CMSC 733
Homework 0

Rohith Jayarajan
A. James Clark School of Engineering
University of Maryland
College Park, Maryland
Email: rohith23@terpmail.umd.edu

***Abstract*—Boundary or contour detection is an important problem in computer vision and is well studied. Other important problems like hierarchical image segmentation can be reduced to a contour detection problem. A version of the state-of-the-art edge detection algorithm Pb-Lite is discussed in this project. The Pb Lite approach causes a jump in the performance of boundary detection by suppressing false positives in the textured regions.**

I. Introduction

Edge detection is a well studied problem in computer vision and image processing as it is important in applications where we want to recognize patterns in the image. One of the early methods to approach this problem is to find discontinuities in the intensity across the image; which are called edges. Since then there have been classical edge detection algorithms developed like the Sobel operator; a discrete differentiation operator which when applied on the image gives the gradient intensity function of that image, Canny edge detector; a multistage algorithm involving non-maximum suppression and tracks edges by hysteresis, and MarrHildreth algorithm among many others. But the recent Pb (Probability of Boundary) edge detection algorithm outperforms all the existing contour detection algorithms by using the information of texture and color discontinuities along with intensity discontinuities.

In this work, a simplified version of the Pb (Probability of Boundary) edge detection algorithm called the Pb-Lite algorithm is implemented. Although, a simplified version of the algorithm, it still outperforms the Canny and Sobel edge detectors.

A. Phase 1: Pb-Lite Contour Detection

An overview of the Pb-Lite algorithm is described in the figure 1. The various components of the algorithm pipeline is discussed in the following sections

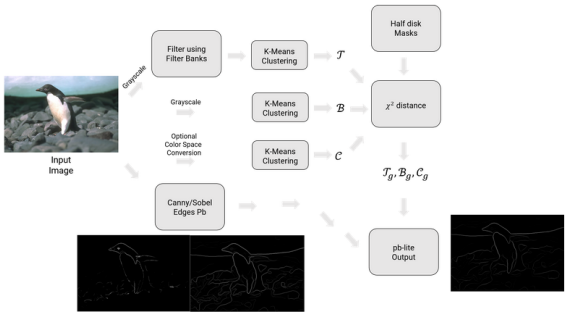

Fig. 1. Pb-Lite overview

1) Filter Banks: The first step in the Pb-Lite Contour Detection is to create a set of filters with which the image of interest is filtered to produce useful responses. A filter bank is defined as a collection of filters. There are three types of filter banks used for this process:

a) Oriented DoG Filters: The collection of oriented Derivative of Gaussian (DoG) filters is a simple yet powerful collection of filters. These can be generated by convolving a Gaussian kernel with a Sobel filter. We generate oriented DoG filters with o orientations and s scales where scale is given by the standard deviation σ of the Gaussian kernel to give oxs filters in this filter bank. A set of 32 filter banks with 2 scales and 16 orientations were generated in this work and can be seen in figure 2

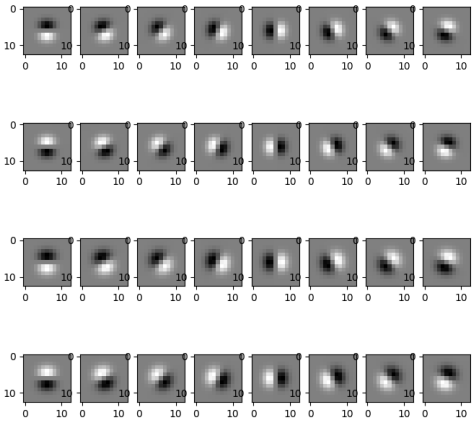

Fig. 2. Oriented DoG filter bank

b) Leung-Malik Filters: Another set of filters are the multi scale, multi orientation filter bank consisting of the Leung-Malik filters or LM filters. LM filter bank consists of first and second order derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian (LOG) filters; and 4 Gaussians. There are two variants to the LM filters. The LM Small (LMS) filter bank which occur at the basic scales of $\sigma = 1, \sqrt{2}, 2, 2\sqrt{2}$. The first and second derivative filters occur at the first three scales with an elongation factor of 3. The Gaussians occur at the four basic scales while the 8 LOG filters occur at σ and 3σ. The LM Large (LML) filters occur at the basic scales $\sigma = \sqrt{2}, 2, 2\sqrt{2}, 4$. The LM filter bank generated is shown in figure 3.

