Boundary Driven Interactive Segmentation

Shai Bagon
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL 76100
Email: www.wisdom.weizmann.ac.il/~bagon

***Abstract*—This paper presents a novel approach and interface to interactive image segmentation. Our interface uses sparse and inaccurate boundary cues provided by the user to produce a multi-layer segmentation of the image. Using boundary cues allows our interface to utilize a single "boundary brush" to produce a multi-layer segmentation, making it appealing for devices with touch screen user interface. Our method utilizes recent advances in clustering to automatically recover the underlying number of layers without explicitly requiring the user to specify this input.**

I. Introduction

Interactive image segmentation is an important step in many image manipulation tools. Traditionally, interactive tools addressed the problem of figure/ground segmentation and offered "region-based" interaction. That is, the user marks *inside* the foreground and the background *regions* (see e.g., [1], [2], [3], [4]). These methods are quite effective for the binary segmentation problem of figure/ground separation. However, they are not trivial to extend to multiple layer segmentation. These extensions either require the use of multiple "brushes" to mark the different regions [5], [6], or recursive application of interactive binary segmentation [2]. Both of these interactive multi-layer segmentation approaches are quite cumbersome and restrict the user to work on the different layers in a "one-at-a-time" manner, rather than allow for him/her to *jointly* address the separation of all the layers in an image.

A completely different approach to interactive segmentation is suggested by [7], [8]. Instead of marking regions, the user delineates all the *boundaries* of the object, providing a trimap for the segmentation. This "trimap-based" method is very straight forward to extend to multiple layer separation: the same "brush" is used to draw a trimap around *all the boundaries* between the different layers. However, this extension to multiple layers is also not too practical: delineation of *all* the boundaries usually requires too much effort from the user.

In this work we propose a novel *boundary-based* approach for interactive segmentation. Our novel interactive approach requires the user to mark only *sparse and incomplete* boundary hints from which we are able to produce a multiple-layer segmentation of the image. Fig. 1 shows an example of our "boundary-based" interface and the resulting multi-layer segmentation. Our approach combines the merits of both region-based and trimap-based methods. On the one hand, it requires only sparse and crude boundary cues, much like the scribbles in the "region-based" method. While on the other hand, it trivially extends to multi layer segmentation, like the trimap-based approach. Thus we gain the trivial extendability of trimap interface without burdening the user with the necessity of tracing all the boundaries.

Our novel interface can be of special relevance in cases where the layer-of-interest is better defined via its boundaries (rather than via its appearance). Such cases may arise when trying to segment camouflaged objects or animals. Our proposed interface is also very intuitive for usage on touch-screen-based devices: all "touches" are interpreted in the same manner, as boundary strokes.

A possible extension to the proposed interface may be to add a second "brush" to indicate "non-boundary" region. That is, after applying one (or more) "boundary" scribbles the user wishes to refine the resulting multi-layer segmentation by indicating a falsely induced boundary. The second "non-boundary" brush is then applied to add a spare and crude scribble over the boundary indicating this boundary as "false".

The contribution of this paper are:

(i) Novel approach for user interaction: "boundary scribbles" instead of the traditional "region-scribbles".

(ii) Minimal "overhead" interaction effort: no need to "switch brushes", indicate desired number of layers, or any other auxiliary inputs. During interaction the user focuses solely on the image without the need for additional menubars or dialog boxes.

(iii) Our novel approach applicable "as-is" to multi layer segmentation.

The rest of this paper is organized as follows. Sec. II provides an overview of our new method for interaction, the algorithmic challenge it poses and how to utilize novel clustering approaches to overcome this challenge. Details specific to our implementation of our interactive method are given in Sec. III. Sec. IV provides experimental results. We conclude in Sec. V.

(a) Input image and boundary scribbles (red)

(b) Resulting segmentation

Figure 1. **Boundary-driven interactive multi-object segmentation:** *(a) The user provides only crude and partial indications to the locations of boundaries between the relevant objects in an image (red). The user does not specify how many layers he/she expect to get. (b) The output of our algorithm correctly segments the image into multiple segments. Correctly estimating the desired number of layers. Image was taken from [9].*

II. Boundary Driven Interactive Segmentation

While "region-based" and "trimap-based" interactive segmentation approaches get the number of layers as an input (either by the number of different "brushes", or the number of separated regions), our "boundary-based" interface does not explicitly get this crucial information from the user. Moreover, "region-based" and "trimap-based" approaches have access not only to the number of layers, but also have good information about the appearance of the different layers. Therefore, existing interactive segmentation methods builds explicit appearance models for the different layers (e.g., GMMs, histograms etc.) and uses very little gradient information only for boundary localization. That is, layers are separated according to their appearance-models.

