An introduction to vision through image segmentation using graph cuts based on MRF

Kiran Kumar

School of Informatics and Computer science
Indian University, Bloomington
{knkumar}@indiana.edu
http://soic.indiana.edu

Abstract. This report explores image segmentation using graph cuts. The image is viewed as a MRF and a combinatorial optimization on the labels. In due process we also explore techniques that can be applied to make the problem easier, challenges faced and general representational issues. The report comes from a computer science background and builds up knowledge on prominent signal processing ideas and vision techniques.

Keywords: image segmentation, Markov random fields, graph cuts, combinatorial optimization

1 Introduction

Image segmentation is a varied combination of using visual cues and spatial coherence which tend to bind pixels which are similar in a notion of fitting together from the human visual cortex. The problem is widely studied for more than the last 20 years and only recently, we have figured out good techniques which are applicable. Image segmentation has varied applications by itself and as part of a framework for higher abstractions and/or complex vision problems. This makes the necessity for good image segmentation solutions invaluable.

Image segmentation itself has varied solutions, the number of segments, what features we are looking for in segments, these form various parameters which changes the original problem. Before we delve in to the image segmentation process and look at some possible solutions, we need to understand the visual field and describe its characteristics. This would help us understand the various subtleties and techniques which can be applied in describing visual segmentation, features and cues.

1.1 Digital Representation

The image in the visual field is represented by photons, these have certain qualities, the most important being the wave particle duality.[2] This reflective property of objects is captured digitally by means of pixels. These pixels are the building blocks of any image. The general pixel representation is RGB. In the

RGB space we represent the color space of the image with the composites of red, blue and green. Each of these are called the channels of color space. Ideally, we use 8 bits for each color, there 255 possible values for each component. We can also represent the image using 18-bits, 32-bits and 48-bits. There are also CMYK, sRGB and scRGB color space which is used in HDR.[6].

The RGB color representation is great but does not give us an intuitive idea of color. If you want orange, set $R = 255G = 142B = 13$ and darker orange $R = 210G = 65B = 0$. It becomes very hard to intuitively understand the color representation and make calls on which techniques to apply. We switch to the HSI (Hue, saturation, Intensity model) to give us a better understanding of the color distribution[7]. This is seen in color pickers across digital image applications.

With an understanding of image representation in digital medium, we can begin to apply techniques to make sense of the information present in an image.

1.2 Segmentation techniques

Some of the popular methods in segmentation include Active Contours, region based segmentation and markov random fields - graph cuts. Active contours use splines to attach to a edge or boundary and perform segmentation. These techniques are good for video tracking since once the parameters of the curve is established its easy to resegment the image. The region based techniques are divisive and agglomerative clustering methods, where splitting and merging are performed. Generalized clustering mechanisms like k-means and GMM can be applied to form the region splitting and merging. The markov random fields turns the image into a graph and computes a graphcut to define the different segmentations [1]. The problem here turns into one of energy minimization and can be handled using known techniques.

2 Markov Random Fields

Markov random fields as the name suggests uses the markov property and applies this to a field. The markov property says the current state depends only on the previous state and not on all prior states. So, if we have a random variable X, then $P(X_n = x_n|X_{n-1} = x_{n-1}...X_0 = x_0) = P(X_n = x_n|X_{n-1} = x_{n-1})$. When we consider a digital image, it is represented as a $m \times n$ grid or lattice, and a each pixel forms a state for the markov property. The previous state then just becomes the neighbouring pixel in the lattice. If we consider a pixel (i, j) in the image grid, we can consider the neighbourhood to be $(i-1, j)(i, j-1)(i+1, j)(i, j+1)$. In essence, what we are saying here is that we need each node is affected by a small circle or square around the node and the size of this boundary gets set to 1 because of the markov property as shown in figure 1.

The conditions under which a random variable X is said to be a MRF on nodes N with a neighbourhood system S are $P(x) > o, x \epsilon X$ and $P(x_i|x_{N-i}) = P(x_i|x_{Si})$[3][4].

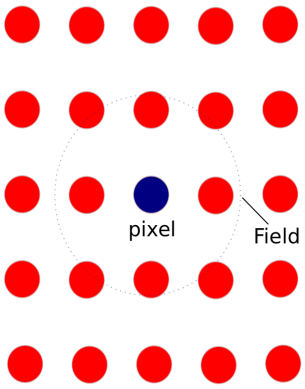


Fig. 1: The markov field neighbourhood of pixels

Using the Hammersley-Clifford theorem this MRF can be characterized by a gibbs function -

$$P(x) = \frac{1}{Z} exp(-\sum_{c \epsilon C} V_c(x))$$

where Z is

$$Z = \sum_{x \epsilon X} exp(-\sum_{c \epsilon C} V_c(x))$$

and $V_c(x)$ is the clique potential and indicates the prior probability. It can also be understood as the potential energy in the clique.

