Image Segmentation Using Minimal Graph Cuts

Anders P. Eriksson, Olof Barr and Kalle Åström
Centre for Mathematical Sciences,
Lund University, SWEDEN
anderspe, barr, kalle@maths.lth.se

Abstract

We present a novel technique for simultaneous segmentation and classification of image partitions using graph cuts. By combining existing image segmentation approaches with simple learning techniques we manage to include prior knowledge into this visual grouping process. This has resulted in an method that partitions images into two parts based on previously seen example segmentations. Preliminary results are also presented in support of our suggested approach.

1 Introduction

Image segmentation can be defined as the task of distinguishing objects from background in unseen images. Typically this division is based on low-level cues such as intensity, homogenity or contours. Four popular approaches based on such cues are threshold techniques, edge-based methods, region-based techniques and connectivity- preserving relaxation methods. Regardless of the approach, the difficulty lies in formulating and including prior knowledge into the segmentation process. How does one describe ones perception of what constitutes foreground in an arbitrary image through low level cues? As distinguishing between foreground and background becomes harder and requires a higher level of scene understanding this task becomes increasingly difficult.

In this work we attempt to address this issue. By combining existing image segmentation approaches with simple learning techniques we seek to include prior knowledge into this visual grouping process. We wish to partition images into two parts based on previously seen example segmentations. The approach taken here is based on graph cut techniques. This was motivated by the simple fact that it has been one of the more successful approaches in image segmentation. In addition, as it will be seen, it also allowed for a straightforward incorporation of prior knowledge into its formulation. A suggestion for an efficient implementation along with some preliminary results on two different types of images are also given.

In terms of computer vision subfields, the proposed technique could be seen as being placed somewhere between segmentation, classification and detection.

2 Theory of Graph Cuts

A graph cut is the process of partitioning a directed or undirected graph into disjoint sets. The concept of optimality of such cuts is usually introduced by associating an energy to each cut. Problems of this kind have been well studied within the field of graph theory but can for graphs with more than only a few nodes be notoriously difficult. Nevertheless, ever since it became apparent that many low-level vision problems can be posed as finding energy minimizing cuts in graphs these techniques have received a lot of attention in the computer vision community. Graph cut methods have been successfully applied to stereo, image restoration, texture synthesis and image segmentation. Below we give a brief overview of graph cuts for image segmentation as well as an introduction to some basic definitions.

2.1 Min-cut/Max-flow cuts

Given a graph $G = \{V, E, W\}$, where V denotes its nodes, E its edges and W the affinity matrix, which associates a weight to each edge in E. A cut on a graph is a partition of V into two subsets A and B such that

$$A \cup B = V, \quad A \cap B = \emptyset$$

Perhaps the simplest and best known graph cut method is the min-cut formulation. The min-cut of a graph is the cut that partitions G into disjoints segments such that the sum of the weights associated with edges between the different segments are minimized. That is, the partition that minimizes

$$C_{min}(A,B) = \sum_{u \in A, v \in B} W_{uv}. \tag{1}$$

However, as this is an NP-hard combinatorial optimization problem, the task of finding the solution can be a formidable one. In order to overcome this one can relax (1) into a semi-definite program[4], resulting in a convex problem for which efficient solvers exist. However, the task of finding the solution to the original problem from the relaxed one still remains an open issue. Another commonly used approach is based on a slight reformulation of the original min-cut problem. By adding the requirement that two predefined nodes, denoted terminal nodes or source and sink nodes, in G must be in separate sets, the complexity of the problem is significantly reduced. Finding the min-cut separating the source and the sink, the s-t cut, can be achieved in polynomial time [1]. If one views the weights associated to each node as a flow capacity it can be shown that the maximal amount of a flow from source to sink is equal to the capacity of a minimal cut. Therefore the min-cut problem is also known as the max-flow problem.

2.2 The Image Seen as a Graph

The general approach to constructing an undirected graph from an image is shown if fig. 1.

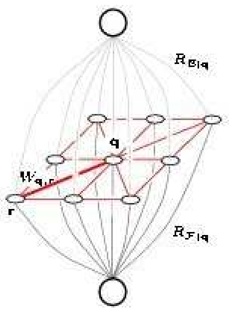

Figure 1: Graph representing a 3-by-3 image.

