ES255 Final Project Report

Liam Mulshine

December 15, 2017

1 Introduction

The human visual system is extremely complex. This fact, however, has not stopped researchers in the field of computer vision from continuing to work towards their ultimate goal of designing a computer system that can match or outperform human vision. Most computer vision systems perform high level classification tasks. They are specialized to read image or video data, and infer from it useful information about the outside world. For example, some robotics applications might use a computer vision system to detect human beings, or the structure of the environment through which they are navigating. As automated systems in our society grow more prevalent, the need for robust computer vision systems similarly increases.

One particularly important sub-field of computer vision, called segmentation, is necessary for most higher level classification schemes. Segmentation addresses the challenge of identifying and separating distinct regions within images or video streams. Once segmentation is complete, higher level classification systems can infer the contents of each defined segment.

In this paper, I will discuss a form of segmentation, known as binary segmentation. Specifically, I will focus on the challenge of separating an image into two distinct regions: the "foreground" and "background". The remainder of this paper will proceed as follows. First, I will provide a more technical description of binary image segmentation, and discuss some of the research on which I will base my work. I will then formulate a solution to the binary segmentation problem for both gray-scale and color images, starting with a description of an effective probabilistic image model, and then deriving an optimal solution based on this model. Finally, I will discuss my numerical implementation and present the results that I achieved on real image data.

2 Background

The binary image segmentation problem can be constructed as follows. Suppose an image, I, contains N pixels. The image can be represented as an Nx1 vector, $I = [I_1, I_2, ..., I_N]$, composed of the N individual pixel values in I. The solution to the binary image segmentation problem is an Nx1 vector, $\omega = [\omega_1, \omega_2, ..., \omega_N]$,

that defines an optimal segmentation of image I. Each element of ω, indicates the segment - "foreground" or "background" - that the corresponding pixel I_i belongs to.

Boykov and Jolly demonstrate that the binary image segmentation problem has an optimal solution in the form of a maximum a-posteriori estimator, provided that the image is modeled as a Markov Random Field [1]. Rother, Blake and Kolmogorov extend this solution to color images, and introduce an additional iterative energy minimization scheme to improve upon the initial segmentation estimate [3]. In the sections that follow, I will discuss the formulation of the optimal MAP-MRF estimator, which is commonly used for image segmentation problems, and demonstrate its performance on a variety of images.

2.1 The Markov Random Field

The most common probabalistic image model used for image segmentation in computer vision is the Markov Random Field [1,3]. In general, the image model used must capture the contextual constraints within an image. In this sense, it must model the fact that neighboring pixels are often correlated, taking on similar values if they are part of the same segment. In the next section, I will provide some background on the Markov Random Field and discuss how it can be used as a probabilistic image model, capturing regional homogeneity of color and brightness.

At a high level, a Markov Random Field (MRF) is a set of random variables that satisfy the Markov Property. Each random variable can be represented as a node in an acyclic, undirected graph. Nodes within the graph are connected to other nodes with variable strength links, where the strength of the link indicates how strongly the two nodes are correlated. An MRF is composed of cliques, which are groupings of connected nodes. The value that a given node takes on is only dependent upon the value assumed by each node within its clique. Perhaps most importantly for our model, the Markov Random Field must satisfy the Gibbs Distribution, which characterizes the probability that a given system is in state, ω. The Gibbs distribution, $f(\omega)$, is defined as [4]

$$f(\omega) = \frac{1}{Z} \exp(-\sum_{c \in C} V_c(\omega)) \tag{1}$$

In (1), $V_c(\omega)$, is the potential energy function for clique, c, and Z is a normalizing constant. The sum of the energy of each clique within the MRF (the term in the exponential), is known as the Gibbs Energy. This term will appear in the solution for $\hat{\omega}_{MAP}$.

