Implementing the “GrabCut” Segmentation Technique as a Plugin for the GIMP

Matthew Marsh*
Rhodes University
Computer Science Department
Grahamstown, South Africa

Shaun Bangay†
Rhodes University
Computer Science Department
Grahamstown, South Africa

Adele Lobb‡
Rhodes University
Computer Science Department
Grahamstown, South Africa

Abstract

Image segmentation requires a segmentation tool that is fast and easy to use. The GIMP has built in segmentation tools, but under some circumstances these tools perform badly. “GrabCut” is an innovative segmentation technique that uses both region and boundary information in order to perform segmentation. Several variations on the “GrabCut” algorithm have been implemented as a plugin for the GIMP. The results obtained using “GrabCut” are comparable, and often better than the results of all the other built in segmentation tools.

CR Categories: I.4.6 [Computing Methodologies]: Image Processing and Computer Vision—Segmentation

Keywords: Image segmentation, ”GrabCut”

1 Introduction

Image editing requires the ability to quickly and easily extract foreground objects from digital images. This process of separating an image into different parts is known as image segmentation. Humans perform image segmentation very easily by using object recognition. We find this to be an extremely simple task, but it is definitely not a trivial task for computers.

In order for computers to perform image segmentation the algorithm performing the segmentation needs to use information encapsulated in the digital image to calculate the best segmentation. Computers cannot easily recognise objects, and so edge detection and region detection methods are most commonly used in order to segment images.

The GIMP is an image editing package that requires an easy to use segmentation tool that produces good results. It already contains a magic wand segmentation tool and an intelligent scissors segmentation tool, but as the results of this project show, these tools are often cumbersome and perform badly under specific conditions (see section 5.1).

“GrabCut” is an innovative segmentation technique that uses both region and boundary information contained in an image in order to perform segmentation [Rother et al., 2004]. The aim of this project is to provide a segmentation tool for the GIMP that is quick and easy to use, by implementing the “GrabCut” segmentation technique.

2 Related work

There are many different image segmentation techniques which have already been developed. The majority of these techniques use either region or boundary information contained in the image in order to perform segmentation. Boundary information is the edge information in an image which can be gained by noting the difference between the colours of nearby pixels. A large difference in colour will indicate an edge. A region is the area enclosed by an edge. The colour of neighbouring pixels in a region will be similar.

Thresholding is the simplest form of segmentation, and works by comparing each pixel in an image to a threshold value. This segmentation technique ignores all boundary and region information in the image. Thresholding has been implemented as the Magic Wand tool in PhotoShop [INCORP, 2002], the GIMP and most image editing packages.

Edge detection segmentation techniques minimize an energy function in order to calculate the best segmentation. [Mortensen and Barrett, 1995] introduce an Intelligent Scissors tool based on live wire boundaries. Various points on the segmentation boundary are indicated and the points in between are interpolated by minimizing an energy function. Most image manipulation packages include some variation on Intelligent Scissors. [Kass et al., 1987] describe an edge detection mechanism called Snakes, which also minimizes an energy function.

Image segmentation can also be performed by using graphs. [Boykov and Jolly, 2001] introduce a greyscale segmentation technique which uses both region and boundary information in order to perform segmentation. A graph is built up with nodes in the graph representing pixels in the image. Weights between nodes are determined by an energy function, such that a minimum cost cut on the graph produces the best segmentation of the image. Region information is modeled by using greyscale histograms. [Rother et al., 2004] extend the technique by [Boykov and Jolly, 2001] to work with colour by using Gaussian Mixture Models.

3 Design

Section 3.1 gives an overview of how the “GrabCut” algorithm works and how graph cuts are used to perform segmentation. This section is followed by section 3.2 which describes three methods that are used to model region information contained in images. The first method is for greyscale images and the second and third are for colour images.

*e-mail:g02m1682@campus.ru.ac.za

†e-mail:s.bangay@ru.ac.za

‡e-mail:a.lobb@ru.ac.za

AFRIGRAPH 2006, Cape Town, South Africa, 25–27 January 2006.
© 2006 ACM 1-59593-288-7/06/0001 $5.00

3.1 How “GrabCut” Works

“GrabCut” is a segmentation technique that uses graph cuts to perform segmentation. Like most segmentation techniques “GrabCut” uses information encapsulated in the image. Most segmentation techniques make use of either edge information or region information in the image. “GrabCut” makes use of both edge and region information. This information is used to create an energy function which, when minimized, produces the best segmentation.