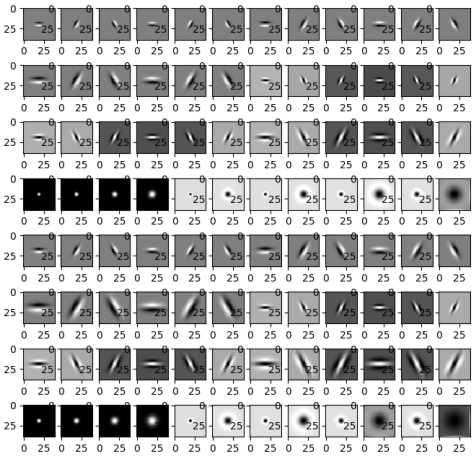

Fig. 3. Leung-Malik filter bank

c) Gabor Filters: Filters designed derived from the ones present in the human visual system, the Gabor filter bank is a Gaussian kernel function modulated by a sinusoidal plane wave. The LM filter bank generated is shown in figure 4.

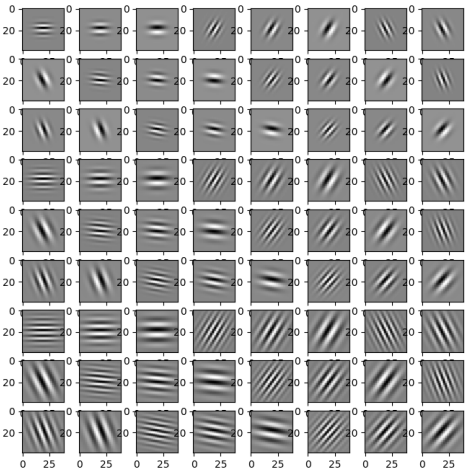

Fig. 4. Gabor Filters filter bank

2) Texton Map $\mathcal{T}$: To generate the Texton Map $\mathcal{T}$, the input image is filtered with each filter of the filter bank. This gives **N** filter responses at each pixel of the image. Hence we obtain an **N** dimensional vector on which we apply a K-Means clustering with the number of clusters **K = 64**. The Texton map of the image shown in figure 5 can be visualized in 6. All the successive operations will be performed on image in figure 5 to find its contours.

Fig. 5. Image of interest

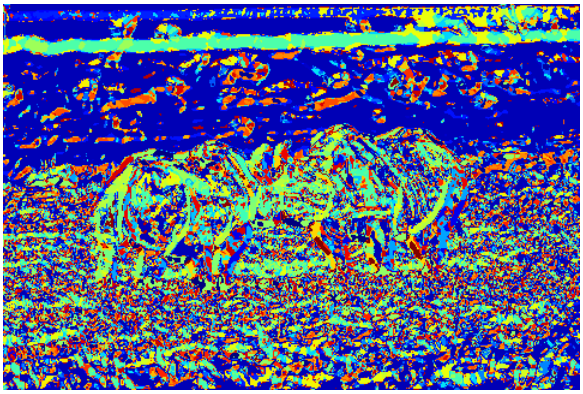

Fig. 6. Texton map

3) Brightness Map $\mathcal{B}$: The Brightness Map $\mathcal{B}$ is generated by performing a K-Means clustering on the grayscale equivalent of the color image with the cluster size **K = 16**

The brightness map of the image in figure 5 is shown in figure 7.

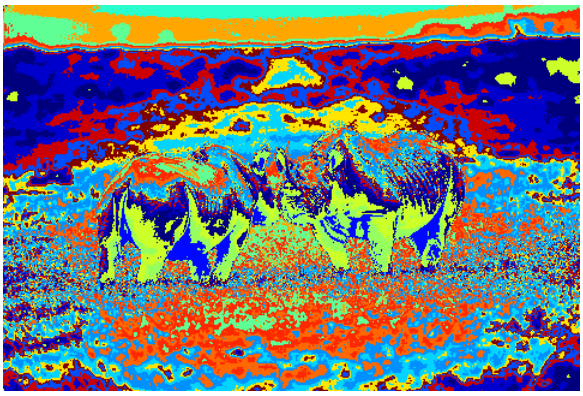

Fig. 7. Brightness map

4) Color Map $\mathcal{C}$: Color Map $\mathcal{C}$ captures the changes in the chrominance in the image. It is generated by performing a K-Means clustering on the color image with the cluster size **K = 16**

The Color map of the image in figure 5 is shown in figure 8.

