GrabCut - Iterative Foreground Extraction using Iterated Graph Cuts

Dohoon Kim
dohoon.kim@wustl.edu

Abstract

Image segmentations have been proposed significantly on the problem of optimizing user interactive foreground extraction from background in still images. This project implements an approach "GrabCut" that outperforms other state of the art interactive approaches regarding to accuracy and efficiency on user interaction. This project aims to achieve optimization with iterative Graph Cuts that incorporates min cut algorithm to reduce the load of user interaction. To enhance optimization, tuning the number of components for GMM and user editing of denoting solid foreground and background have been made.

1 Introduction

The problem of image segmentation is critical in separating the object from the background in images. Initially, Graph Cuts method addressed by Boykov and Jolly [1] addresses the solution by best balancing boundary and region properties among all segmentations satisfying the constraints. Hard constraints are built through marking the object pixels and background pixels. The pixels in the image are connected as nodes in a graph model and soft constraints get built on the edges between the nodes. Then, with min-cut estimation, object and background are hard segmented. Connections between nodes or pixels consist of energy associated to cut out. Min-cut algorithm goes along with minimizing the energy and cuts connections based upon that and generate hard segmentations on object and background. Other previous methods on image segmentation adopted either texture(color) information or edge(contrast) information. Using Graph Cuts method that combines color information and contrast information through trimaps and probabilistic color models gives robust segmentation on foreground and background to the hardly distinguishable image of object and background.

GrabCut is a technique derived from GraphCuts that iterative GraphCuts alternatively update the image model parameters and segment based on minimum cut on updated energies. Significant difference in Graph Cuts and GrabCut is that GrabCut implements the Gaussian Mixture Model for model parameters for foreground and background map while Graph Cuts takes account histograms of gray values from image foreground and background distributions. GrabCut renders user to have light load on interaction as user starts with marking the bounding box for the object image. This initially segments the inside and outside of the box as foreground and background, then iterative estimation of parameters and energies, and incomplete labeling of segments are built. For more accurate segmentation, GrabCut implements the border matting of separating object pixels and background pixels precisely and user can mark for foreground and background on the segmented output with incomplete labeling and run min-cut algorithm to get more precise segmentation.

(a) Original flower (b) Segmented flower

Figure 1: Flower image with GrabCut

This project aims to implement general GrabCut algorithm and test accuracies of image segmentation comparing to the ground truth. In Figure 1(a), original flower is displayed and before going into segmentation using iterative Graph Cuts, the bounding box containing the foreground object, flower, is chosen and initially pixels within box are marked to be object pixel and pixels outside of box are marked to be background pixel. Then with 10 iterations of Graph Cuts, the accurate segmentation of flower from the background was accomplished.

2 Related Work

As priorly stated, image segmentation on separating image object from background was implemented with various methods. GrabCut is a development from Graph Cuts implementing min-cut algorithm. There are optimized algorithms of Graph Cuts and related method developed previously. This project will analyze the past various user interactive methods for image segmentation related to Graph Cuts.

One method that is relevant to Graph Cuts is Normalized Cuts [2] which implements the globally optimizing criterion for partitioning a graph for min-cut. The normalized cut criterion is based on both the total dissimilarities between different groups and total similarities in the groups to be set as a global criterion. According to the paper, this criterion can be solved from generalized eigenvalue problem and min-cut is enhanced more efficiently. With partitions of graph, it is stated the minimizing energy for cutting the edge would be decreased and total energy to be decreased with giving accurate segmentation.

Another method that optimizes the Graph Cuts method is using a tighter bounding box as a powerful prior for foreground pixels [3]. The paper addresses that tighter bound of rectangle around the object makes total energy to minimize to be lessened, leading to higher accuracy and efficiency in image segmentation. Sufficient tightness of the bound is defined as desired segmentation should have parts that are sufficiently close to each of the sides of the bounding box.

One of methods that can deal with different input modalities and noise generated from user selection is based on properties of parametrized family of quadratic optimization problem regarding dominant set clusters [4]. Dominant set clusters are defined as maximally coherent dataset that optimizes quadratic problem.

