segmentation is based on one of 2 basic properties of image intensity values discontinuity and similarity. The first approach is considered where we partition an image based on abrupt changes in intensity. Graph cut property is used to segment the image in which each pixel in an image is considered to be node of the graph and edge weight of the graph is calculated from neighboring pixels or nodes. Min cut or Max flow algorithm is a binary segmentation algorithm and is repeatedly called by Graph cut algorithm to get n segments in the image.

Keywords: Energy minimization, Graph algorithms, Image segmentation, Maximum flow, Minimum cut.

I. INTRODUCTION

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments sets of pixels, also known as super pixels. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic. Graphs can effectively be used for image segmentation. Usually a pixel or a group of pixels are vertices and edges define the dissimilarity among the neighbourhood pixels. Graph cut is one among the segmentation algorithms. Maxflow/Mincut is one of the techniques used to get cut on the graph.

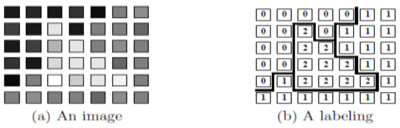

Fig 1(a) and 1(b) Unlabelled image and Labelled image

Consider **Fig 1(b)** where in labeling of pixels is done. Labeling L shown in (b) assigns some label Lp {0,1,2} to each pixel p. Such labels can represent depth (in stereo), object index (in segmentation), original intensity (in image restoration), or other pixel properties. Normally, graph-based methods assume that a set of feasible labels at each pixel is finite. Thick lines in (b) show labeling discontinuities between neighboring pixels. The expansion algorithm finds provably good approximate solutions by iteratively running min-cut/max-flow algorithms on appropriate graphs.

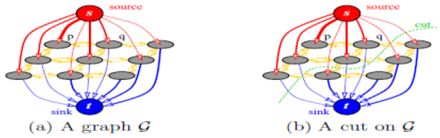

Fig (2)A Directed graph with two vertices without and with cut

A directed graph shown below in **Fig (2)a** with positive edge weights and two special vertices: A source(s) with only outgoing edges and a sink(t) with only incoming edges. On this graph a cut is a binary partition of the vertices into a set S around the source and a set T around the sink. The cost of the cut is the sum of the weights of all the edges inducing flow from source to sink. Cut edges that induce flow in the opposite direction. Finding the cut with minimal cost is solvable in polynomial time. Binary labeling is equivalent to partitioning, so construct a directed graph. All edges in the graph are assigned some weight or cost. A cost of a directed edge (p, q) may differ from the cost of the reverse edge (q, p). In fact, ability to assign different edge weights for (p, q) and (q, p) is important for many graph-based applications in vision. Normally, there are two types of edges in the graph: n-links and t-links. N-links connect pairs of neighboring pixels or voxels. Thus, they represent a neighborhood system in the image. Cost of n-links corresponds to a penalty for discontinuity between the pixels. T-links connect pixels with terminals (labels). The cost of a t-link connecting a pixel and a terminal corresponds to a penalty for assigning the corresponding label to the pixel.

II. MAXFLOW ALGORITHM

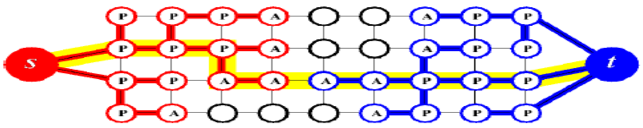

Fig (3) Augmented path from source(s) to sink(t)

The max-flow algorithm presented here belongs to the group of algorithms .Fig 3: Example of the search trees S (red nodes) and T (blue nodes) at the end of the growth stage when a path (yellow line) from the source s to the sink t is found. Active and passive nodes are labeled by letters A and P, correspondingly. Free nodes appear in black on augmenting paths. It builds search trees for detecting augmenting paths. Build two search trees, one from the source and the other from the sink. The other difference is that reuse of these trees is done and never start building them from scratch.

The drawback of this approach is that the augmenting paths found are not necessarily shortest augmenting path; thus the time complexity of the shortest augmenting path is no longer valid. The trivial upper bound on the number of augmentations for our algorithm is the cost of the minimum cut |C|, which results in the worst case complexity O(mn2|C|).

$$S \subset \mathcal{V}, \quad s \in S, \quad T \subset \mathcal{V}, \quad t \in T, \quad S \cap T = \emptyset. \quad ....(1)$$

Fig 3 illustrates the basic terminology. Two non-overlapping search trees ‘S’ and ‘T’ with roots at the source s and the sink‘t’, correspondingly. In tree S all edges from each parent node to its children are non-saturated, while in tree T edges from children to their parents are non-saturated. The nodes that are not in S or T are called “free”. The nodes in the search trees S and T can be either “active” or “passive”. The active nodes represent the outer border in each tree while the passive nodes are internal. The point is that active nodes allow trees to “grow” by acquiring new children (along non-saturated edges) from a set of free nodes. The passive nodes cannot grow as they are completely blocked by other nodes from the same tree. It is also important that active nodes may come in contact with the nodes from the other tree. An augmenting path is found as soon as an active node in one of the trees detects a neighboring node that belongs to the other tree.

The algorithm iteratively repeats the following three stages:

• “growth” stage: search trees S and T grow until they touch giving an s −> t path

• “augmentation” stage: the found path is augmented, search tree(s) break into forest(s)

• “adoption” stage: trees S and T are restored.