Going from here, we need to compute the interaction between the pixel and its neighbours. The potential energy can be just the intensity of the pixel, or a combination of intensity, contour and texture which describe the pixel value. Then we have a energy minimization problem at our hand, because minimum energy implies a stable configuration. The energy formulation consists of two parts, a unary potential and a binary potential. If we have a system of foreground seeds V_f, background seeds V_b and a set of pixel labels $x_i = 0, 1$ representing the background and foreground respectively, pixels represented as vertices V and interaction between them as edges E, then

$$E^{\lambda}(X) = \sum_{u \epsilon V} D_{\lambda}(x_u) + \sum_{(u,v) \epsilon E} V_{uv}(x_u, x_v))$$

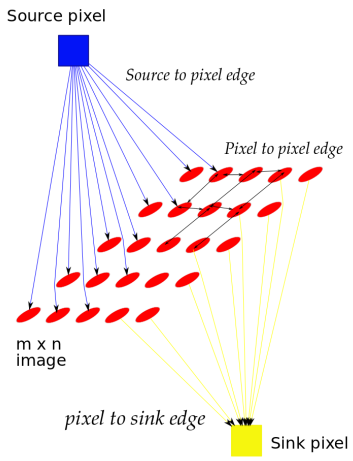


Fig. 2: The graph representation for segmentation

$$D_\lambda = \begin{cases} 0 & \text{if } x_u = 1, u \notin V_b \\ \infty & \text{if } x_u = 1, u \in V_b \\ \infty & \text{if } x_u = 0, u \in V_f \\ f(x_u) + \lambda & \text{if } x_u = 0, u \notin V_f \end{cases}$$

This helps assign a high cost for mislabeled pixels and low cost for correctly labeled pixels. The foreground gets a bias, since assigning a pixel as foreground when it does not belong to either foreground or background seed is zero. To mark a pixel as background when it does not belong to either foreground or background incurs a cost as defined. Here, $f(x_u)$ is the RGB Gaussian estimate of the pixel probability between foreground seeds and background seeds.

$$f(x_u) = lnP_f(x_u) - lnP_b(x_u)$$

To compute the prior for both the background and foreground seed, we compute the mean and covariance in the RGB space, and then compute the multivariate gaussian for both the foreground and background. Since we are taking log probability, the constant terms cancel out.

$$f(r, g, b) = \frac{1}{|\Sigma|^{1/2}} exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu))$$

The unary potential forms the edges between source-pixel and pixel-sink as foreground and background. The difficult part is to estimate the value of lambda for a given image. The idea is to run for a multitude of lambdas so as to converge

on a viable solution. The background seeds can be all the edges along the border, and the foreground seeds can be made of a regular grid array or placed by help of a k-means using the eigenvectors. In the regular grid approach, we need to split the image to a grid so that even a small segment is represented well. If we select a narrow grid, too many splits, we end up making too many computations and if we choose wide grids, too few splits, we may end up losing out on small segments. Using a 3×3 or a 5×5 grid performs well to capture the segments.

The interaction potential or the pairwise potential V_{uv} ensures spatial coherence in segmentation. It adds a penalty for assigning different labels to spatially coherent similar pixels. To compute the pairwise potential we need to represent the interaction as a function of spatial pixel similarity. Contour values make a good approximation for the spatial pixel similarity and represent the edges of pixel dissimilarity in a image very well. We can use a contour value of the image at each pixel to compute the pairwise neighbourhood function.

$$V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ exp(-\frac{max(gPb(u), gPb(v))}{\sigma^2}) & \text{if } x_u \neq x_v \end{cases}$$

Here gPb is the output of the global PB contour detector from berkeley.[5]