On the other hand, the main challenge our boundary-based interactive approach faces is correctly inferring the underlying number of layers given the incomplete boundary scribbles marked by the user. Since the user provides only sparse boundary cues it is not straight forward to identify the different layers. Look, for example, at the image and user inputs in figure 1: the user provided four cues. However, there are only three layers: the tree, the squirrel and the background grass. It is up to our interface to tell that three of the boundary cues are related to the "tree" layer while the fourth marks the boundary of the "squirrel" layer. Inferring which cues are relevant to each of the layers is equivalent to inferring the underlying number of layers. Furthermore, without explicit identification of the layers in the image it is not possible to use appearance based models for separating the different layers.

Therefore, our novel interface calls for a different approach in order to cope with this challenges of identifying the underlying number of layers, associating the sparse cues with the correct layer boundaries, and the absence of appearance models for the different layers. Instead of appearance models, we use *affinity* model: we no longer associate pixels to learned appearance models, but rather use pair-wise affinities *between the pixels* to group them together to form the different layers. However, unlike traditional affinity based image segmentation methods (e.g., Ncuts [10]) that uses only positive (attraction) affinities between the pixels, we add *negative* affinities (repulsion) as well. Adding negative affinities allows us to utilize Correlation Clustering (CC) [11]: a more sophisticated clustering method that automatically recovers the underlying number of clusters.

Negative affinities in image segmentation come very naturally from boundary cues: pixels on the same side of a boundary are likely to be in the same layer (attraction), while pixels on opposite sides of a boundary are likely to be in different layers (repulsion). We use this observation to drive our novel paradigm to interactive multi-layer image segmentation.

When augmenting the proposed interface with the second type of "non-boundary" scribbles, all pixels within each "non-boundary" scribble induce strong attraction among themselves (i.e., likely to be in the same layer). Adding these attraction terms across the scribble should cancel out the repulsion terms that induced the undesired boundary.

Note that our approach is fundamentally different than the interface alluded to in [12, Fig.6]. Their approach is based on appearance models derived from "region-cues". The "boundary-cues" are only used to improve boundary localization through a non-submodular formulation.

III. Our Implementation

This section provides implementation details of our interactive interface: from boundary cues to multi-layer segmentation. Note that sparse boundary cues may be utilized in other different manners to provide multi-layer segmentation of an image.

A. The Correlation Clustering Functional

To make this work self-contained we provide a brief description of the CC functional and current methods for optimizing it.

Let $W \in \mathbb{R}^{n \times n}$ be an affinity matrix combining attraction ($w_{ij} > 0$) and repulsion ($w_{ij} < 0$). Any partition of n points can be written as $L \in \{1, 2, \dots\}^n$. The CC functional [11] maximizes the intra-cluster attraction. An optimal partition L maximizes:

$$E_{CC}(L) = \sum_{ij \in \mathcal{E}} w_{ij} [l_i = l_j] \quad (1)$$

Where $[\cdot]$ is the indicator function. That is, the CC functional strives to maximize the attraction within each and every cluster. Therefore, an optimal partition L, according to the CC functional, is such that all points within each cluster strongly attract each other, while points in different clusters repel each other.

The CC functional has a nice property: it is independent of the number of clusters k, and in fact it may be used to compare different partitions with *different* number of clusters (see [13, §2.1]). Therefore, the underlying number of clusters k is recovered by the optimal partition L that maximizes $E_{CC}(L)$.

Optimizing (1) is NP-hard [11]. Recently [13] showed an efficient approximation algorithms that can optimize (1) for large n efficiently.

B. From Cues to Affinities

A single boundary cue provides *local* information about one "meaningful" boundary in the image. This boundary is meaningful in the sense that it separates layers as the user wishes to define them. We would like to utilize this local information, propagate it intelligently from the boundary cue outward and induce pair-wise pixels affinities according to this local information.

To be more precise, the information a single cue provides has two parts: (a) All pixels across the cue are dissimilar and belong to different layers (repulsion). (b) All pixels on either side of the cue are similar and should belong to the same layer (attraction). Looking at Fig. 2(a) a boundary cue (marked red) suggests that all pixels mark dark are similar to each other, all pixels marked white are similar, while dark and white pixels are dissimilar. We would like to propagate this notion of similarity further away from the cue. That is, to induce similarity further away from the dark and white pixels. For this end we use semi-supervised method suggested by [14]: each cue now (softly) partition the pixels into those similar to its dark side and those similar to its white side (Fig. 2(b)). Note that other semi-supervised methods may be used for this end, e.g., [15], [16].