At the growth stage the search trees expand. The active nodes explore adjacent non-saturated edges and acquire new children from a set of free nodes. The newly acquired nodes become active members of the corresponding search trees. As soon as all neighbors of a given active node are explored the active node becomes passive. The growth stage terminates if an active node encounters a neighboring node that belongs to the opposite tree. In this case a path is detected from source to sink as shown in Fig 3.The augmentation stage augments the path found at the growth stage. Since the push is through the largest flow possible some edge(s) in the path become saturated. Thus, some of the nodes in the trees S and T may become “orphans”, that is, the edges linking them to their parents are no longer valid (they are saturated). In fact, the augmentation phase may split the search trees S and T into forests. The source s and the sink t are still roots of two of the trees while orphans form roots of all other trees.

The goal of the adoption stage is to restore single-tree structure of sets S and T with roots in the source and the sink. At this stage a new valid parent for each orphan is found. A new parent should belong to the same set, S or T, as the orphan. A parent should also be connected through a non-saturated edge. If there is no qualifying parent remove the orphan from S or T and make it a free node. All its former children are declared as orphans. The stage terminates when no orphans are left and, thus, the search tree structures of S and T are restored. Since some orphan nodes in S and T may become free the adoption stage results in contraction of these sets.

After the adoption stage is completed the algorithm returns to the growth stage. The algorithm terminates when the search trees S and T cannot grow (no active nodes) and the trees are separated by saturated edges. This implies that a maximum flow is achieved.

III. EXPERIMENTAL RESULTS

Table 1 shows the execution time for different methods for image segmentation. The graphical analysis of the execution times can be viewed in Fig(4). Fig(5) shows normal gray scale and colored images and their segmented results.

Sl.no	Algorithm	Minimum Execution time in seconds(s)
1	Graph cut	0.5625
2	Contour	3
3	K- means	20
4	PatternMatching	120
5	Region growing	11

Table 1 Comparison of execution time in seconds of different algorithms

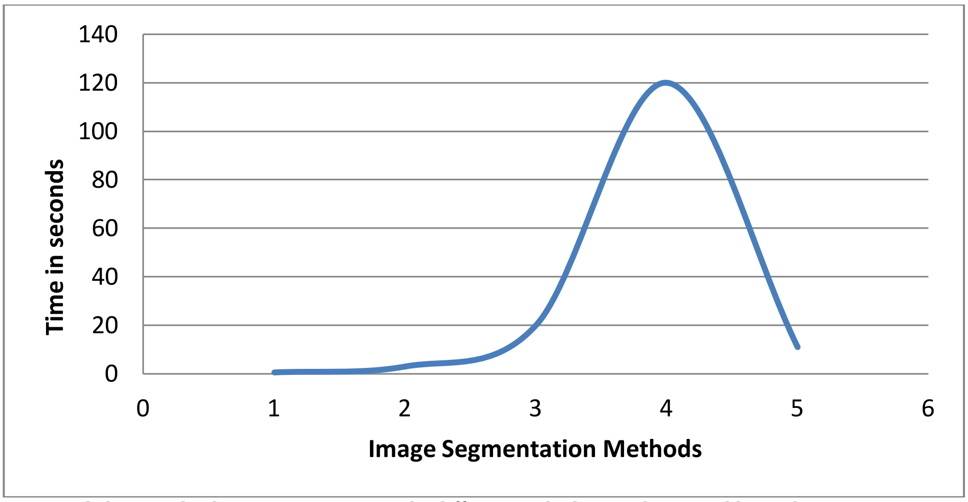

Fig. 4 A sample line graph where x-axis represents the different methods according to Table1 and y –axis time in seconds

Input Image **Segmented image**

Fig(4) A normal gray scale and colored images and their segmented results.

IV. Conclusion

Graph cut algorithm works several times i.e 2 - 5 faster than other methods making real time performance possible. The powerful mincut or maxflow algorithm from combinatorial optimization can be used to minimize certain important energy functions in computer vision.

Acknowledgement

The author is immensely grateful to the valuable guidance provided by Dr.Nandini Sidnal, HOD, Department of Computer Science& Engg, KLE Dr.M.S.S CET, Udyambag, Belgaum.

References

1. Yuri Boykov, Olga Veksler, and Ramin Zabih." Fast approximate energy minimization via graph cuts". IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239, November 2001.
2. Yuri Boykov, Vladimir Kolmogorov: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Mimization in Vision. IEEE Trans. Pattern Anal.Mach.Intel26(9):1124-1137 (2004)
3. Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In International Conference on Computer Vision, volume I, pages 26–33, 2003.
4. JianboShi and Jitendra Malik, "Normalized Cuts and Image Segmentation", IEEE Transactions on PAMI, Vol. 22, No. 8, Aug 2000S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
5. P. Felzenszwalb and D. Huttenlocher. Image segmentation using local variation. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 98-104, 1998.

AUTHOR(S) PROFILE

Roopa Hubballi received the M.Tech degree in Computer Science from Gogte Institute of Technology in 2009. She is lecturer in Computer Science Dept. of K.L.E Dr. M.S.S.C.E.T Belgaum since 2011.Her area of interest is image segmentation and pattern recognition.

