Interactive Boundary Prediction for Object Selection

Hoang Le[1], Long Mai[2], Brian Price[2], Scott Cohen[2], Hailin Jin[2], and Feng Liu[1]

[1] Portland State University, Portland, OR, USA
[2] Adobe Research, San Jose, CA, USA

Abstract. Interactive image segmentation is critical for many image editing tasks. While recent advanced methods on interactive segmentation focus on the region-based paradigm, more traditional boundary-based methods such as Intelligent Scissor are still popular in practice as they allow users to have active control of the object boundaries. Existing methods for boundary-based segmentation solely rely on low-level image features, such as edges for boundary extraction, which limits their ability to adapt to high-level image content and user intention. In this paper, we introduce an interaction-aware method for boundary-based image segmentation. Instead of relying on pre-defined low-level image features, our method adaptively predicts object boundaries according to image content and user interactions. Therein, we develop a fully convolutional encoder-decoder network that takes both the image and user interactions (e.g. clicks on boundary points) as input and predicts semantically meaningful boundaries that match user intentions. Our method explicitly models the dependency of boundary extraction results on image content and user interactions. Experiments on two public interactive segmentation benchmarks show that our method significantly improves the boundary quality of segmentation results compared to state-of-the-art methods while requiring fewer user interactions.

1 Introduction

Separating objects from their backgrounds (the process often known as interactive object selection or interactive segmentation) is commonly required in many image editing and visual effect workflows [6, 25, 33]. Over the past decades, many efforts have been dedicated to interactive image segmentation. The main goal of interactive segmentation methods is to harness user input as guidance to infer the segmentation results from image information [11, 18, 22, 36, 30]. Many existing interactive segmentation methods follow the region-based paradigm in which users roughly indicate foreground and/or background regions and the algorithm infers the object segment. While the performance of region-based methods has improved significantly in recent years, it is still often difficult to accurately trace the object boundary, especially for complex cases such as textures with large patterns or low-contrast boundaries (Fig. 1).

To segment objects with high-quality boundaries, more traditional boundary-based interactive segmentation tools [11, 16, 28] are still popular in practice [6,

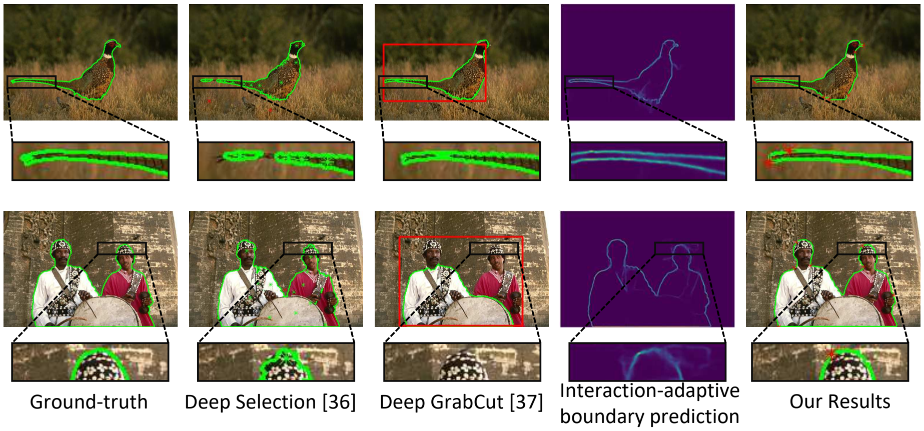


Fig. 1: Boundary-based segmentation with interactive boundary prediction. Our method adaptively predicts appropriate boundary maps for boundary-based segmentation, which enables segmentation results with better boundary quality compared to region-based approaches ([36, 37]) in challenging cases such as thin, elongated objects (1^{st} row), highly textured regions (2^{nd} row).

33]. These methods allow users to explicitly interact with boundary pixels and have a fine-grained control which leads to high-quality segmentation results. The main limitation faced by existing boundary-based segmentation methods, however, is that they often demand much more user input. One major reason is that those methods rely solely on low-level image features such as gradients or edge maps which are often noisy and lack high-level semantic information. Therefore, a significant amount of user input is needed to keep the boundary prediction from getting distracted by irrelevant image features.

