Improved GrabCut Technique for Segmentation of Color Image

Basavaprasad B.
Department of Computer Science
Government First Grade College
Raichur – 584101, India

Ravindra S. Hegadi
Department of Computer Science
Solapur University
Solapur – 413255, India

ABSTRACT
An improved method of the GrabCut Technique has been implemented in this paper which works on image segmentation quite interactively and user friendly and which reduces the user effort. This paper emphasizes on modification of GrabCut image segmentation which is an iterative algorithm that combines statistics and Graph Cut in order to accomplish detailed image segmentation with proper input. The proposed algorithm requires an initial selection of object to be segmented. The algorithm will deflate to capture the object of interest, which has different image feature as compared to its background. This algorithm does not need any more user intervention during its segmentation process. The proposed algorithm could achieve an effective segmentation of objects from background for different classes of images.

General Terms
Image processing, graph based segmentation.

Keywords
GrabCut, Graph cut, GMM, color clustering, border-matting, energy minimization, foreground, background, Gibbs energy, Alpha Matting optimization.

1. INTRODUCTION
Graph based algorithms have recently gained popularity for implementation in image segmentation tasks in computer vision. These algorithms use better and combined statistical models with them. The GrabCut technique is one of such graph based technique which may be implemented using Graph Cut techniques as a part of its refining process of initial user segmentation between foreground and background of image. GrabCut algorithm is originally developed at Microsoft Research Cambridge, UK. The graph-cut approach can be improved in three aspects. Firstly, through the development of more powerful, iterative form of the optimization techniques, secondly, the strength of the iterative algorithm may be used to simplify significantly the user interaction which is essential for a given good quality of result, and thirdly, an algorithm for “border-matting” has been developed for the simultaneous estimation of both the alpha-matte around an object boundary and the colors of foreground pixels [1]. The user only needs to input a very uneven segmentation between foreground and background. Characteristically this is done by drawing a rectangle around the objective of interest. The way that it is achieved technically is by using a grouping of both Graph Cuts and statistical models of the foreground and background structure in the color space [2]. This paper emphasizes on modification of GrabCut image segmentation which is an iterative algorithm that combines statistics and Graph Cut in order to accomplish detailed image segmentation with proper input. The proposed algorithm requires an initial selection of object to be segmented. The algorithm will deflate and will capture the object of interest, which has different image feature as compared to its background. This algorithm does not need any more user intervention during its segmentation process. The graph technique for image segmentation is discussed in section 2. The formulation of the proposed GrabCut based algorithm is described in section 3. The modeling of object foreground and its background is discussed in section 4. Section 5 describes the results and conclusions are drawn in section 6.

2. BACKGROUND
In this section we define some terminologies that will be used throughout the paper for explaining the graph based image segmentation methods. Let $G = (V, E)$ be a graph, where $V = \{v_1, v_2, \cdots, v_n\}$ is a set of vertices corresponding to the image elements, which are represented as pixels or regions in the Euclidean space. E is set of edges connecting certain pairs of neighbouring vertices. Each edge $\boldsymbol{e}(v_i, v_j)$ has a corresponding weight $W(v_i, v_j)$ which measures certain quantity based on the property between the two vertices connected by an edge. Image is partitioned into mutually exclusive components, such that each component A is a connected graph $G' = (V', E')$, where V'contains all the vertices of G' and E' continuous edges built by set of nodes of V'. The well-accepted segmentation criteria require that image elements in each component should have uniform and homogeneous properties in the form of brightness, colour or texture, etc., and elements in different components should be dissimilar.