Basically each pixel in the image is viewed as a node in a graph, edges are formed between nodes with weights corresponding to how alike two pixels are, given some measure of similarity, as well as the distance between them. In attempt to reduce the number of edges in the graph only pixels within a smaller, predetermined neighborhood $\mathcal{N}$ of each other are considered. The two terminal nodes, the source and the sink does not correspond to any pixel in the image but instead are viewed as representing the object and background respectively. Edges are formed between the source and sink and all other non-terminal nodes, where the corresponding weights are determined using models for the object and background.

The min-cut of the resulting graph will then be the segmentation of the image at hand. This segmentation should then be a partition such that, owing to the definition of image-pixel resemblance, similar pixels close to each other will belong to the same partition. In addition, as a result of the terminal weights, pixels should also be segmented in such a manner so they end up in the same partition as the terminal node corresponding to the model (object or background) they are most similar to. An illustration of the segmentation process can be seen in figure 2.

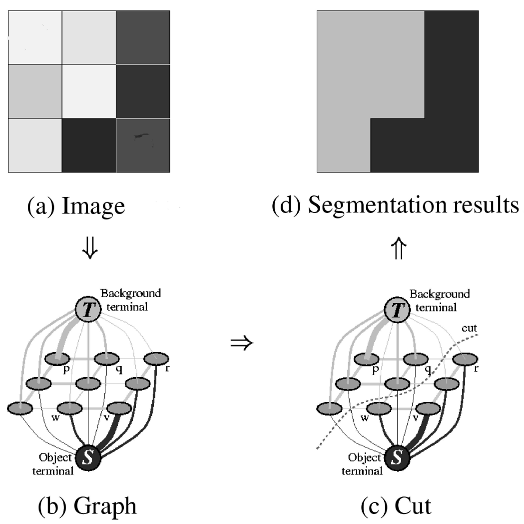

Figure 2: Example segmentation of a very simple 3-by-3 image. Edge thickness corresponds to the associated edge weight. (Image courtesy of Yuri Boykov.)

The edge weight between pixel i and j will be denoted W_{ij}^{I} and the terminal weights between pixel i and the source (s) and sink (t) as W_i^s and W_i^t respectively and are given by

$$W_{ij}^{I} = e^{(-\frac{r(i,j)}{\sigma_R})} e^{(-\frac{||\mathbf{w}(i)-\mathbf{w}(j)||^2}{\sigma_W})} \tag{2}$$

$$W_i^s = \frac{p(\mathbf{w}(i)|i \in s)}{p(\mathbf{w}(i)|i \in s)+p(\mathbf{w}(i)|i \in t)} \tag{3}$$

$$W_i^t = \frac{p(\mathbf{w}(i)|i \in t)}{p(\mathbf{w}(i)|i \in s)+p(\mathbf{w}(i)|i \in t)} \tag{4}$$

Here $||\cdot||$ denotes the euclidian norm, $r(i,j)$ the distance between pixel i and j and λ, σ_R and σ_W are tuning parameters weighing the importance of the different features. Hence, W^I_{ij} contains the inter-pixel similarity, that ensures that the segmentation more coherent. W^s_i and W^s_i describes how likely a pixel is to being background and foreground respectively.

3 Image Descriptors, Pixel Models and Prior Knowledge

As mentioned in the previous section prior knowledge is incorporated into the graph cut framework through the terminal nodes. For this purpose we need a way to describe each pixel as well as model the probability of that pixel belonging to the foreground or the background.

The image descriptors used in the current implementation are based on texture and color. For texture descriptors we used the output of a bank of 30 Gabor filters. These are a type of complex valued filters that are defined by harmonic functions modulated by a Gaussian distribution. Their close relation to process in the primal visual cortex along with a number of additional desirable filter properties has made them very popular within the image processing community. The three color channels are simply appended to this 60-dimensional, real-valued vector resulting a 63-dimensional descriptor vector v for each pixel in the image I.

The probability distribution for these descriptors are modeled using a Gaussian Mixture Model (GMM).

$$p(v|\Sigma,\mu) = \sum_{i=1}^{k} \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{(-\frac{1}{2}(v-\mu_i)^T\Sigma_i^{-1}(v-\mu_i))}$$

From a number of training images, fig. 3, with hand labelled regions the GMM parameters are then fitted through Expectation Maximization, [2]. This fitting is only carried out once and can be viewed as the learning phase of our proposed method.