In order to perform segmentation a graph is built, where nodes in the graph represent pixels in the image. In addition two special nodes are also created. These are the Sink and Source nodes. Every pixel node in the graph is connected to the Source and Sink node. The Source node represents the foreground of the image, and the Sink node the background. In order to segment the image the Source and Sink nodes must be separated. Figure1 shows a simplified graph. Links between the pixel nodes and Source and Sink nodes have been removed in this graph for clarity. The linked spheres indicate pixel nodes, and the Source and Sink node are labelled as such.

The energy function is incorporated into the graph as weights between pixel nodes and weights between pixel and Source or Sink nodes. Weights between pixel nodes are determined by edge information in the image. Thus a strong indication of an edge between two pixels (a large difference in pixel colour) results in a very small weight between two pixel nodes. The region information determines the weights between pixels nodes and the Source and Sink nodes. These weights are calculated by determining the probability of the pixel node being part of the background or foreground region.

In order for foreground and background regions to be created, some pixels in the image need to be labelled before segmentation as either foreground or background. This is referred to as the clue marking stage. Any pixels that are labelled during this stage are set as hard constraints. This means that during the segmentation process, hard labelled pixels cannot change their labelling.

The Min-cut/Max-Flow algorithm created by [Boykov and Kolmogorov, 2004] is used in order to segment the graph. This algorithm determines the minimum cost cut that will separate the Source and Sink nodes. The cost of the cut is determined by the sum of all the weights of the links that are cut. Once the Source and Sink nodes are separated, all pixel nodes connected to the Source node become part of the foreground, and the rest become part of the background.

3.2 Variations on the “GrabCut” Technique

In order to create an energy function, region and boundary information needs to be extracted from the image. Three different techniques are implemented for this purpose:

1. For greyscale images boundary information is obtained by taking the difference in intensity values between neighbouring pixels. A histogram is used to model region information.

2. Gaussian mixture models and an EM clustering algorithm are used to model region information in colour images, and the boundary information is obtained by taking the difference in colour values between neighbouring pixels. The EM clustering algorithm that is used is an implementation based on the suggestions in a book by [Kung et al., 2004] in the section entitled Expectation-Maximization Theory.

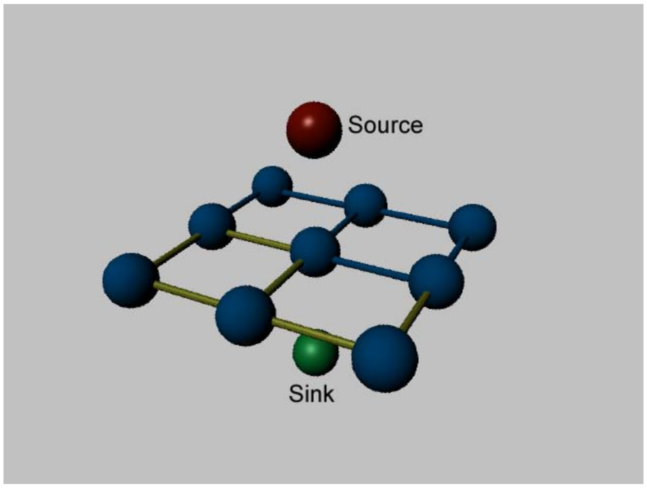


Figure 1: Relationship between image pixels and graph representation

3. A histogram for each colour channel is created in order to model region information, and the boundary information is obtained by taking the difference in colour values between neighbouring pixels.

Histograms are built up by noting the frequency at which a pixel occurs within a group of pixels. For greyscale images the intensity of each pixel can range between 0 and 255. A histogram with 256 bins is created for greyscale images, so that each bin holds the frequency at which pixels with the intensity of the bin number occur in an image. The red, green and blue channels in colour images also have a possible range of 0 to 255, so histograms are built up in the same way.

3.2.1 Greyscale

Pixels marked as either foreground or background during the clue marking stage are used to model region information by building up foreground and background histograms. The weights between a pixel node and the Source node are determined by the probability that the pixel node lies inside the foreground histogram. Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background histogram.

