

# A Review On Existing Interactive Graph Cut Image Segmentation Technique

Priyanka Panchhi  
M.Tech 2<sup>nd</sup> year, GIMT Kanipla

Anurag Jain  
Asstt. Prof. , GIMT Kanipla

## Abstract

An image can be considered as a group of nodes as vertices and edges as links. Various techniques are formed based upon this assumption and energy minimization. Graph cut is one of the promising techniques for image segmentation. Boykov and Kolmogorov use mincut/ maxflow graph principal for image segmentation. We have discussed some other techniques such as grab cut which is user interactive and very effective technique. In last the current problems were also highlighted.

## 1. Introduction

The paper addresses the various interactive foreground extraction techniques used in computer vision of research. Foreground image segmentation is the task of identifying an object from the background. Many approaches are their based upon color information, edge information or region connectivity. The problem of image segmentation has received a lot of attention since the early days of computer vision of research. The paper is organized as follows – in section II basic of Graph segmentation method is discussed in detail. The concept of min cut and max flow is given in section III. Grab cut algorithm and detail is given in section IV. And finally the conclusion and future work is given in section V and VI respectively.

## II. Graph cut

The Graph cut image segmentation approach by Boykov and jolly is the foundation of interactive image segmentation. In this section we review some basic about graphs and graph cut image segmentation. A directed weighted (capacitated) graph  $G = (V, E)$  consists of a set of nodes  $V$  and a set of directed edges  $E$  that connect them. The nodes correspond to pixels. A graph also contains some additional special nodes called terminals. In the context of vision, terminals correspond to the set of labels that can be assigned to pixels. The terminals are usually called the source and the sink. 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  $(q, p)$ .

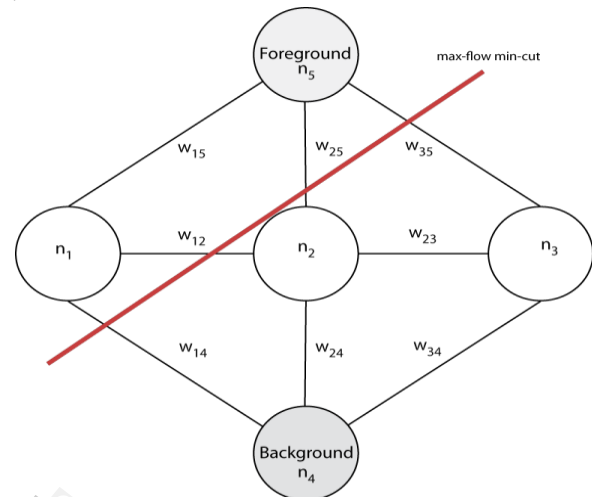


Figure 1: Example of a graph to segment to foreground and background

There are two types of edges in the graph: N-links and T-links. N-links connect pixels in the 8-neighborhood. These links describe the penalty for placing a segmentation boundary between the neighboring pixels. We want this penalty to be very high in regions of low gradient and low in regions of high gradient (edges). The N-link weights are constant throughout the execution of the algorithm. Thus, they can be computed once and reused. T-links connect each pixel to the foreground and background nodes. These describe the probability that each pixel belongs to the foreground or to the background. For N-links the appropriate link weight between pixel  $m$  and  $n$  is:

$$N(m, n) = 50/\text{dist}(m, n) e^{-\beta \|z_m - z_n\|^2} \quad (1)$$

Where  $z_m$  is the color of pixel  $m$ .

$D_{\text{Fore}}$  and  $D_{\text{Back}}$  are the likelihoods that the pixel belongs to the foreground and background GMMs respectively. There are two T-links for each pixel. The Background T-link connects the pixel to the background node. The Foreground T-link connects the pixel to the Foreground node. The weights of these links depend on the state of the trimap.

The T-Links weights for pixel  $m$  are:

Table 1: T-Link weights

Pixel Type	Back. T-link	Fore. T-link
$m \in \text{TrimapForeground}$	0	K
$m \in \text{TrimapBackground}$	K	0
$m \in \text{TrimapUnknown}$	$D_{\text{Fore}}(m)$	$D_{\text{Back}}(m)$

### III. Min cut/ Max Flow problem

An  $s/t$  cut  $C$  on a graph with two terminals is a partitioning of the nodes in the graph into two disjoint subsets  $S$  and  $T$  such that the source  $s$  is in  $S$  and the sink  $t$  is in  $T$ . The minimum cut problem on a graph is to find a cut that has the minimum cost among all cuts. The minimum  $s/t$  cut problem can be solved by finding a maximum flow from the source  $s$  to the sink  $t$  by using Ford –Flukerson Max Flow theorem [8]. The min-cut and max-flow problems are equivalent in fact; the maximum flow value is equal to the cost of the minimum cut. Mincut of a graph is shown in figure 1 for foreground extraction. A new mincut maxflow algorithm was given by Yovi Boykov et.al [3]. The algorithm belongs to group of algorithm based on augmented paths.

#### Algorithm overview

Figure 2 illustrates basic terminology. Boykov et al. maintain 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”. We have

$$S \subset V, \quad s \in S, \quad T \subset V, \quad t \in T, \quad S \cap T = \emptyset.$$

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.