An image can be modeled as a Markov Random Field if defined as follows. Each pixel, i can be represented as a node within the Markov Random Field, and pixel i's four nearest neighbors (specifically, the pixels above, below and on either side of pixel i), are grouped into pixel, i's clique, c_i. The random variable, ω, is the segmentation variable, holding state about the region in which each pixel lives. Therefore, the Gibbs distribution for the Markov Random Field can

be constructed as in (1) above, characterizing the probability that an image is segmented by ω. The potential functions for each clique will be defined to promote segmentation smoothness.

3 Problem Formulation

With this probabilistic image model defined, the solution to the binary segmentation problem can be constructed. As mentioned earlier, this solution can be formulated as a maximum a-posteriori estimator of the segmentation variable, ω. Given the MRF image model, it can be shown that this maximization problem is equivalent to minimizing the Gibbs Energy of a Markov Random Field. While no closed-form solution to the optimization problem can be reasonably constructed, since ω is highly dimensional, the energy minimizing ω can be found using a minimum graph cut approach [1]. In this section I will first discuss the formulation of $\hat{\omega}_{MAP}$ for grayscale images. Then I will move onto the more complicated case of color images.

3.1 Gray-scale Image Segmentation

As mentioned, the optimal segmentation variable, ω, can be written in the form of a MAP estimator

$$\hat{\omega}_{MAP} = \underset{\omega}{\operatorname{argmax}}\ f(\omega|I) \tag{2}$$

The a-posteriori distribution above, $f(\omega|I)$, characterizes the probability that a given image, I, is segmented by ω. This optimization problem seeks to find the segmentation variable, ω, that maximizes the a-posteriori distribution. Applying Bayes rule, the argument in (2) can be expanded

$$f(\omega|I) = \frac{f(\omega)f(I|\omega)}{f(I)} \tag{3}$$

From (2) and (3) it follows that

$$\hat{\omega}_{MAP} = \underset{\omega}{\operatorname{argmax}}\ f(\omega)f(I|\omega) \tag{4}$$

where the distribution, $f(I)$ is omitted since it is constant with respect to ω. Therefore, the MAP estimator is a maximization over the product of two distributions that depend on ω. The first, $f(\omega)$, is the Gibbs distribution from our MRF image model, characterizing the prior probability of seeing the segmentation vector, ω. The second term is the likelihood of image I, given its segmentation. To fully define this maximization problem, the likelihood and prior terms must be defined explicitly.

The likelihood term, $f(I|\omega)$, also called the Data term, can be written as a product of the individual likelihood probabilities

$$f(I|\omega) = f(I|\omega) = \prod_{i}^{N} f(I_i|w_i) \tag{5}$$

where it is assumed that, conditioned on ω_i, pixel I_i is independent of its neighbors. The individual likelihood characterizes the probability that pixel I_i lives in segment, ω_i. Since ω_i takes on one of two values from the set, $\{0, 1\}$, two distributions must be defined, which describe the spread of grayscale intensities within the "background" and "foreground" regions respectively. In the simplest case, a univariate Gaussian is used

$$f(I_i|w_i) = \frac{1}{\sqrt{2\pi}\sigma_{w_i}} \exp\big(-\frac{(I_i - \mu_{w_i})^2}{2\sigma_{w_i}^2}\big) \tag{6}$$

where the mean, μ_{ω_i}, and the variance, $\sigma_{\omega_i}^2$, are defined as the sample mean and sample variance within regions that exhibit properties of the "foreground" and "background" segments, defined a-priori.

The Gibbs prior, $f(\omega)$, has the same form as (1), where the clique potentials are defined in terms of the "Ising Prior" [1]

$$V_c(\omega, I) = \gamma \sum_{(i,j)\in c} [\omega_i \neq \omega_j] \exp(-\beta(I_i - I_j)^2) \tag{7}$$

As its form suggests, the "Ising prior" promotes smoothness within each clique in the MRF, and, consequently, is referred to as the Smoothness term. In (7), the function $[\omega_i \neq \omega_j]$ is an indicator function, assuming the value 1 whenever neighboring pixels i and j within a given clique are unequal, and 0 otherwise. This effectively assigns a cost to inhomogeneous cliques, penalizing segmentations that promote inhomogeneity. The constant, β, is chosen to normalize the squared pixel difference such that a high cost is assigned only to cliques whose pixel values are similar in intensity but exhibit inhomogeneous segmentation. The constant, γ, directly affects this cost's weight. The exact value of both constants varies depending upon the images used.