3.2.2 Colour using Gaussian Mixture Models

[Rother et al., 2004] use Gaussian Mixture Models in order to model colour regions in images. A foreground and background Gaussian Mixture Model are created by using the colour of pixels provided in the clue marking stage. The EM clustering algorithm is used to determine the means, covariance matrices and mixture parameters of five colour clusters in the background and foreground Gaussians. The weights between a pixel node and the 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.

The EM clustering algorithm as currently implemented, tends to be extremely sensitive to local maxima. This means that once a cluster is detected it will be used as one of the five clusters, even though better solutions may exist. It also tends to create very large

Gaussians encompassing all the data points, instead of creating 5 small Gaussians. The process of determining the Gaussians is very slow.

When Gaussian Mixture Models are implemented the “GrabCut” algorithm becomes iterative. First the initial Gaussian Mixture Models are calculated and all pixels are labelled as either background or foreground. The following steps are then repeated until convergence.

1. Estimate the means, covariance matrices and mixing parameters of the background and foreground Gaussian Mixture Model based on the current segmentation.

2. Assign to every pixel a Gaussian component from the foreground and background model.

3. Create the graph and segment it, marking pixels as either foreground or background.

The foreground and background marking method differs from the greyscale case. In this case the user drags a rectangle around the foreground object, and the inverse of the selection is set as hard background. The pixels inside the rectangle are allowed to change between foreground and background on each iteration.

3.2.3 Colour using Histograms

This implementation is the same as the greyscale version, except here histograms are built up for each colour channel. The probability that a pixel lies in the foreground is then calculated by the sum of the probabilities that its red value lies in the red foreground histogram, its blue value in the blue foreground histogram and its green value in the green foreground histogram. The weights between a pixel node and the Source node are determined by the probability that the pixel node lies inside the foreground region. Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background region.

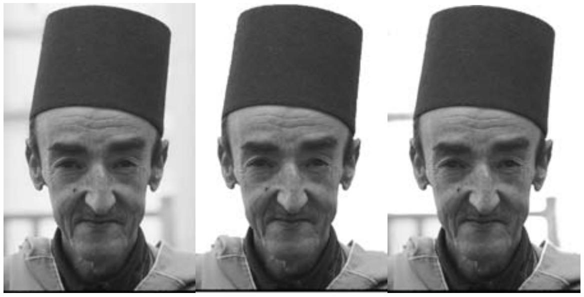
Colour histograms do not fully represent colour regions in an image. This is because the histograms are separate, and have no concept that a colour is made up of 3 RGB values. Gaussian Mixture Models are much better at representing colour regions as they cluster actual colours, not individual RGB values.

4 Implementation

The plugin is based on a plugin template for the GIMP. The GIMP development toolkit is used to create a basic graphical interface that allows the selection of a foreground and background brush to label pixels in the clue marking stage.

A transparent layer is created above the image, and the user is allowed to paint clues onto this layer. This allows the brush strokes to be captured, while retaining the integrity of the original image.

A 2D array is used to store the labeling of the pixels. The actual segmentation process just changes the labelling of pixels in this array without affecting the image. The alpha layer of the image is then edited to reflected this labelling, and the GIMP’s built in feature of being able to convert an alpha layer to a selection is used to select just the pixels belonging to the foreground region.



(a) Original Image (b) Segmentation using "GrabCut" (c) Segmentation using Magic Wand

Figure 4: A comparison between the Magic Wand tool and “GrabCut”

5 Results

The following results are all obtained using Linux on a 3 Ghz Pentium 4 PC with 1 Gig of RAM.

