Saliency Segmentation based on Learning and Graph Cut Refinement: A Report

Advisor

Dr. Maryam Abedi

Student

Mohammad Shahpouri

August

2022

Contents

List of Figures

List of Tables

List of Equations

1 Method

First the input image is segmented into superpixels. Then the trained classifier is applied to each superpixel individually to compute confidences of each superpixels. Based on confidences, superpixels are classified into salient/background. Finally the results are refined with graph-cut optimization.

The two main steps are the confidence map computation and the graph-cut refinement. They are discussed in sections 1.1 and 1.2, respectively.

1.1 Saliency Map

Since single pixel doesn't have enough reliable features to classify salient object from background, superpixels are applied. The applied features are color, size, location, and texture of superpixels.

Color: Mean of RGB and HSV values, hue histogram, and saturation histogram are the extracted features from color.

Location and Size: The mean of normalized x and y coordinates for pixels in a superpixel and also the normalized area of a superpixel are computed as location and size features.

Texture: Mean and standard deviation of the absolute responses, histogram of maximum responses, and the maximum responses, the most frequent texton [1] channel, texton histogram, and the number of different texton channels in each superpixel were extracted.

They normalize the features then utilize a boosting algorithm with small decision trees [2] for training. At the testing step, confidence is computed for each superpixel. Saliency map is formed by assigning the confidence of the superpixel to all pixels in that superpixel.

1.2 Refinement with Graph-Cuts

Binary graph cut optimization [3] for refinement are utilize to enhance the boundary and classifier results. The problem is formulated as binary salient object or background segmentation. Their graph cut energy function is the standard form introduced in the framework

$$E(f) = \sum_{p \in \mathscr{P}} D_p(f_p) + \lambda \sum_{\{p,q\} \in \mathscr{N}} V_{p,q}(f_p, f_q) \tag{1}$$

Where $\mathscr{P}$ is the set of image pixels, f_p is the binary label assigned to pixel p, f is collection of pixel-label assignments, $\mathscr{N}$ is a 4-neighborhood system, D_p is the data term, $V_{p,q}(f_p, f_q)$ is the smoothness term for two neighboring pixels.

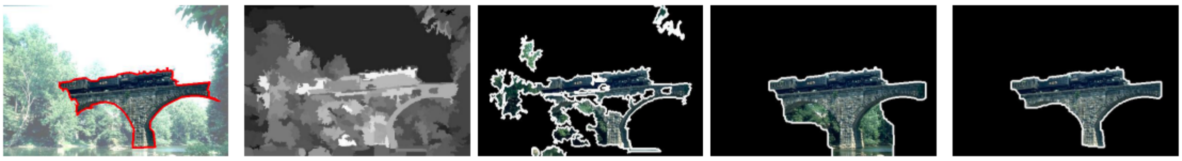
The smoothness term is the one from [4] which is

$$V_{p,q}(f_p, f_q) \propto \exp(-\frac{\Delta I^2}{2\sigma^2}).\delta(f_p \neq f_q) \tag{2}$$

Where ΔI denotes the intensity difference of two neighboring pixels, σ^2 is the variance of intensity difference of pixels, and $\delta(\cdot)$ is 1 if its argument is true and 0 otherwise. This $V_{p,q}$ encourages the boundary between labels to align with significant image edges. Data term consists of two parts

$$\begin{aligned} D_p(1) &= -\ln Pr(C_p|1) - \gamma.\ln(m_p) \\ D_p(0) &= -\ln Pr(C_p|0) - \gamma.\ln(1 - m_p) \end{aligned} \tag{3}$$

Where 1 is the salient object and 0 is the background, C_p is the quantized color of pixel p, m_p is the confidence of pixel p, and γ controls the relative importance of the two parts. m_p is renormalized to be in the range $[0, 1]$. $[0, 0.5]$ and $(0.5, 1]$ specifies the background and salient object, respectively.



(a) ground truth (b) saliency map (c) classifier result (d) GC refinement without confidences (e) GC refinement with confidences

Figure 1: Excluding vs. including confidences in Graph-cut segmentation

2 Experimental Results

Dataset	Saliency Map Error				Graph Cut Refinement Error			
	Itti	Hou	Achanta	Theirs	Itti	Hou	Achanta	Theirs
BSD	25.01	26.24	26.32	21.25	25.47	26.64	30.63	20.12
ASD	17.63	19.38	12.15	10.57	14.38	17.68	14.60	7.58
SED	22.00	25.71	25.17	12.99	21.48	25.81	25.85	15.56

Table 1: Performance comparison of different saliency detection methods. Columns $2-5$ show percentage errors obtained from saliency map. Columns $6-9$ show percentage errors after utilizing graph cut refinement.

References

[1] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 1–15, Springer Berlin Heidelberg, 2006. 1

[2] R. E. Schapire, *The Boosting Approach to Machine Learning: An Overview*, pp. 149–171. New York, NY: Springer New York, 2003. 1

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, Nov 2001. 1

[4] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision*, vol. 70, pp. 109–131, Nov 2006. 2