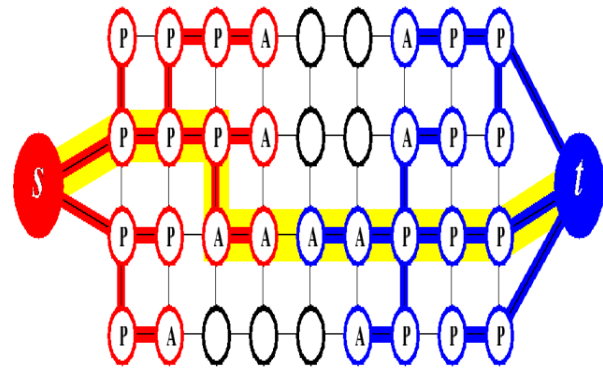


Figure 2: Example of the search trees  $S$  (red nodes) and  $T$  (blue nodes) at the end of the growth stage when a path (yellow line) from the source  $s$  to the sink  $t$  is found. Active and passive nodes are labeled by letters  $A$  and  $P$ , correspondingly. Free nodes appear in black.

The algorithm iteratively repeats the following three stages:

- “growth” stage: search trees  $S$  and  $T$  grow until they touch giving an  $s \rightarrow t$  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 we detect a path from the source to the sink, as shown in Figure 2.

The augmentation stage augments the path found at the growth stage. Since we push 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 we try to find a new valid parent for each orphan. 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 we remove the orphan from  $S$  or  $T$  and make it a free node. We also declare all its former

children 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. The corresponding minimum cut can be determined by  $S = S$  and  $T = T$ .

#### IV. Grab Cut Algorithm

GrabCut is a way to perform 2D segmentation in an image that is very user friendly. The users only need to input the very rough segmentation between foreground and background. Typically this is done by drawing a rectangle around the object of interest also called trimap as shown in figure 3.



Figure 3: Example of Grabcut

Grabcut extends graphcut to color images and to complete trimaps. Grabcut is a very promising image editing tool for foreground extraction. Justin. F. Tablot et. al implemented the algorithm[6] in 2006. The summary of grabcut is given here.

1. User creates an initial trimap by selecting a rectangle. Pixels inside the rectangle are marked as unknown. Pixels outside of rectangle are marked as known background.
2. Computer creates initial image segmentation, where all unknown pixels are tentatively placed in the foreground class and all known background pixels are placed in the background class.

3. Gaussian Mixture Models (GMMs) are created for initial foreground and background classes.
4. Each pixel in the foreground class is assigned to the most likely Gaussian component in the foreground GMM. Similarly, each pixel in the background is assigned to the most likely background Gaussian component.
5. The GMMs are thrown away and new GMMs are learned from the pixel sets created in the previous set.



Figure 4: Result without user touchup

6. A graph is built and Graph Cut is run to find a new tentative foreground and background classification of pixels.
7. Steps 4-6 are repeated until the classification converges.

Figure 4 shows an image where significant user touchup to achieve good segmentation because the part of fish is mistakenly excluded and some coral are included.

#### V. Conclusion

The paper describes the various interactive image segmentation technique used in computer vision with details. All the interactive segmentation techniques are based on graph cut algorithm can also be used for energy optimization of graph using mincut/maxflow

algorithm. Grabcut works well for the image segmentation by selecting a trimap and it is a user friendly image segmentation tool. Grabcut is also faster than other interactive image segmentation techniques. However for some type of images grab cut fails completely.

## VI. Future Work

Graph cut can be used for energy optimization using mincut/maxflow algorithm. Although grab cut is much faster than previous method but delay due to graph cut step can be optimized by using some technique. It can also be extended to various color models and N-D segmentation algorithm.

Number of components for Gaussian mixture model was arbitrary chosen. A better approach can be used for determining number of components based upon color complexity of image to find clean separation between foreground and background of image.

Grab cut would be greatly simplified if the border mating could be computing in some graph cut oriented data structure to achieve good segmentation without user touchups.

## VII References

[1] Mohammad Jahangiri and Daniel Heesh, "Modified Grabcut for Unsupervised Object Segmentation," *IEEE International Conference Image Processing (ICIP)*, pp.2389-2392, 2009.

[2] Daniel Chen, Brenden Chen, George Mamic, Clinton Fookes and Sridha Sridharan, "Improved Grabcut Segmentation via GMM Optimisation," *IEEE International Conference*, 2007

[3] Yuri Boykov and Vladimir Kolmogorov, "An Experimental Comparison of min-cut/max-flow algorithm for energy optimization in vision," *IEEE transaction on PAMI*, Vol. 26, No. 9, pp. 1124-1137, 2004.

[4] Fahim Manan, "Interactive Image Segmentation , " *Center of Intelligence Machines*, 2007.

[5] Carsten Rother, Vladimir Kolmogorov, Andrew Blake "GrabCut" - Interactive Fore-ground Extraction using Iterated Graph Cuts. *Microsoft Research Cambridge*, UK

[6] Justin F. Talbot, Xiaoqian Xu, "Implementing GrabCut," *Brigham Young University*, Revised: April 7, 2006.

[7] Tomos Malmer, "Image Segmentation Using GrabCut." May, 2010.

[8] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.