Indian Institute of Technology Indore

CS 419 Computer Vision

GrabCut
Interactive Foreground Extraction

Group_10

Snehashriie Bhukya (180001013)
Jagruthi Patibandla (180001021)
Shravya Ramasahayam (180001052)

December 3, 2021

1 Introduction

Foreground extraction is a process of extracting the foreground of an image from its background. It plays an important role in computer vision applications, such as photo editing, photo enhancement, image classification, image and video understanding, surveillance systems, style transfer improvement, etc. This problem can be solved by employing image segmentation techniques. We will be using GrabCut to solve the problem of interactive foreground extraction of an object whose separation from its background is composite, like in cases where the edges of the object to be segmented are fuzzy. It aims to achieve high performance at the cost of minimal interactive efforts by the user.

2 Methodology

GrabCut is an interactive segmentation. Based on the region selected by the user, the algorithm tries to figure our foreground and background of the image. The pixels selected are called as 'seeds'. A pixel can either be labelled as definitely background or definitely foreground. This is a means of hard segmentation.

Let α be the matte, generally $0 \leq \alpha \leq 1$. For hard segmentation, $\alpha \in \{0, 1\}$.

$$\alpha_i = \begin{cases} 0 & if\ i^{th}\ pixel\ is\ background \\ 1 & if\ i^{th}\ pixel\ is\ foreground \end{cases}$$

Calculation of rest of the foreground and background pixels is performed using these seed pixels. The region and boundary information is used to create an energy function which, when minimized, produces the best segmentation.

Algorithm 1 GrabCut Algorithm

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 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

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.

3 Dataset used

We used the dataset given here. It has the ground truth alpha matte which is used to evaluate the performance of the model.

4 Implementation

4.1 Grab-Cut

Grab-Cut is an innovative segmentation technique that uses:

- Both region and boundary information contained in an image
- Graphs to store region and boundary information.
- A Min-Cut/Max-Flow algorithm

Construct a graph whose nodes represent the pixels in an image. Create 2 extra nodes source and sink. The source node represents the foreground while the sink node represents the background. Every pixel of the image is connected to both the source and sink nodes. Each edge has a weight. The higher weight between pixels implies that they are similar to each other. The weight between the sink node and the pixels, the source node and the pixels is determined by using the GMM model. To achieve foreground extraction we need to find a minimum cut in the graph constructed that separates the source and the sink nodes.

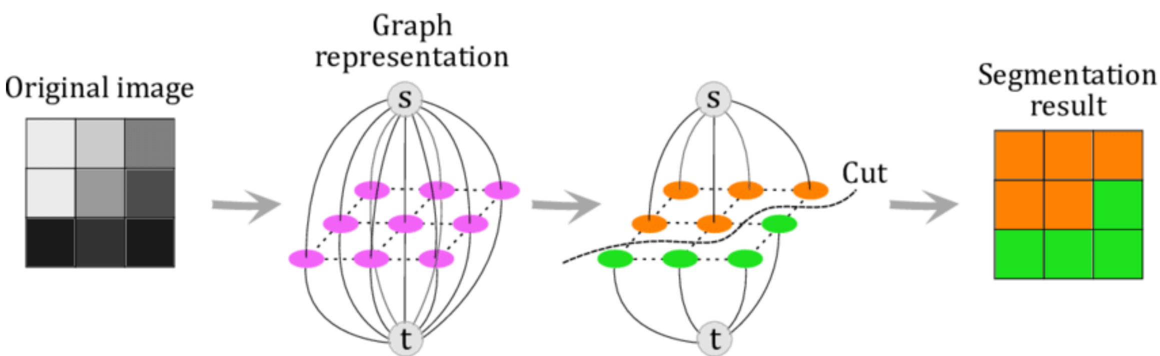


Figure 1: Pictorial representation of Graph-cut Algorithm

4.2 GMM Model

In order to get the Background and Foreground labeling weights, we generate two k-Gaussian Mixture Models. A clustering algorithm is used to determine the means, covariance and mixture parameters of the colour clusters in the background and foreground Gaussians. The weights between a pixel node and Source node are determined by the probability that the pixel node lies inside the foreground Gaussian. Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background Gaussian.

4.3 User Interaction

The user gives "hints" to the algorithm by marking the image pixels as: background pixel, foreground pixel, probable background pixel, probable foreground pixel. The algorithm simultaneously displays an extracted foreground window along with the alpha matting window. For a pixel i which is part of foreground: W_{iF} = inf and $W_{iB} = 0$. For a pixel i which is part of background: W_{iB} = inf and $W_{iF} = 0$ (where W_{iB} = weight between sink and other pixel nodes and W_{iF} = weight between source and other pixel nodes).

(a) Input Image

(b) User Interaction

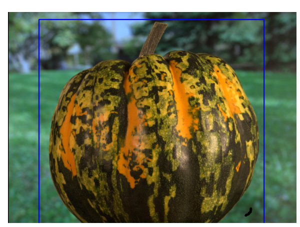
(c) Foreground

(d) Alpha Matte

4.4 Results

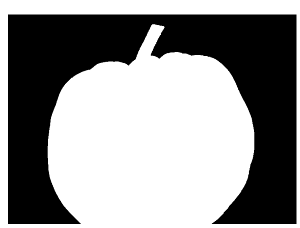
The code was tested on random images taken from the dataset. The accuracy was computed by comparing the obtained matting output with the ground truth matting given along with the dataset. The accuracy on average for pictures with a clear distinct boundary between foreground and background was around 95% . The algorithm has produced acceptable performance with minimum user interaction.

(a) Input Image



(b) User Interaction

(c) Foreground



(d) Alpha Matte

When the input has fuzziness or camouflage portions, we can get an acceptable accuracy with little more user interaction. In the flowerpot example given below, user has interacted by giving the box for foreground and then 6 instances of sure background scribbles. The accuracy achieved is around 86%.

(a) Input Image

(b) User Interaction

(c) Foreground

(d) Alpha Matte

4.5 Conclusions

The algorithm performs well with considerable user interaction. The cases of special interest are:

- Input has a fuzzy boundary between the foreground and background
- Camouflage input i.e. there is an overlap in the colorspace of the foreground and the background.

In both the cases, GrabCut produces better matting and foreground extraction compared to standard matting techniques like magic wand, intelligent scissors.

4.6 References

[1] Rother, Carsten Kolmogorov, Vladimir Blake, Andrew. (2004). GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. ACM Trans. Graph.. 23. 309-314. 10.1145/1186562.1015720.

[2] https://www.cs.ru.ac.za/research/g02m1682/.

[3] Gauriau, Romane. (2015). Shape-based approaches for fast multi-organ localization and segmentation in 3D medical images.