Segmentation with Graph Cuts

Zhayida Simayijiang
Stefanie Grimm

Abstract

The aim of this project is to study graph cut methods for segmenting images and investigate how they perform in practice.

Keywords: Segmentation, Graph Cuts, Maxflow

1 Segmentation

Segmentation is an important part of image analysis. It refers to the process of partitioning an image into multiple segments. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse.

Segmentation can be used for object recognition, occlusion boundary estimation within motion or stereo systems, image compression, image editing, or image database look-up.

Segmentation by computing a minimal cut in a graph is a new and quite general approach for segmenting images. This approach guarantees global solutions, which always find best solution, and in addition these solutions are not depending on a good initialization. In our case the segmentation will be based on the image gradient with seeds provided by the user and on the mean intensity of an object.

2 Graph Theory

Graph theory is the study of graphs. A graph is an abstract representation of a set of objects, where several pairs of the objects are connected by links. It is a mathematical structure and is used to model pairwise relations between objects from a certain collection.

To give a more mathematical description of a graph, we introduce some definitions:

In a graph $\mathbf{G = (V, E)}$, V and E denote the set of vertices and edges of G, respectively. A **weighted graph** associates a positive label (weight) with every edge in the graph. A **directed graph** G consists of a set of vertices V and a set of ordered pairs of edges. An **s-t graph** is a weighted directed graph with two identified nodes, the source s and the sink t. An **s-t cut**, $c(s,t)$, in a graph G is a set of edges E_{cut} such that there is no path from the source to the sink when E_{cut} is removed from G. The **cost** of a cut E_{cut} is the sum of the edge weights in E_{cut}. The **flow** $f(u,v)$ is a mapping $f : E \to R^+, (u,v) \mapsto f(u,v)$ which fulfills the conservation of flows and the weight constraint [2]. The **value of the flow** is defined by $|f| = \sum_{v \in V} f(s,v)$, where s is the source of the graph. It represents the amount of flow passing from the source to the sink.

The maximum flow problem is to maximize $|f|$, that means to route as much flow as possible from the source to the sink. The minimum cut problem is to minimize $c(S,T)$ i.e. to find an s-t cut with minimal cost.

The max-flow min-cut theorem states: The maximum value of an s-t flow is equal to the minimum weight of an s-t cut.

Our goal will be to segment an image by constructing a graph such that the minimal cut of this graph will cut all the edges connecting the pixels of different objects with each other.

3 Segmentation with Graph Cuts

The goal is to segment the main objects out of an image using a segmentation method based on graph cuts. We used MAXFLOW - software for computing the mincut/maxflow of a graph. This software library implements the maxflow algorithm described in "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision."(Yuri Boykov and Vladimir Kolmogorov).

A graph-based approach makes use of efficient solutions of the maxflow/mincut problem between source and sink nodes in directed graphs. To take advantage of this we generate a s-t-graph as follows: The set of nodes is equal to the set of pixels in the image. Every pixel is connected with its d-neighbourhood ($d = 4, 8$). For the selection of weights, we have chosen two different models:

Segmentation based on the gradient of an image

In our first case we use the fact, that sharp edges in an image cause high gradient values. To distinguish between edges in the background and contours of our object we give some input information about the position of the object and the background. The sets $S, T \subset V$ (connected to the source and sink, respectively) denote user seeds painted in blue and red, respectively. (The choice of the colors simplifies the recognition but is rather arbitrary.) They are chosen from different regions of the image (i.e. they are disjoint sets of pixels) and have a sufficiently large size. The resulting s-t-min-cut then provides a globally minimum cost cut between the sets of pixels in S and T.

The weights of the graph are given as follows: The weight have to be the way that the minimal cut goes along the borders of the object we want to segment. Therefore the cost of a cut has to be high inside the object or the background and low at the borders of the object. To guarantee this we first smoothed the image with a Gaussian kernel to avoid sharp edges which are not belonging to the main object. Afterwards we computed the gradient of the image to detect the brinks. Based on this the weight for the edge between two pixels x_1 and x_2 becomes: $w_{1,2} = (M - \frac{||\nabla I(x_1)||+||\nabla I(x_2)||}{2})^3$, where $M := \max ||I(x_i)||$ and I is the smoothed, transformed into black and white and normalized double format representation of the input image. This choice assures that the difference between the weights is large enough to work using the maxflow function.

To segment the object, we set edges with very high weight from the source to the user seeds in the object and from the background user seeds to the sink. This ensures a flow via these edges in the maximal flow solution. Applying the MAXFLOW - software gives the minimal cut of this graph.

Segmentation based on the mean intensity

In this case we assume a characteristic mean value of the intensity for every object and the background. To simplify the computations we act on the assumption that there is only one object and we know its the mean intensity μ_1 just as the mean intensity for the background μ_0. The main goal is now to find the pixels of the object by comparing their intensity to μ_1. Therefore we want to minimize the so called Energy function:

$E(\Omega) = \lambda \cdot \text{length}(\partial\Omega) + \sum(I(x_i) - \mu_0)^2 + \sum(I(x_i) - \mu_1)^2$

The sum of the first two terms are the data term while the last term acts as a regularization.

In order to apply graph theory to this minimization problem, we set an edge from every pixel to the source and set the weights equal to $(I(x_i) - \mu_1)^2$. Likewise we connect each pixel with the sink weighted by $(I(x_i) - \mu_0)^2$. The weights between a pixel and its neighbours is equal to λ for all edges. A low λ emphasises the data term while a high λ stress the length of the object boundary. Again applying the MAXFLOW - software gives the minimal cut of this graph.

4 Examples

Segmentation based on the gradient of an image

Information about the position and the shape of the object and background are given by user seeds in terms of blue and red labels in the image.