Combining the Smoothness and Data terms the argument that is maximized can be fully defined

$$f(I|\omega)f(\omega) = \prod_{i}^{N} f(I_i|\omega_i) \frac{1}{Z} \exp\big(-\sum_{c\in C} V_c(\omega, I)\big) \tag{8}$$

This can be rewritten in the form of the Gibbs distribution (1),

$$f(I|\omega)f(\omega) = \frac{1}{Z} \exp(-U(\omega, I) \tag{9}$$

where the new Gibbs energy, $U(\omega, I)$, is defined in terms of the individual log-likelihoods and the Ising prior

$$U(w) = -\sum_{i=1}^{N} \log(f(I_i|\omega_i)) + \sum_{c \in C} V_c(w, I) \tag{10}$$

Rewriting $\hat{\omega}_{MAP}$ in terms of the Gibbs distribution (9), it is clear that the maximum a-posteriori estimator minimizes the Gibbs energy for a given image, I.

$$\begin{aligned} \hat{\omega}_{MAP} &= \underset{\omega}{\operatorname{argmax}}\ f(I|\omega) f(\omega) \\ &= \underset{\omega}{\operatorname{argmax}}\ \frac{1}{Z} \exp(-U(w)) \\ &= \underset{\omega}{\operatorname{argmin}}\ U(w) \end{aligned} \tag{11}$$

Unfortunately, a closed-form solution to this minimization is not achievable given ω's high dimensionality. Fortunately, a solution in the form of a minimum graph cut exists and can be solved numerically with reasonable efficiency [1]. I will discuss this numerical implementation in section 4.

3.2 Color Image Segmentation

A similar formulation for $\hat{\omega}_{MAP}$ can be carried out on color images. However, since the color image contains three covariant channels per pixel, new data and smoothness terms must be constructed to capture the additional complexity.

The Markov Random Field can again be used as the general probabilistic image model. Therefore, the solution to the segmentation problem has a form similar to (11). The prior (Smoothness term), $f(\omega)$, can still be described by the Gibbs distribution, (1). However the clique potentials are redefined to account for the fact that pixels are 3-Dimensional.

$$V_{(i,j)}(\omega, I) = \gamma \sum_{(i,j) \in C} [\omega_i \neq \omega_j] \exp(-\beta ||I_i - I_j||^2) \tag{12}$$

The likelihood (Data term), on the other hand, must be fully redefined. In order to capture all three pixel dimensions and more complex foreground and background distributions, a multivariate Gaussian Mixture Model is employed. Remember that, when performing binary image segmentation, the foreground object and background scene might contain a range of unique colors. It would be ineffective to generalize and claim that every unique color in either region can be described by a single Gaussian. Instead, the Gaussian Mixture Model describes each region with multiple Gaussian components, each of which having some associated prior. In the color image, each pixel is assigned to the component by which it is best supported. In this sense, the multivariate GMM captures both the covariance between pixel channels and the added complexity of multimodal foreground and background color distributions.

The form of each GMM component, k_i for segment ω_i with Gaussian parameters $\hat{\theta}$ is defined as,

$$f(I_i|\omega_i, k_i, \theta_{k_i}) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_{k_i})}} \exp(-\frac{1}{2}[I_i - \mu(\omega_i, k_i)]^T \Sigma_{k_i} [I_i - \mu(\omega_i, k_i)]) \tag{13}$$

where the mean $\mu_i(\omega_i, k_i)$ and covariance Σ_{k_i} are the sample mean and covariance of each pixel assigned to component k_i of the GMM.