Another work built from Graph Cuts is image segmentation with shape priors [5]. This algorithm renders edges in a graph to hold both the information as the connection between nodes and prior shape information. And, during min-cut process of Graph Cuts, the cutting of edges would be determined by the weight of connection and prior shape so that even though two object pixels may have higher weight of connection thus the edge between them is at the verge of cutting out, higher prior shape information on this may prevent unnecessary termination of edge.

Final work that was toward optimizing the min-cut algorithm from using multiple images of the same object to apply automatic segmentation, introduced by [6]. Multiple images are all layered and two images, pixels across these two images define pixels' energy. Spatial similarity and discriminative cluster information help deciding two pixels, which are in two images, belong to same one. Then the authors use minimum cut based on energy with co-segmenting all the images and generate optimized solution for image segmentation.

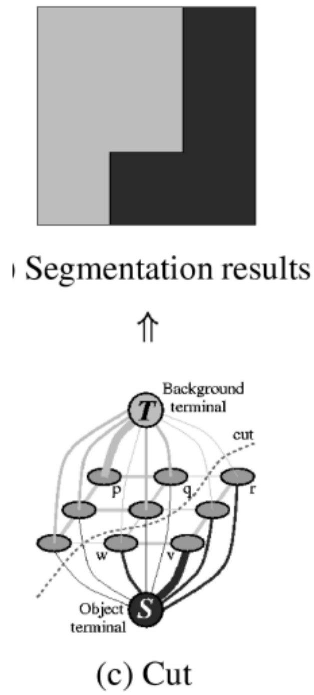


Figure 2: Min-Cut example from [7]

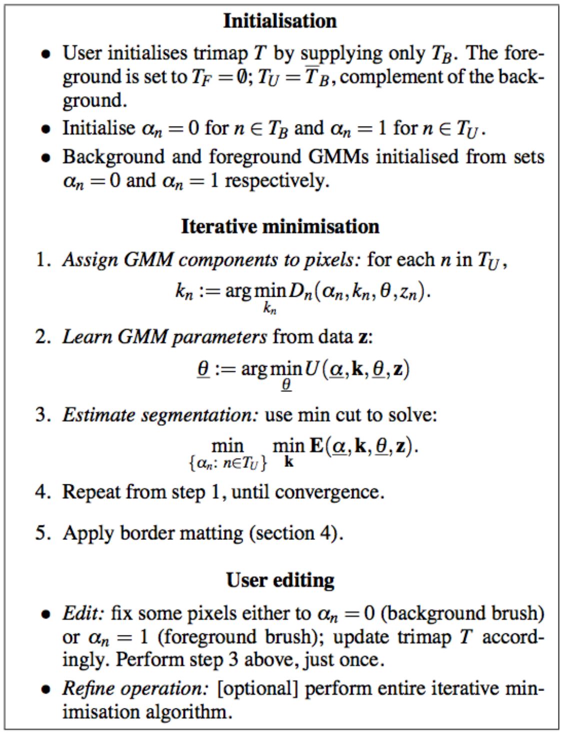
3 GrabCut

3.1 Max-flow and Min-cut theorem

Max-flow and Min-cut theorem is a pivotal element in GrabCut. Typically saying Min-cut algorithm is fundamental of Graph Cuts in that the graph modeled image with nodes as pixels and connections as shown in Figure[2] between them as edges, cut the connections with maximal weights in sequential on connections to partition graph in image segmentation. Details of equation will be in next section.

3.2 Implementation and Algorithm

The algorithm initiates from creating alpha map that is indicator of foreground and background pixels. The



Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n :\, n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

Figure 3: GrabCut algorithm

(a) Segmented image (b) User interaction

Figure 4: GrabCut and user interaction

(a) segment after GrabCut (b) segment after interaction

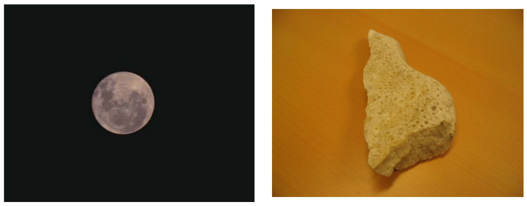
Figure 5: GrabCut after user interaction

trimap T consisting of T_B,T_F,T_U is initiated with the bounding box that user can constrain the foreground pixels as in rectangle shape. Thus, T_B takes all the pixels outside of bounding box as background and T_U and T_F are all initiated with the pixels inside the bounding box. Alpha map mark the pixels that belong to T_B as 0 and the pixels that belong to T_U as 1. Then, Gaussian Mixture Model for foreground and background get initiated from the alpha maps respectively. For initialization of two GMMs, the project implements K-means clustering with k as number of components for future use and stores these clusters for the first iteration. The resulting segmentation is built upon K-means clustering working as GMM to set Gaussian components for each foreground GMM and background GMM.