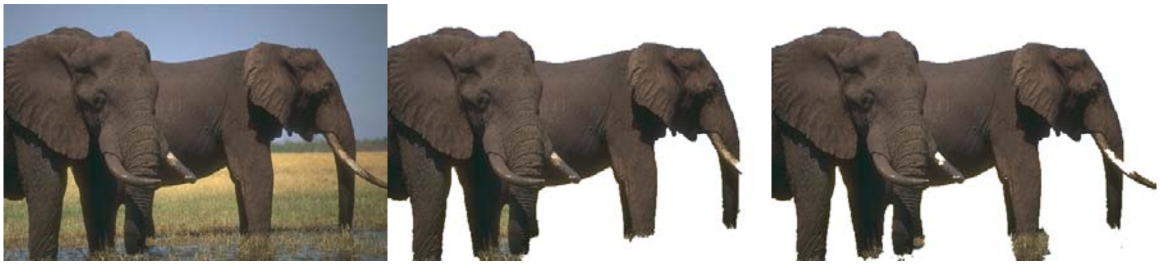
The segmentation results on black and white images are very good. The processing time is quick (on average a few seconds), and the user interaction process is simple.

The Gaussian colour implementation is not satisfactory. The process of calculating the starting Gaussian parameters is slow and the initial values significantly affect the segmentation results. Further improvements to the EM Clustering implementation are currently underway. In order to test the rest of the process some manually chosen starting parameters for the Gaussians are hard coded in for each image. The results here are reasonable, but the process is very slow due to the iterative nature of the algorithm.

The Colour implementation using histograms produces very good results, and is still quick. The results using this technique are at least as good, and often better than the much slower Gaussian method. Figure 2 shows a comparison between segmentation results using these two methods, and the time it took to perform the segmentation. Figure 3 shows some segmentation results obtained by using the colour histogram implementation.

5.1 Comparing “GrabCut” to the Segmentation Tools Built Into GIMP

The “GrabCut” plugin performs as well or better than the other segmentation tools built into the GIMP. The Magic Wand tool is easy to use; however it only produces good segmentation results on very simple images. Figure 4 shows how “GrabCut” produces better results than the Magic Wand tool on a simple image. The Intelligent Scissors tool is powerful, and the results are comparable to the results achieved with “GrabCut”, however it can be very cumbersome. It places a large work load on users if they attempt to segment images that are highly textured. “GrabCut” places a relatively low work load on users for all images. Figure 5 shows how a highly textured image is easily segmented using “GrabCut”. Few clues need to be given, and the actual segmentation processing took 3 seconds. Segmenting this image using Intelligent Scissors took 2 minutes and produced poor results.