As in the construction of (8), the smoothness and data terms can be combined, resulting in a new Gibbs distribution for color images

$$\begin{aligned} f(I|\omega, k, \hat{\theta}) f(w) &= \prod_i^N f(I_i|\omega_i, k_i, \hat{\theta}) f(\omega_i, k_i) \exp\big(-\sum_{c \in C} V_c(\omega, I)\big) \\ &= \frac{1}{Z} \exp(-U(\omega, I, \hat{\theta}, \hat{k})) \end{aligned} \tag{14}$$

where the Gibbs energy, $U(\omega, I, \hat{\theta}, \hat{k})$, is defined in terms of the individual log-likelihoods and clique energies

$$U(\omega, I, \hat{\theta}, \hat{k}) = \sum_i^N \big[-\log(f(I_i|\omega_i, k_i, \theta_{k_i}) - \log(f(\omega_i, k_i))\big] + \sum_{c \in C} V_c(\omega, I) \tag{15}$$

Notice the additional log likelihood term, $f(\omega_i, k_i)$. This defines the prior probability of seeing Gaussian mixture model component, k_i, within segment ω_i.

As in the gray-scale case, the optimal estimator, $\hat{\omega}_{MAP}$, can be found by minimizing the Gibbs energy (15) of the newly defined Markov Random Field

$$\begin{aligned} \hat{\omega} &= \underset{\omega}{\operatorname{argmax}} f(I|\omega, \hat{\theta}, \hat{k}) f(\omega) \\ &= \underset{\omega}{\operatorname{argmax}} \frac{1}{Z} \exp(-U(\omega, I, \hat{\theta}, \hat{k})) \\ &= \underset{\omega}{\operatorname{argmin}}\ U(\omega, I, \hat{\theta}, \hat{k}) \end{aligned}$$

In the next section I will discuss a numerical solution to this optimization problem.

4 Numerical Implementation

The problem formulation for binary segmentation on both color and gray-scale images has led to an optimal solution in the form of an energy minimization. Unfortunately, since ω has high dimensionality, an analytical solution cannot be easily found. Therefore, we turn to an approach that is well suited for energy minimization within an MRF: the minimum graph cut [1, 3].

From a high level, the minimum graph cut finds the optimal segmentation through a field of interconnected nodes, each with variable strength links. In the image model that has been constructed, pixels are represented as nodes within the graph. Each pixel is linked with the 4 pixels within its clique, and the strength of each inter-pixel connection is defined by the Ising prior, which was discussed in section 3. In addition, every node is connected to two terminal nodes, representing the image foreground and background respectively. The strength of these terminal connections is defined by the likelihood of the given pixel under the appropriate terminal's distribution. Below is a visualization of the form of this graph for a 3x3 image [1]

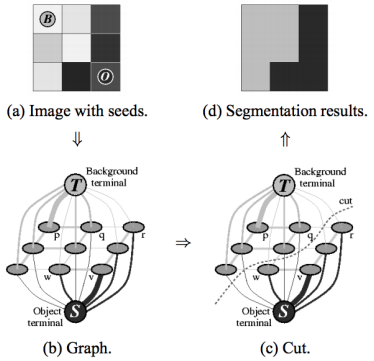

Figure 1: Graph-cut visualization

Implementation of the minimum graph cut is discussed in Boykov's "Fast Approximate Energy Minimization via Graph Cuts" [2]. Implementing this minimum cut scheme was not my project's primary focus, so I utilized a MATLAB wrapper [6] which incorporates Kolomogorov's C++ maxflow/minimum cut algorithm [5] in my numerical implementation.