There are many segmentation methods. Among them graph theoretical techniques have more features in practical applications. These techniques categorize the image into mathematically well-defined structures, making the formulation of image segmentation problem more accurate and the computations more efficient. In these techniques the image is treated as a weighted and undirected graph [5]. In graph theoretic definition, the degree of dissimilarity between two components can be computed using graph cut methods, which are related to a set of edges by which the graph G will be partitioned into two disjoint sets. The segmentation of an image can be interpreted in form of graph cuts. In image segmentation, noise and other ambiguities bring uncertainties in order to understand the image content. The accurate solution to image segmentation is difficult to obtain. Therefore, it is more appropriate to solve this problem with Minimal methods. The optimization-based approach formulates the problem as a minimization of some established criterion, whereas one can find an exact or approximate solution to the original certain visual problem. In this case, the optimal bi-partitioning of a graph can be taken as the one which minimizes the cut value. In a large volume of literature image segmentation is also formulated as a labeling problem, where a set of labels L is assigned to a set of sites in S. In two-class segmentation, for example, the problem can be described as assigning a label S_i from the set

$L = \{object, background, foreground\}$ to site S_i where the elements in S are the image pixels or regions. Labeling can be performed separately from image partitioning, while they achieve the same effect on image segmentation. We may find in literature that many methods perform both partitioning and labeling simultaneously.

Graph cut techniques can be used efficiently to solve image segmentation problems. Using these techniques image segmentation problems can be formulated in terms of energy minimization which in turn can be formulated as max-flow problem in a graph [2].

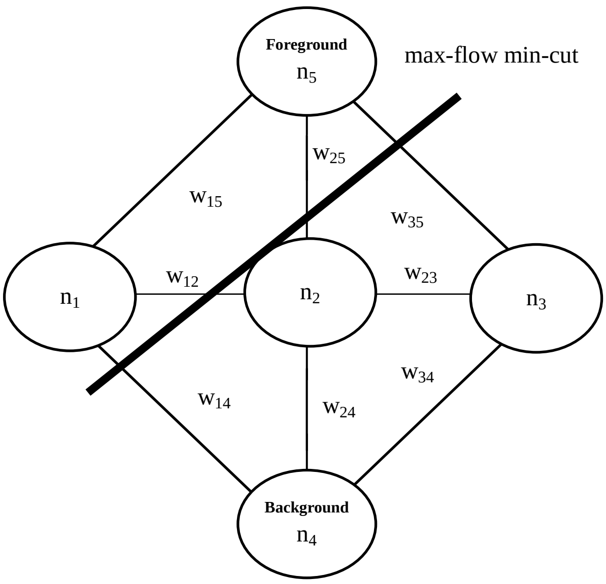


Figure 1: A graph being segmented into foreground and background

3. IMPROVED GRABCUT ALGORITHM

3.1 Data structures:

GrabCut requires four different bits of information for each pixel. Each image is stored in its own array. Each array is the same size as that of the original image.

i. Colour – an RGB value (z)
ii. Trimap – either TrimapUnknown, TrimapBackground, or TrimapForeground
iii. Matte – in the initial hard segmentation step, either Matte-Background or MatteForeground
iv. Component Index – a number between 1 and K, where K is the number of Gaussian components in a GMM (k).

The image is now obtained which consists of pixels z_n in RGB colour space. As it is impractical to construct acceptable colour space histograms; we use Gaussian Mixture Models (GMMs) which are in practice already used for soft segmentation. Each GMM, one for the background and one for the foreground, is taken to be a full-covariance Gaussian mixture with K components (typically $K = 5$). In order to deal with the GMM tractably, in the optimization framework, an additional vector $K = \{k_1, \cdots, k_n, \cdots, k_N\}$ is introduced, with Component Index $k_n \in \{1, \cdots, K\}$, assigning, to each pixel, a unique GMM component, one component either from the background or the foreground model, accordingly as $\alpha_n = 0\ or\ 1^1$.

The Gibbs energy for segmentation now becomes

$$E(\underline{\alpha}, \boldsymbol{k}, \underline{\theta}, \boldsymbol{z}) = U(\underline{\alpha}, \boldsymbol{k}, \underline{\theta}, \boldsymbol{z}) + V(\underline{\alpha}, \boldsymbol{z})$$

depending also on the GMM component variables $\boldsymbol{k}$. The data term U is now defined, taking account of the colour GMM models, as

$$U(\underline{\alpha}, \boldsymbol{k}, \underline{\theta}, \boldsymbol{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, \boldsymbol{z_n}),$$