4 Experiments

The examples presented below are all preceeded by a training phase, one per object class, as described above. As this is the a priori information that will determine the weights of the edges of the graph that represents the image. The inter-pixel similarity is computed according to (2) and the terminal weights from (3-4) and section 3 to form the affinity matrix. The resulting graph can then be cut, or segmented in low order polynomial time by the algorithm proposed by [1].

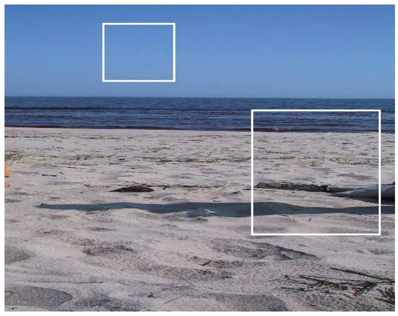

Figure 3: Example training image with two hand labelled regions. Left: sky. Right: non-sky.

We have evaluated our method on two different images, an underwater image of a coral reef and an ordinary holiday picture, and three different object categories. In the coral images the goal was to detect and segment out bleached coral and for the holiday snaps the two object categories were sky and sand. The results are shown in figs. 4 and 5.

For average size images, 320-by-320 pixels, the run-time for this method on a standard PC is approximately 2 minutes.

5 Conclusions

In this paper we have suggested a method for automatic detection, segmentation and classification of textured regions in color images. It describes how prior information can be brought into a graph cut framework through the use of terminal node weights and learning techniques. An efficient implementation is also presented along with some very promising results on an underwater image of a coral reef as well as an ordinary holiday picture.

Future work includes an more thorough examination of different object\background models. The choice of model order of the Gaussian mixture models should be made automatic. The image descriptors and the issue of scale invariance needs addressing, Finally, multiway segmentation as well as the possibilities of including prior shape information

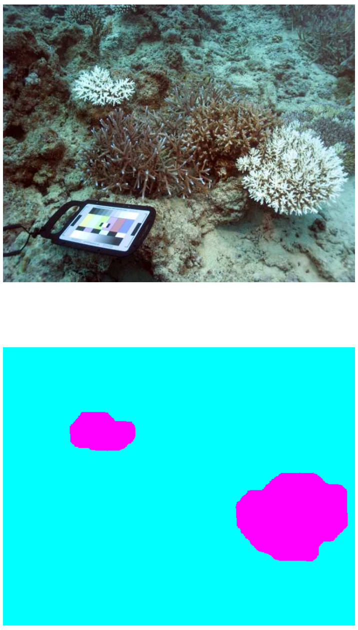

Figure 4: Segmentation of an image of a coral reef into diseased coral\background.

into the segmentation process could also prove to be promising candidates for continued research.

Acknowledgements

This work has been funded by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot.

References

[1] Y. Boykov, O. Veksler and R. Zabih "Fast Approximate Energy Minimization via Graph Cuts," *ICCV*, 377-384, 1999.

[2] A. Dempster, M. Laird and D. Rubin "Maximum likelihood from incomplete data via the EM algorithm" *J. R. Stat. Soc.*, 1977.

[3] J. Shi and J. Malik "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888-905, 2000.

[4] L. Vandenberghe and S. Boyd "Semidefinite programming," *SIAM Review*, 38(1): 49-95, March 1996.

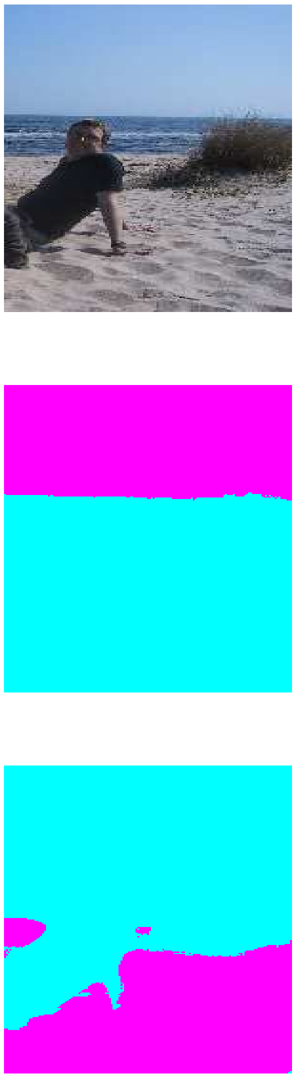

Figure 5: Segmentation of an ordinary holiday picture (top) into sky\background (middle) and sand\background (bottom).