3 Edmond Karps

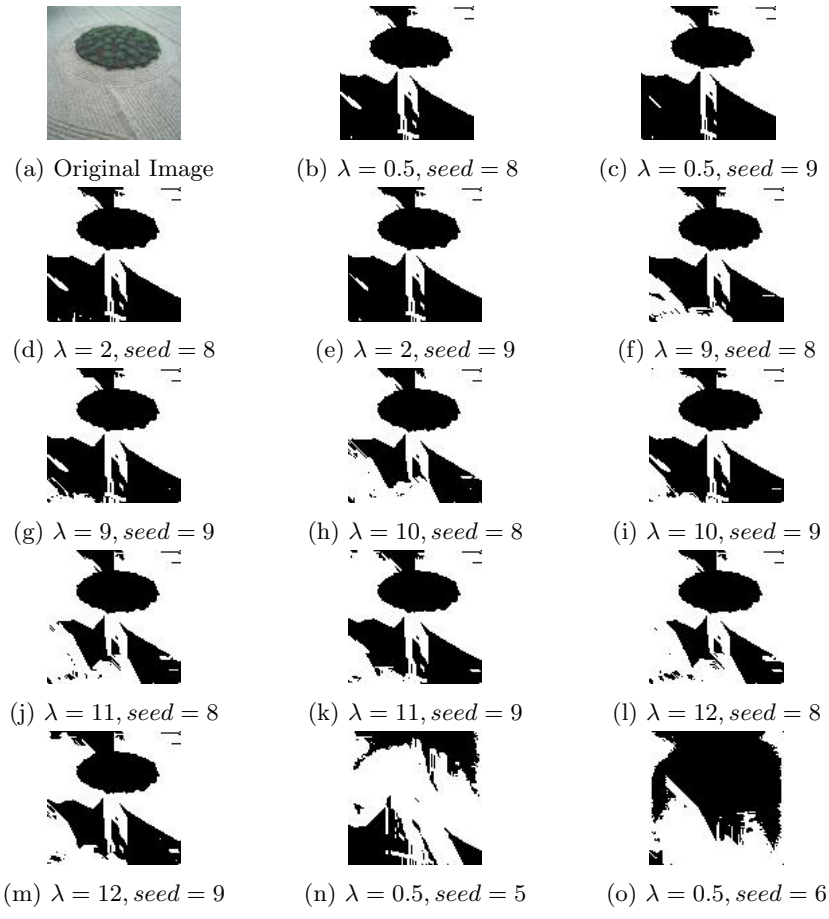
The graph representation of the energy function can be seen in figure 2. Once we have a graph representation of the image with the foreground and background nodes, we can apply combinatorial optimization techniques such as ford-fulkerson and edmond-karps to figure out the min-cut solution to the problem. It follows from the max-flow min-cut theorem, that the max-flow is equal to the min-cut in a graph. Given a graph G with capacity C between vertices V we can compute the max-flow as follows.

Initialize Initialize the flow to zero.
1 Find a path from source to sink in G, such that, $c(u, v) - f(u, v) > 0$
2 Find the minimum residual capacity r along this path and send the flow along the path $f(u, v) = r$ and return the flow $f(v, u) = -r$.
Repeat until a unvisited path satisfies the constraints

The min-cut is defined as the edges which have the flow equal to the capacity or which cannot have a residual flow.

4 Results

There are various issues with the segmentation based only upon the contour values, applying texture values would improve the results from experiments. Here there is a image with a green field at the center, surrounded by rows of fields. The image has both texture properties and segmented properties. There is a clear distinction for the foreground and background and this image forms a good test subject.



(a) Original Image (b) $\lambda = 0.5, seed = 8$ (c) $\lambda = 0.5, seed = 9$

(d) $\lambda = 2, seed = 8$ (e) $\lambda = 2, seed = 9$ (f) $\lambda = 9, seed = 8$

(g) $\lambda = 9, seed = 9$ (h) $\lambda = 10, seed = 8$ (i) $\lambda = 10, seed = 9$

(j) $\lambda = 11, seed = 8$ (k) $\lambda = 11, seed = 9$ (l) $\lambda = 12, seed = 8$

(m) $\lambda = 12, seed = 9$ (n) $\lambda = 0.5, seed = 5$ (o) $\lambda = 0.5, seed = 6$

Fig. 3: Results of image segmentation

5 Conclusion

The problem of image segmentation is hard. Finding the correct value of lambda and being able to mark the seeds automatically for a given image form the challenges which need to be answered. Placing the seeds manually is acceptable for many applications and makes the solution easier to compute. Automatic seed placement needs to make use of better techniques in terms of spatial coherence and pixel similarity areas. Lambda values make a huge difference in the quality of segments for different images. Computing a small but correct range of lambda values to explore for a set of images would help get better results.

Bibliography

[1] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241 –3248, june 2010.

[2] H. John Caulfield and Joseph Shamir. Wave particle duality considerations in optical computing. *Appl. Opt.*, 28(12):2184–2186, Jun 1989.

[3] Barry A. Cipra. An introduction to the ising model. *Am. Math. Monthly*, 94(10):937–959, December 1987.

[4] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741, 1984.

[5] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[6] Rajeev Ramanath, Wesley E. Snyder, Youngjun Yoo, and Mark S. Drew. Color image processing pipeline in digital still cameras.

[7] Alvy Ray Smith. Color gamut transform pairs. *SIGGRAPH Comput. Graph.*, 12(3):12–19, August 1978.