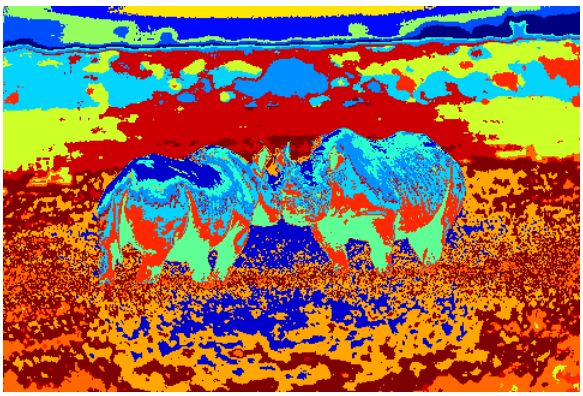

Fig. 8. Color map

a) Half Disc Masks: Half disks masks help compute differences of values across different shapes and sizes which return us $\mathcal{T}_g$, $\mathcal{B}_g$, and $\mathcal{C}_g$. These are pairs of binary images of half discs at different scales which can be seen in figure 9

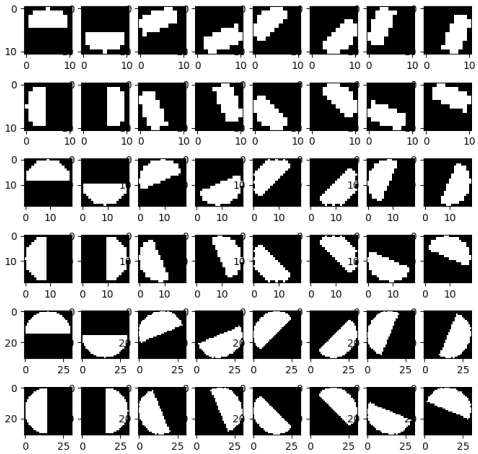

Fig. 9. Half disc masks at different scales and sizes

5) Texture, Brightness and Color Gradients $\mathcal{T}_g$, $\mathcal{B}_g$, $\mathcal{C}_g$: The Texture, Brightness and Color Gradients $\mathcal{T}_g$, $\mathcal{B}_g$, $\mathcal{C}_g$ encode how much the texture, brightness and color distributions are changing at a pixel. These gradients are found out by comparing the distributions in left and right of the half disc pairs which gives us the χ^2 measure. The χ^2 is simply the sum of squared difference between histogram elements along with a soft normalization so that less frequent elements still contribute to the overall distance.

This procedure generates a 3D matrix of $mxnxN$ where N is the total number of filters.

Figure 10, 11 and 12 show the texture, brightness and color gradients of image in figure 5 respectively.