where

$$D(\alpha_n, k_n, \underline{\theta}, \boldsymbol{z}) = -\log p(\boldsymbol{z_n} | \boldsymbol{\alpha_n}, k_n, \underline{\theta}) - \log \pi(\boldsymbol{\alpha_n}, \boldsymbol{k_n})$$

and $p(\cdot)$ is a Gaussian probability distribution, and $\pi(\cdot)$ are mixture weighting coefficients, so that (up to a constant):

$$\begin{aligned} &D(\alpha_n, k_n, \underline{\theta}, \boldsymbol{z}) \\ &= -\log \pi(\boldsymbol{\alpha_n}, \boldsymbol{k_n}) \\ &+ \frac{1}{2} \log det \sum(\alpha_n, k_n) \\ &+ \frac{1}{2}[z_n \\ &- \mu(\alpha_n, k_n)]^T \sum(\alpha_n, k_n)^{-1} [z_n \\ &- \mu(\alpha_n, k_n)]. \end{aligned}$$

Therefore, the parameters of the model are now

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \textstyle\sum(\alpha, k), \alpha = 0, 1, k = 1, \cdots K\},$$

Where π represents the component weights, μ represents the mean (an RGB triple) and $\sum$ represents the covariance of the $2K$ Gaussian components for the background and foreground distributions.

The smoothness term V is basically unchanged from the monochrome except that the contrast term is computed using Euclidean distance in colour space:

$$V(\underline{\alpha}, Z) = \gamma \textstyle\sum_{(m,n)\in C} [\alpha_n \neq \alpha_m]^{-\beta \|z_m - z_n\|^2}.$$

3.2 The GrabCut segmentation algorithm:

This section describes the GrabCut hard segmentation algorithm which is iterative image segmentation in GrabCut Colour data modelling [1].

Algorithm

- Initialize trimap T by supplying only T_B, the fore-ground is set to $T_F = \Phi$; $T_U = T_B$, complement of the back-ground.
- Initialize $\alpha_n = 0\ for\ n \epsilon T_B\ and\ \alpha_n = 1\ for\ n \in T_U$.
- Background and foreground GMMs initialized from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Iterative minimization Algorithm

1. Assign GMM components to pixels: for each n in T_U,
$$k_n := arg \min_{k_n} D_n(\alpha_n, k_n, \underline{\theta}, \boldsymbol{z_n})$$
2. Learn GMM parameter from data Z:

$$\underline{\theta} := arg \min_{\underline{\theta}} U(\underline{\alpha}, k, \underline{\theta}, \boldsymbol{z})$$

3. Estimate segmentation: use min cut to solve
$$\min_{\{\alpha_n : n \epsilon T_U\}} \min_{k} E(\underline{\alpha}, k, \underline{\theta}, \boldsymbol{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting.

User Editing

- Edit: fix some pixel either to $\alpha_n = 0$ (back-ground brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- Refine operation: perform entire iterative minimization algorithm.

We propose the GrabCut algorithm which will improve the performance by reducing the user effort and also user friendly.

3.3 The proposed modified algorithm:

1. The user input three things: The foreground, background and the unknown part of the image that can be either foreground or background. This is normally done by selecting a rectangle around the object of interest. The region inside that rectangle as unknown and pixel outside the rectangle as known background marked automatically. The user can select the image to be segmented just by browsing it from the dialog box.
2. The algorithm creates an initial image segmentation, where the unknown pixels are placed in the foreground class and all known background pixels are classified as background with different partitions using Gibbs energy.
3. The foreground and background are modelled as Gaussian Mixture Models (GMMs) using the Orchard-Bouman clustering algorithm with additional K Vector for framing purpose for every iteration of the algorithm which improve the performance of the algorithm**.**
4. Every pixel in the foreground assigned most probable Gaussian component in the foreground GMMs. The same process is done with the pixels in the background but with components of the background GMMs.
5. New GMMs are learned from the pixel sets that where created in the previous step.
6. A graph is built and Graph Cut is generated automatically to find a new classification of foreground and background pixels.
7. Repeat step 4 – 6 until the classification converges.