For the gray-scale image, a one-shot solution is employed. The implementation begins with initialization of the foreground and background distributions. Then, two sparse matrices are defined that specify the weights of each inter-pixel connection (based on the Ising prior) and each terminal connection (based on the log likelihood). The minimum cut/maxflow algorithm [5, 6] is used to find the optimal segmentation, $\hat{\omega}_{MAP}$.

The optimal segmentation for the color image is determined in a similar way. However, as in "GrabCut" [3], an iterative energy minimization is employed. Upon initialization, a GMM is constructed for both the foreground and background regions (the user defines a segment of the image that characterizes the background; the rest of the image is initially accepted as the foreground). After initialization, the method iterates through the following three steps until energy convergence. First, pixels are assigned to the most likely GMM component for the GMM that they are associated with. Based on this componentwise assignment, new GMM parameters are learned. Finally, an graph is constructed

using the data and smoothness terms defined in section 3.2, and an energy minimizing segmentation is determined via the minimum graph cut [6].

5 Results and Analysis

I tested my implementation on both gray-scale and color images. Gray-scale testing was fairly simple, and served mainly as a "proof-of-concept" for the graph cut approach that I've discussed. My testing on color images was more extensive. In this section, I will briefly discuss the results that I achieved on gray-scale images. Then I will move onto an analysis of the results achieved for multiple color images, demonstrating the effectiveness of iterative energy minimization.

5.1 Gray-scale Image Segmentation Results

I tested my implementation on the gray-scale image shown on the left below. Two separate regions within the image were identified and used to initialize the foreground and background univariate Gaussian distributions. Results are demonstrated in the black and white mask image on the left.

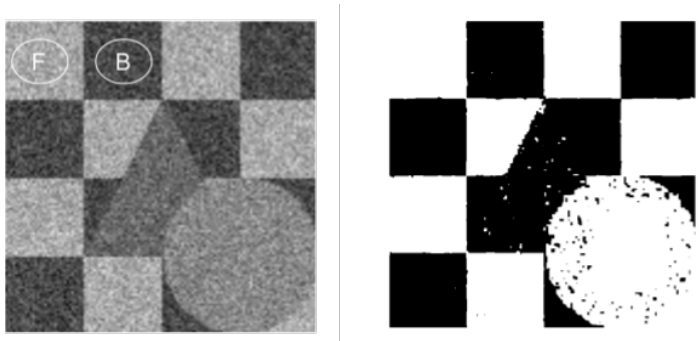

Figure 2: Gray-scale test image (left) and binary segmentation result (right)

Notice the strong performance in regions whose intensity distribution closely matches that of either the initialized foreground and background regions. The triangular and circular regions are not segmented as smoothly, however, because of the simple univariate Gaussian model's failure to capture multimodal intensity distributions. The GMMs employed for color image segmentation are more effective at capturing this heightened complexity.

5.2 Color Image Segmentation Results

The segmentation results achieved on color images using an iterative energy minimization scheme and multivariate GMM color distributions were, in general,

quite strong. I first tested my implementation on the test image employed in "GrabCut" [3], of a man in front of a leafy background. The results are provided below.

Figure 3: Color test image for iterative binary segmentation scheme (left) and experimental result (right)

I then tested my implementation on a variety of additional images, and have provided the results from two experiments.

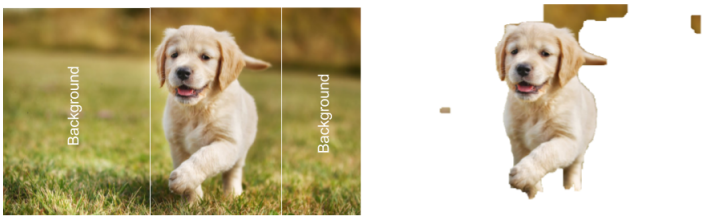

Figure 4: Color test image for iterative binary segmentation scheme (left) and experimental result (right)