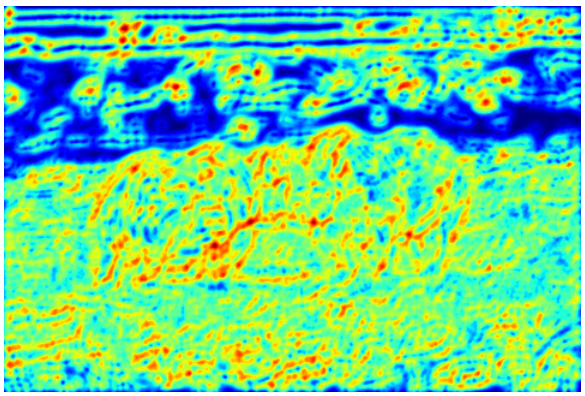

Fig. 10. Texture gradient

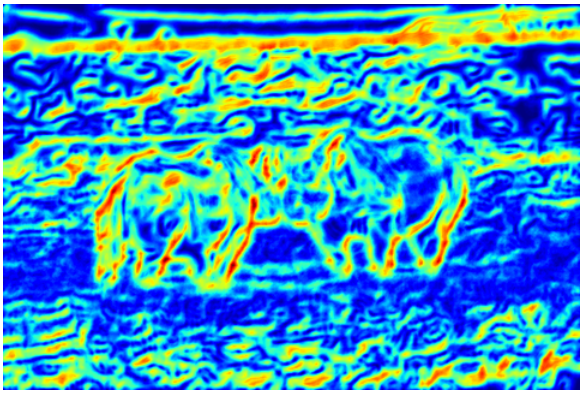

Fig. 11. Brightness gradient

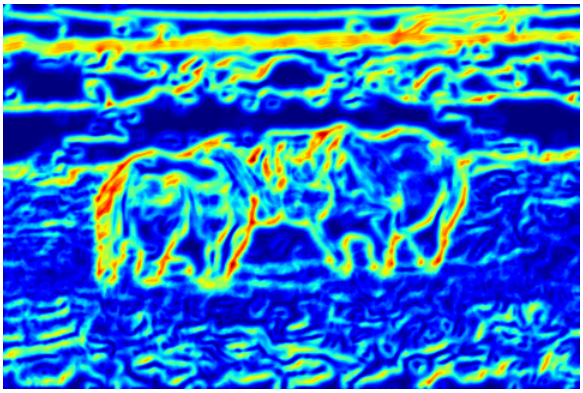

Fig. 12. Color gradient

6) Sobel and Canny Baselines: When Sobel and Canny edge detectors are applied to the image in figure 5, we get the contour results as shown in figure 13 and 14 respectively.

Fig. 13. Sobel baseline

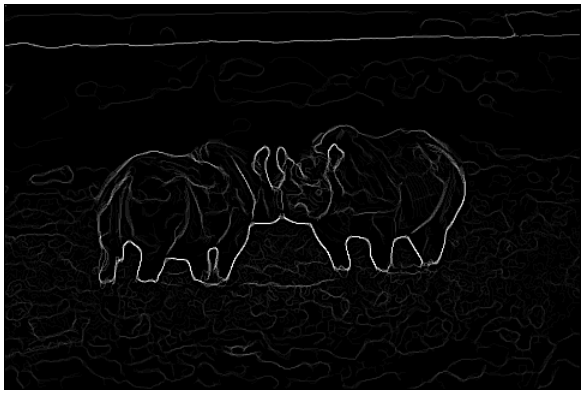

Fig. 14. Canny baseline

7) Pb-Lite Output: The last step of the pipeline is to combine information from the features (Texture, Brightness and Color Gradients $\mathcal{T}_g$, $\mathcal{B}_g$, $\mathcal{C}_g$) with the baseline outputs from the Sobel or Canny edge detection using the below equation.

$$PbEdges = \frac{1}{3}(\mathcal{T}_g + \mathcal{B}_g + \mathcal{C}_g$$

$)\odot$ (w$_1 * cannyPb + w_2 * sobelPb$)(1)

The mean over feature vector at a pixel will be somewhat proportional to Pb. The final output given by the algorithm Pb-Lite is shown in figure 15.

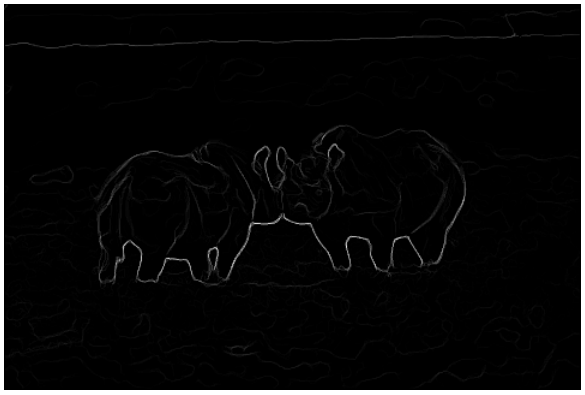

Fig. 15. Pb Lite output

II. Conclusion and Analysis

The PbLite algorithm produces contours that are significantly better than those produced by Sobel and Canny due to the following reasons:

1) Sobel and Canny both model edges as sharp discontinuities in the brightness channel and Canny adds to this by using non-maximum suppression and hysteresis thresholding steps. But in the case of PbLite, not only the brightness but the texture and color information is also taken into account making it superior.
2) The description in Sobel and Canny are not obtained by considering response of the image to filters at different scales. The use of different scales in PbLite gives it the edge.

3) Both Canny and Sobel edge detectors quantify the presence of a boundary using local measurements. PbLite couples multiscale local brightness, color and texture cues to a powerful globalization framework using spectral clustering making it far more superior than Canny and Sobel edge detectors.
4) From the wide gap in performance of PbLite with against the human level score, there must exist some features not taken into consideration by the algorithm.
5) Due to a large number of parameters involved in the PbLite Algorithm, choosing the best set of parameters for an image is a really daunting task.
6) Spectral clustering could provide better performance than the K-means clustering used in this implementation.

REFERENCES

[1] CMSC733 Course Website, *https://cmsc733.github.io/2019/hw/hw0/*,
[2] Pablo Arbelaez,Michael Maire, Charless Fowlkes, Jitendra Malik *Contour Detection and Hierarchical Image Segmentation*
[3] David Forsyth, Jean Ponce *Computer Vision: A Modern Approach*