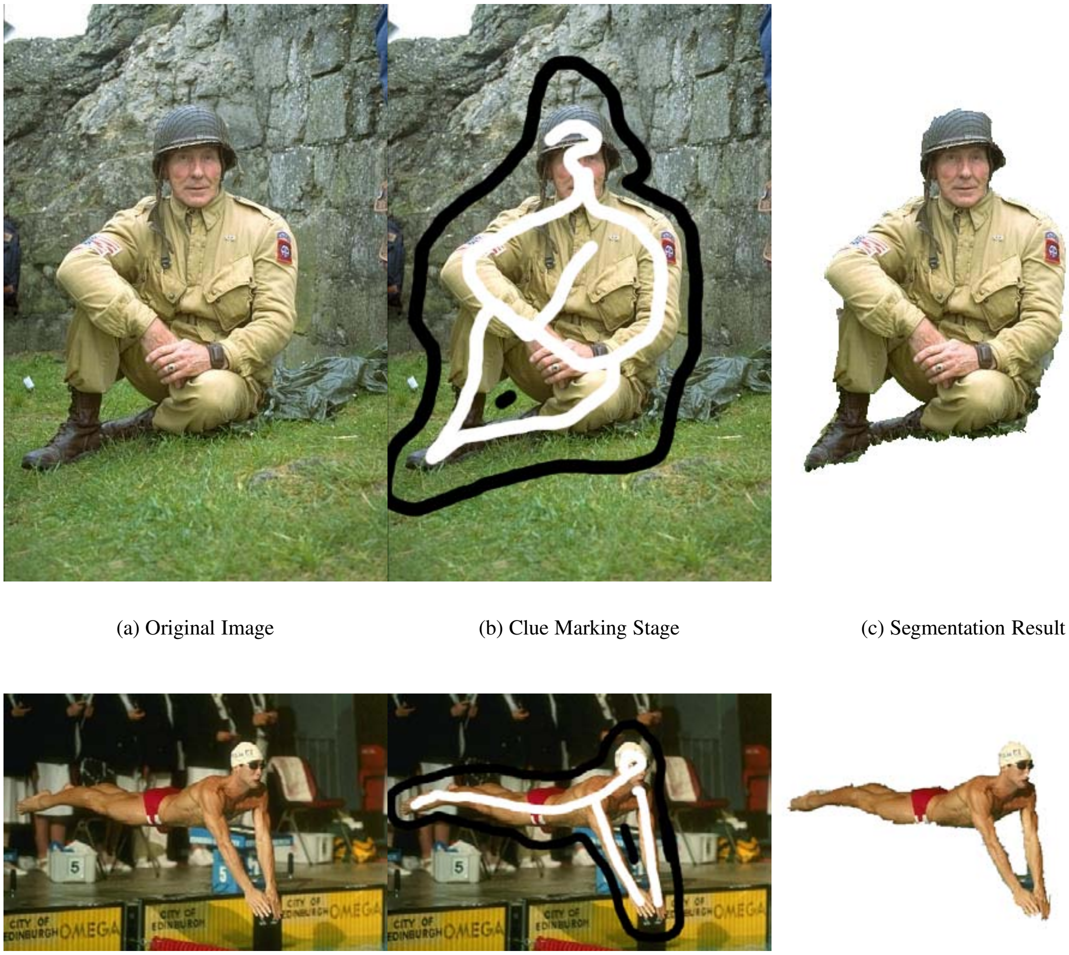


(a) Original Image

(b) Image Segmented Using Colour Histgrams (3s to segment)

(c) Image Segmented Using Gaussian Mixture Models (35s to segment)

Figure 2: Colour segmentation results



(a) Original Image

(b) Clue Marking Stage

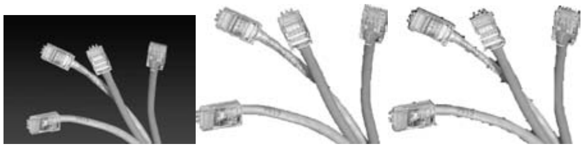
(c) Segmentation Result

(d) Original Image

(e) Clues Drawn

(f) Segmentation Result

Figure 3: Segmentation results using colour histograms



(a) Original Image

(b) Fine Textures are Segmented Quickly and Easily using "GrabCut"

(c) Segmentation results using Intelligent Scissors

Figure 5: Segmenting fine textures with “GrabCut”

6 Conclusion

The greyscale implementation of “GrabCut” is fast and easy to use. Intensity histograms are sufficient for modeling region data in greyscale images. It performs as well or better than the other segmentation tools built into the GIMP on greyscale images.

The EM clustering algorithm plugin is slow and the segmentation results depend heavily on the starting Gaussian parameters. When hard coded starting parameters are used, the segmentation results are comparable to the results from the histogram implementation. The method of only marking background pixels provides the algorithm with less information to work with and so increases the computational time.

The colour histogram implementation retains the speed of the black and white version, and the results are still good, even though some information is lost in the modelling of regions (see section 3.2.3). The method of drawing onto the image with a background or foreground brush provides more information to the algorithm and is more intuitive to users in cases where the foreground region is not rectangular.

6.1 Contributions

A greyscale and two colour versions of “GrabCut” have been implemented for the GIMP. The GIMP requires a fast, easy to use segmentation technique, and this is provided to the open source community in the form of the “GrabCut” plugin.

A novel approach of extending the method of representing region information with histograms in greyscale images to colour images has been implemented by creating a histogram for each colour channel. This method produces good results.

7 Future Work

Further work needs to be conducted on the EM clustering algorithm to ensure that it is implemented correctly. [Rother et al., 2004] make use of a matting technique on the edges of the segmented images. This is useful for regions with very detailed textures. Fine textures arise in images containing smoke, clouds, hair, leaves etc. The authors of the “GrabCut” paper suggest a matting brush that is used to indicate which regions on the image need matting. In the future an implementation of the matting brush will be included in the “GrabCut” plugin.

References

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision (ICCV)*.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

INCORP, A. S. (2002). *Adobe Photoshop User Guide*.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kung, S. Y., Mak, M. W., and Lin, S. H. (2004). *Biometric Authentification: A Machine Learning Approach*. Prentice Hall PTR, Unknown.

Mortensen, E. N. and Barrett, W. A. (1995). Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM Press.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314.