The next step as seen in Figure[3], parameters for two GMMs are updated using the alpha map or segmentation updated from previous iteration or initialization. For each foreground and background GMM, the parameters, mean and covariance are learned with the pixels segmented as object and background respectively.

After this step, for estimating segmentation using min-cut on the energies on connections between nodes, the image is formulated as a graph model. Edges for connections are weighted every iteration for min-cut process. As we can see from the step 3 in Figure[3], to estimate the segmentation, edges with lowest energies are not cut. First of all, weights information from GMM are initiated as negative log likelihood of each pixel belonging to each GMM. Also the weights from difference between two pixels are initiated. This project tries to use weights in terms of energy minimization for min-cut. Since the weights are negative log likelihood of energy, if weight is low, we can cut the edge based on the weight. Thus, with first weights from GMM, when the pixel is in the bounding box and the decision to assign node to be in foreground is determined by first weight, and using the second weight on the connection between the weights can be combined with first weight to determine whether the pixel can be still marked as foreground or background pixel in alpha matte.

After iterations of Graph Cuts with updating alpha map as segmentation, this project comes up with user interaction to mark mislabeled pixels for foreground and background respectively using brush. As we can see from Figure[4a], background pixels are not totally removed from object pixels around the moon, then user interaction with brushing those around the moon in black and run GrabCut again, then we have accurately segmented moon. The segmentation in binary is well shown in Figure[5].



(a) Original moon (b) Original stone

Figure 6: Original images

(a) GrabCut moon (b) GrabCut stone

Figure 7: GrabCut result

(a) User interaction moon (b) User interaction stone

Figure 8: interaction

(a) GrabCut moon after interaction (b) GrabCut stone after interaction

Figure 9: GrabCut after interaction

4 Experimental Results

For the testing of GrabCut implementation, this project used the dataset with image and ground truth segmentation from University of Oxford. Figure[6] shows the original images plant and stone I tested my GrabCut on. As Figure[6] through Figure[9], it shows the implementation steps for running GrabCut on image and user interaction, then GrabCut again to get higher accuracy.

The accuracy achieved for initial GrabCut for moon is 75% and increases to 91% accuracy after interaction.
The accuracy achieved for initial GrabCut for stone is 88% and increases to 95% after interaction.
As shown in the first page for the flower, the accuracy achieved was 98% without interaction.

5 Conclusion

The GrabCut algorithm is learned to be the iterative Graph Cuts that optimize segmentation through the min-cut method to partition the segments for foreground and background accurately. And, user editing with a light load to brush on the initial segmented image from iterations of Graph Cuts easily increase the accuracy of segmentation after editing as shown in the experimentation.
Future learning is expected to optimize the Graph Cuts in GrabCut algorithm with optimizations related in various papers discussed in Related Work section.

Acknowledgments

References

[1] Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3) (2004) 309314

[2] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000. doi: 10.1109/34.868688

[3] Lempitsky, V.S., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV. (2009) 277284

[4] E. Zemene and M. Pelillo, Interactive image segmentation using constrained dominant sets, in European Conference on Computer Vision (ECCV), 2016, pp. 278294.

[5] D. Freedman, T. Zhang, Interactive graph cut based segmentation with shape priors, Computer Vision and Pattern Recognition (2005) 75762, I.

[6] A. Joulin, F. Bach and J. Ponce, "Discriminative clustering for image co-segmentation," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 1943-1950. doi: 10.1109/CVPR.2010.5539868

[7] Eriksson, Anders & Barr, Olof & strm, Kalle. (2006). Image segmentation using minimal graph cuts.