A layer in the image is characterized by pixels that agree among themselves on the "sidedness" of each scribble. That is they are all similar to the same side of each and every one of the user provided boundary cues. We may associate a membership vector v_i for every pixel i with an entry in the range $[-1, 1]$ for each cue. Taking the correlation between the membership vectors of two neighboring pixels indicates how much they agree or disagree on their "sidedness" for all the cues. Neighboring pixels with positive correlation are in agreement and should belong to the same layer, while neighboring pixels with negative correlation are in disagreement and should be put into different layers. These correlations combine information from all the boundary cues to induce positive (attraction) and negative (repulsion) affinities between neighboring pixels.

This process of propagating boundary information from the cues (which is an adaptation of a method of Stein *et al.* [17]) is illustrated in Fig. 2.

IV. Experimental Results

Fig. 3 shows input images and user marked boundary cues used for computing the affinity matrix alongside our multi-layer segmentation results. Our approach is able to optimize over the number of segments k and output a segmentation with a proper number of segments. Note that at no point did the user provide the underlying number of layers, and our method *automatically* recovered the desired number of layers.

V. Conclusion

This work presents a novel perspective and paradigm to interactive image segmentation. This paradigm is based on sparse and inaccurate *boundary* cues provided by the user. The proposed interface is minimal in the sense that it uses only a single "boundary" brush for *multi layer* segmentation. Using only a single brush makes it very easy and efficient to use (no need to switch brushes). It may also be a very intuitive interface for touch screens where all touches are interpreted as strokes from the same brush, without additional effort on switching brushes in the middle.

Our algorithm uses attraction and repulsion information induced by the boundary scribbles to partition the image into multiple layers. Combining attraction and repulsion information using the Correlation Clustering functional allows our algorithm to automatically recover the underlying number of layers.

References

[1] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004. 1

[2] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *ICCV*, 2001. 1

[3] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy snapping," in *TOG*, 2004. 1

[4] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? a unified approach to segment extraction," *ECCV*, 2008. 1

[5] J. Santner, T. Pock, and H. Bischof, "Interactive multi-label segmentation," in *ACCV*, 2011. 1

[6] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *CVPR*, 2009. 1

[7] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *ECCV*, 2004. 1

[8] J. Wang, M. Agrawala, and M. Cohen, "Soft scissors: an interactive tool for realtime high quality matting," in *TOG*, 2007. 1

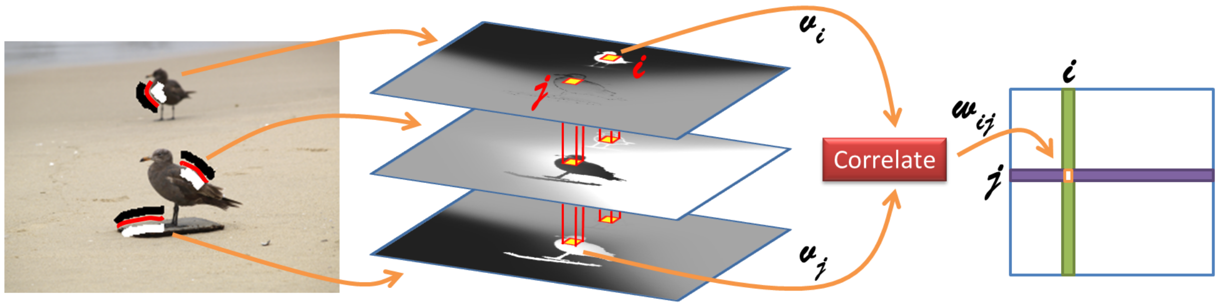
[9] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *CVPR*, 2007. 2, 4

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *PAMI*, 2000. 2

[11] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1, pp. 89–113, 2004. [Online]. Available: http://www.springerlink.com/content/r1011m8201733662/ 2, 3

[12] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *CVPR*, 2007. 2

[13] S. Bagon and M. Galun, "Optimizing large scale correlation clustering," in *arXiv:1112.2903*, 2011. 3



(a) Boundary scribbles → Matting constraints (b) "Soft" segmentations → segment membership vector (c) Membership correlations → affinity matrix

Figure 2. **From boundary scribbles to affinity matrix:** *(a) A boundary scribble is drawn by the user (red),inducing "figure/ground" regions on its opposite sides (black and white regions). (b) For each scribble we use the method of [14] to generate a soft segmentation of the image into two segments: pixel values in the soft segmentation are in the range $[-1, 1]$. Pixels far away from the scribble are assigned 0 as it is uncertain to what side they should belong to. Each pixel i is described using a segmentation membership vector v_i with an entry corresponding to its assignment at each soft segmentation (red columns). (c) A non-zero entry w_{ij} in the sparse affinity matrix is the* correlation *between normalized vectors v_i and v_j of neighboring pixels: $w_{ij} = v_i^T v_j / \|v_i\| \cdot \|v_j\|$. We also add strong repulsion across each scribble.*