From our experiments, we know that this method works well in most images. In some images, the result depends highly on accurate user seeds (for example if sharp edges exist in the background).

When we use 4-neighbor connectivity in the segmentation, in some images, we got sharp edges in the result. A change to 8-neighbor connectivity gives more smooth result. This is reasonable because the 8-neighbor connectivity describe the pixels in images related to their 8 neighbours. These pixels are connected horizontally, vertically, and diagonally, this means in the segmentation they have a 4 more edges to consider than 4-neighbor connectivity.

Segmentation based on the mean intensity

In our experiments, we only worked on the few images (mainly graylevel images), we know when λ is very low, the energy function based on only data term, which makes the result has more noise. When λ is very high, the energy function based on both regularization term and data term, which makes the result has few edges.

5 Conclusion

Segmentation based on graph cuts works very well for most of the images, for some issues it becomes more laborious. This means if we want to base the segmentation on the gradient of an image we need more detailed user seeds if the boundaries of the object don't differ clearly enough from the edges in the background. For the method based on the mean intensity we might need some tries to find the values for μ_1, μ_0 and λ.

Acknowledgement

The authors are grateful to Petter's supervision of the project and careful reading of the report and valuable comments.

Figure 1: Our seeds and segmentation results

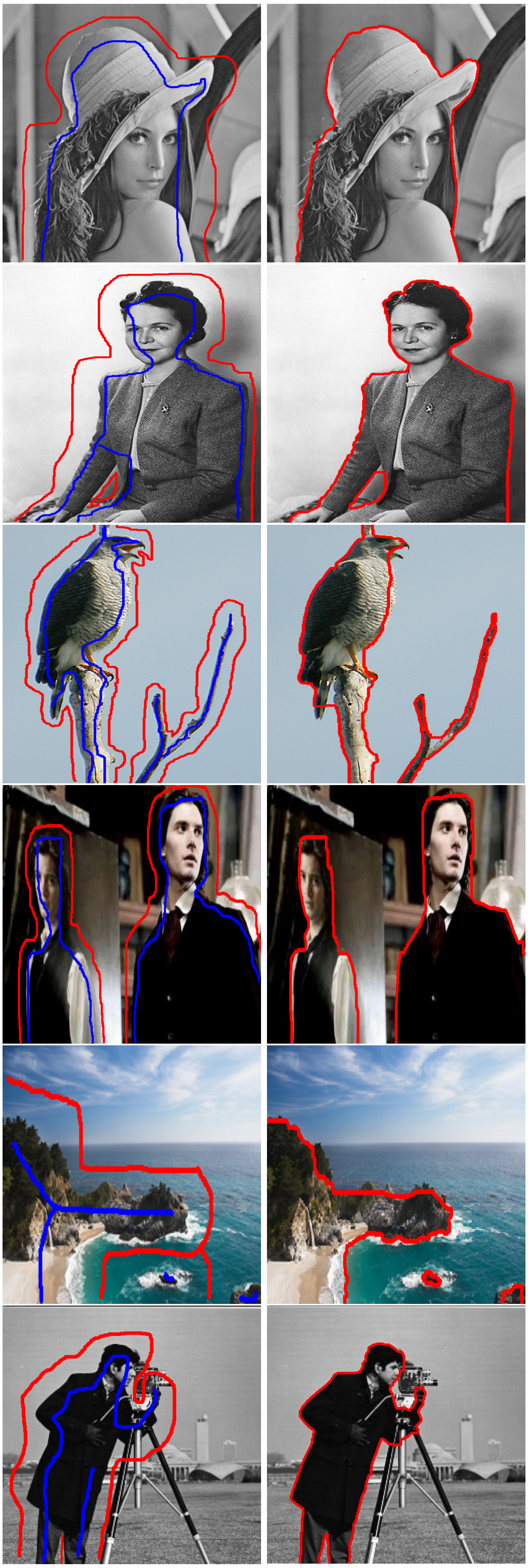

Figure 2: Compairing results for 4 and 8 neighbourhood

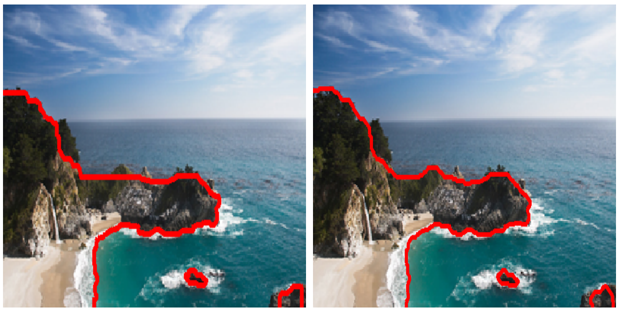

Figure 3: Result for bad user seeds

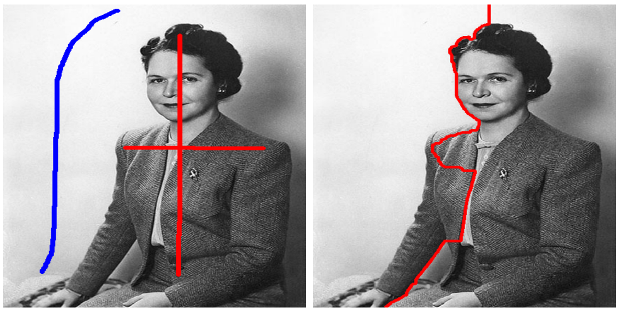

Figure 4: Different result for lamda

(a) λ=20

(b) λ=2000

(c) λ=20000

(d) λ=200000

References

[1] http://en.wikipedia.org/wiki/Segmentation_(image_processing)

[2] http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem

[3] http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software/maxflow-v3.0.src.tar.gz