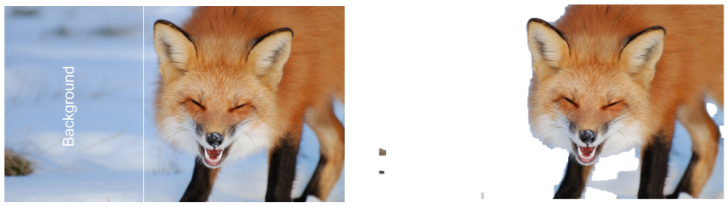

Figure 5: Color test image for iterative binary segmentation scheme (left) and experimental result (right)

Note that the "background" region in each test was crudely identified upon in order to initialize the background Gaussian mixture model. The remaining area in each image was used to initialize the foreground GMMs. Clearly, despite far

from perfect initial estimates of the foreground and background distributions, the binary segmentation scheme performed quite well.

The first experiment on the image used in "GrabCut," [3] was particularly effective. Although there are a few clear imperfections, the result is comparable to those achieved by the original implementation prior to manual retouching.

The second experiment, using an image of a walking puppy, demonstrates strong segmentation performance despite similar foreground and background distributions. The algorithm has some difficulty separating the region above to dog's head. However, again, this imperfection is comparable to those seen in the original implementation, and could be fixed with some simple manual retouching.

The third test on the fox in the snowy environment demonstrates a limitation of the approach used. The algorithm hesitates to assign the regions between the fox's legs to the background, since these regions are isolated from the rest of the image background. Assigning them to the background segment would incur a smoothness cost, which appears to outweigh the cost incurred by cutting the connection to the background terminal node.

Below I've included a plot of the total Gibbs energy over 10 iterations of the iterative segmentation scheme using the fox image from the third experiment. Notice that, in general, energy decreases with each iteration, due to improved Gaussian mixture model estimates.

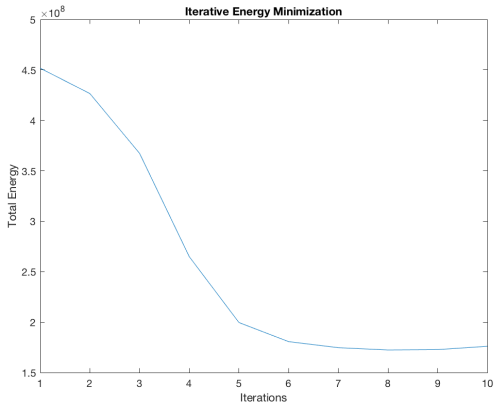

Figure 6: Iterative Energy Minimization

While imperfect, the results achieved with this implementation of iterative graph-cut image segmentation were acceptable. Provided more time, I would address some of the issues identified in the results above. I would also extend this solution to the more general problem of M-ary segmentation, in which more than 2 segments need to be identified. Lastly, I would work to improve the algorithm's computational efficiency so that it could be incorporated into real-time dynamic systems with latency constraints.

6 Conclusion

The graph cut method for binary image segmentation, which I've discussed in detail, effectively identifies the foreground and background regions within an image starting from a crude initialization. As discussed, this method is based on the solution to a maximum a-posteriori estimation problem. Although a closed form solution to this problem was not provided, a strong approximation is shown using a minimum graph cut algorithm. The results that I've presented on real image data effectively demonstrate this method's strengths, while also indicating room for future improvement.

References

[1] Boykov, Y., and Jolly, M.-P. "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images"

[2] Boykov, Y. "Fast Approximate Energy Minimization via Graph Cuts." Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.

[3] Rother, C., Blake, A., and Kolmogorov, V. "'GrabCut'- Interactive Foreground Extraction using Iterated Graph Cuts"

[4] Geman, Donald. Geman, Stuart. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images."

[5] Kolomogorov, V. "MAXFLOW." Computer Vision Software, pub.ist.ac.at/ vnk/software.html.

[6] Rubinstein, M. MATLAB Central - File Exchange, 16 Sept. 2008, www.mathworks.com/matlabcentral/fileexchange/21310-maxflow.