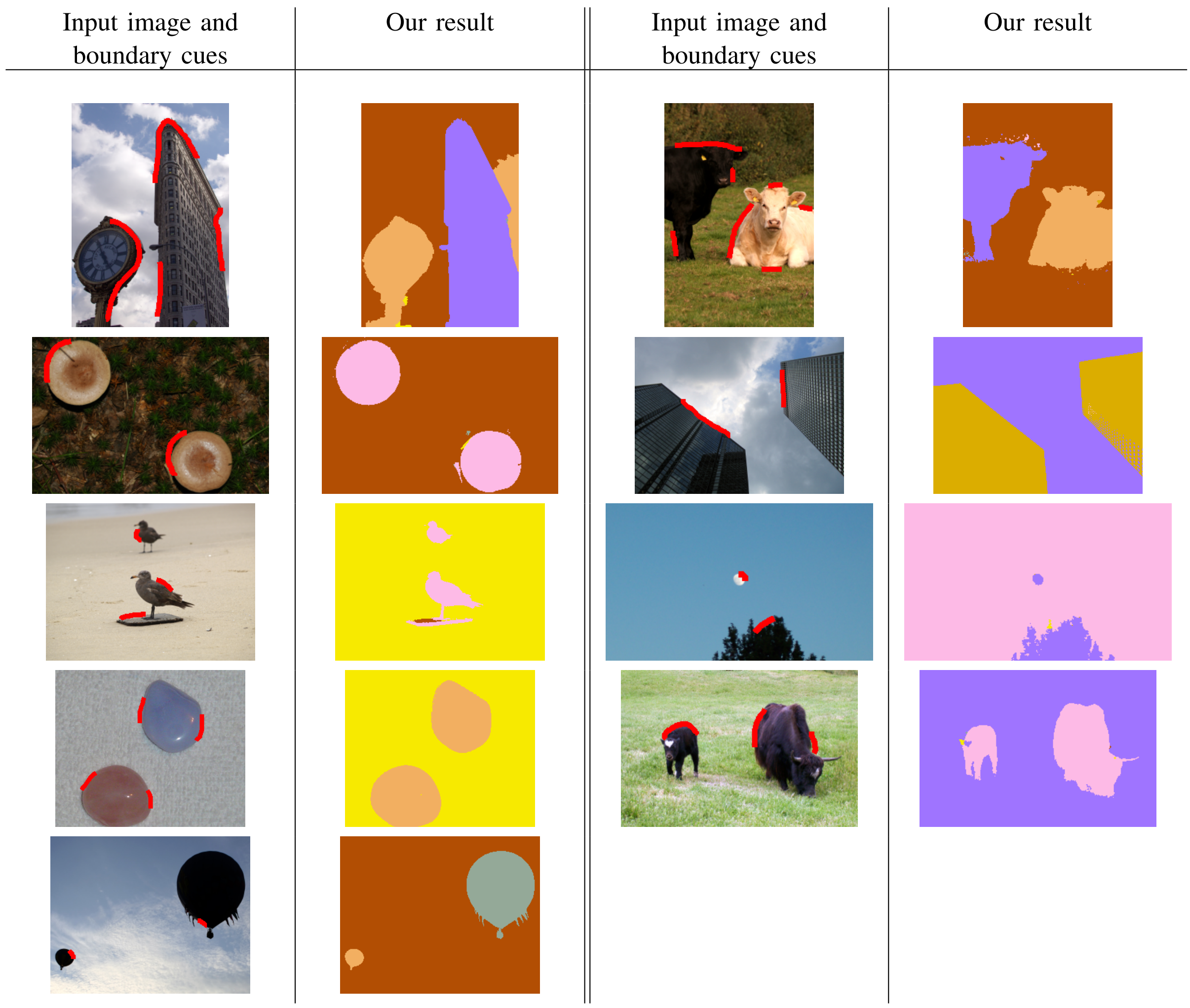


Figure 3. **Interactive segmentation results.** *Input image and user boundary cues (left), our result (right). Images were taken from [9].*

[14] A. Levin, R. Acha, and D. Lischinski, “Spectral matting,” *PAMI*, vol. 30, no. 10, pp. 1699–1712, 2008. 3, 4
[15] A. Bijral, N. Ratliff, and N. Srebro, “Semi-supervised learning with density based distances,” in *NIPS*, 2011. 3
[16] B. Nadler, N. Srebro, and X. Zhou, “Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data,” in *NIPS*, 2009. 3
[17] A. Stein, T. Stepleton, and M. Hebert, “Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection,” in *CVPR*, 2008. 3