4. MODELLING FOREGROUND AND BACKGROUND

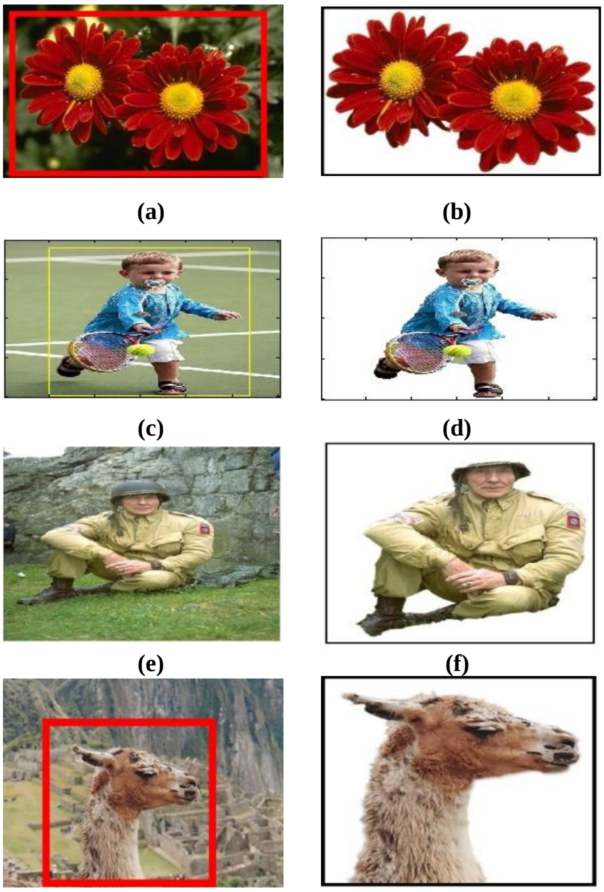
The initial information about the foreground and the background are provided by the user through selection of rectangular around the object of interest. Pixels outside this selection are treated as known background and the pixels inside are marked as unknown. From this data we want to create a model that we can use to determine if the unknown pixels are either foreground or background [2].

In the GrabCut algorithm this is done by creating K components of multivariate Gaussian Mixture Models (GMM) for the two regions: K components for the known background and K components for the region that could be the foreground, giving a total of $2K$ components. The GMM components have the same dimensions as the colour space and are derived from the colour statistics in each cluster. In order to get good segmentation we want to add components with low variance as this makes the cluster easier to separate from the others. There are a lot of proposed ways to create clusters with this property. We tested the colour quantization technique described by Orchard and Bouman [4], that were suggested in implementing GrabCut by Justin F. Talbot and Xiaoqian Xu [3] which works well.

4.1 Colour clustering:

In order to calculate the GMMs, the pixels need to be clustered in some way in order to determine the statistics. This was done by using a binary tree quantization algorithm described by Orchard and Bouman [4]. All pixels are placed in the same cluster in the beginning. The cluster is then slatted around its mean values projection on the first principal component. Colour clustering algorithm is explained in [3].

5. RESULTS



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)**

Figure 2: Segmentation using Improved GrabCut

Figure 2 shows the results of proposed algorithm. Each image in this figure shows that the bounding rectangle alone is sufficient for user interaction to enable foreground extraction to be completed automatically by GrabCut. Different examples are shown after simplification of the problem which is increasingly more problematic in terms of substantial amount of user interfaces that can occur in three cases:

i. Regions of low contrast at the transition from foreground to background.
ii. The true foreground and background distributions overlap partially in colour space.
iii. Background material inside the user rectangle need not to be sufficiently represented in the background region. User interactions for the first case can be reduced by replacing the rectangle with a lasso in Figure 2(b). The first image demonstrates that border matting method can

cope with moderately difficult alpha mattes. For difficult alpha mattes the matting brush is needed.

We tested different images in order to evaluate the performance and accuracy of the segmentation.

5.1 Comparison of Graph Cut and GrabCut

In a first experiment the amount of user interaction for 15-20 segmentation tasks were evaluated. In 10 moderately difficult examples (e.g. flowers) GrabCut needed significantly less interactions. Otherwise both methods performed comparably.

In a second experiment we compare GrabCut using a single outside, Lasso (e.g. red line in Figure 2(a) with Graph Cut using two lassos, outer and inner (e.g. red and white lines in Figure 2(a) & 2(b)). Even with missing foreground training data, GrabCut performs at almost the same level of accuracy as Graph Cut.

In both the experiments we need very less user interactions.

Method	**Image Database**	**Quality and Effort**
Graph Cut	200	Medium quality for moderately difficult image with large degree of user effort
Improved GrabCut	200	Good quality for moderately difficult images with a rather modest degree of user effort

6. CONCLUSIONS

A new algorithm for foreground extraction has been proposed and demonstrated, which obtains foreground alpha mattes of good quality for moderately difficult images with a rather modest degree of user effort. The system combines soft segmentation rather than hard segmentation by iterative graph cut optimization with border matting to deal with blur and mixed pixels on object boundaries. This paper emphasizes on modification of GrabCut image segmentation, which is an iterative algorithm that combines statistics and Graph Cut in order to accomplish detailed image segmentation with proper input. The proposed algorithm does not need much manual work during its segmentation process. Overall the resulting segmentation image will be of good quality for moderately difficult images with a modest degree of user effort.

7. REFERENCES

[1] Carsten Rother, Vladimir Kolmogorov and Andrew Blake, Microsoft Research Cambridge, UK. “GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts, ACM SIGGRAPH, Volume: 23 Issue: 3, Pages: 309-314, August 2004.

[2] Ravindra S. Hegadi, Basavaraj A Goudannavar, "Interactive Segmentation of Medical Images Using GrabCut", IJMI, Volume: 3, Issue: 3, Pages: 168-171, 2011.

[3] Justin F. Talbot, Xiaoqian Xu., “Implementing GrabCut”, Brigham Young University, April 7, 2006.

[4] Orchard, M. T., and Bouman, C. A., “Color Quantization of Images”, IEEE Transactions on Signal Processing Volume 39: Issue: 12, Pages: 2677-2690, 1991.

[5] Basavaprasad B., and Ravindra S. Hegadi.; “Graph theoretical approaches for image segmentation”, Aviskar – Solapur University Research Journal, Volume: 2, Pages: 7-13, 2012.

[6] Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. “Interactive Image Segmentation using an adaptive GMMRF model”, Computer Vision – ECCV, Volume: 3021, Pages: 428-441, 2004.

[7] Kolmogorov, V., Zabin, R. "What energy functions can be minimized via graph cuts?” IEEE Transactions on pattern analysis and Machine Intelligence, Volume: 26, Issue: 2, Pages: 147 – 159, Feb, 2004.

[8] Saban, M.A.E., "Interactive segmentation using curve evolution and relevance feedback", International Conference on Image Processing, Volume: 4, Pages: 2725 – 2728, Oct. 2004.

[9] Chuang, Y., Curless, B., Salesin, D. and Szeliski, R.A “Bayesian approaches to digital matting”, IEEE Conference on Computer Vision and Pattern Recognition, Volume: 2, Pages: 264-271, 2001.

[10] Kass, M., Witkin, A. and Terzopoulos, D., “Snakes: Active contour models”. In Proc. IEEE International Conference on Computer Vision, Pages: 259–268, 1987.

[11] Boykov, Y., "Min-cut and Max- flow algorithms for energy minimization in vision", IEEE Transactions on, Volume: 26, Issue: 9, Pages: 1124 – 1137, Sept. 2004.

[12] BOYKOV, Y., AND JOLLY, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images", ICCV, Volume: 1, Pages: 105–112, 2001.

[13] Jonathan L. Gross, Jay Yellen, A text book on “Graph Theory and its Applications”, Second Edition, Chapman and Hall, 2005.

[14] R.C. Gonzalez and R.E. Woods, A text book on “Digital Image Processing”, Second Edition, Pearson Education 2002.