In this paper, we introduce a new approach that enables a user to obtain accurate object boundaries with relatively few interactions. Our work is motivated by two key insights. First, a good image feature map for boundary-based segmentation should not only encode high-level semantic image information but also adapt to the user intention. Without high-level semantic information, the boundary extraction process would be affected by irrelevant high-signal background regions as shown in Fig. 1. Second, we note that a unique property of interactive segmentation is that it is inherently ambiguous without knowledge of the user intentions. The boundary of interest varies across different users and different specific tasks. Using more advanced semantic deep feature maps, which can partially address the problem, may risk missing less salient boundary parts that users want (Fig. 2). In other words, a good boundary prediction model should be made adaptively throughout segmentation process.

Our key idea is that instead of using a single feature map pre-computed independently from user interactions, the boundary map should be predicted adaptively as the user interacts. We introduce an interaction-adaptive boundary

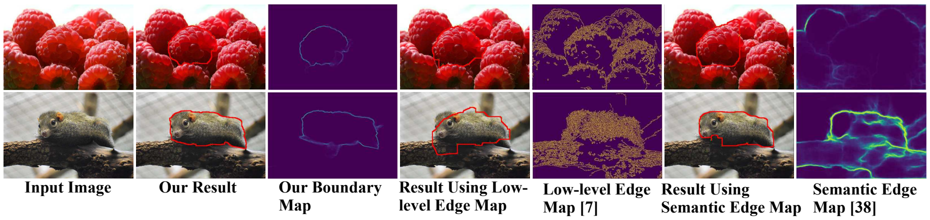


Fig. 2: Adaptive boundary map vs. pre-computed feature maps. Low-level image features (e.g. image gradient maps or edge maps) often lack high-level semantic information, which distracts the boundary extraction with irrelevant image details. Using more advanced semantic deep feature maps ([38]), while partially addressing the problem, may risk missing parts of the desired boundary as the user intention is unknown prior to interaction.

prediction model which predicts the object boundary while respecting both the image semantics and the user intention. Therein, we develop a convolutional encoder-decoder architecture for interaction-aware object boundary prediction. Our network takes the image and the user-specified boundary points as input and adaptively predicts the boundary map, which we call the interaction-adaptive boundary map. The resulted boundary map can then be effectively leveraged to segment the object using standard geodesic path solvers [11].

Our main contribution in this paper is the novel boundary-based segmentation framework based on interactive boundary prediction. Our method adaptively predicts the boundary map according to both the input image and the user provided control points. Our predicted boundary map can not only predict the high-level boundaries in the image but also adapt the prediction to respect the user intention. Evaluations on two interactive segmentation benchmarks show that our method significantly improves the segmentation boundary quality compared to state-of-the-art methods while requiring fewer user interactions.

2 Related Work

Many interactive object selection methods have been developed over the past decades. Existing methods can be categorized into two main paradigms: region-based and boundary-based algorithms [16, 22, 24]. Region-based methods let users roughly indicate the foreground and background regions using bounding boxes ([21, 30, 34, 37]), strokes ([2, 3, 5, 13, 15, 19, 22, 36]), or multi-label strokes [31]. The underlying algorithms infer the actual object segments based on this user feedback. Recent work in region-based segmentation has been able to achieve impressive object segmentation accuracy [36, 37], thanks to advanced deep learning frameworks. However, since no boundary constraints have been encoded, these methods often have difficulties generating high-quality segment boundaries, even with graph-cut based optimization procedures for post-processing.

Our research focuses on boundary-based interactive segmentation. This frameworks allow users to directly interact with object boundaries instead of image regions. Typically, users place a number of control points along the object boundary and the system optimizes the curves connecting those points in a piece-wise manner [9, 10, 26, 28, 32]. It has been shown that the optimal curves can be formulated as a minimal-cost path finding problem on grid-based graphs [11, 12]. Boundary segments are extracted as geodesic paths (i.e. minimal paths) between the user provided control points where the path cost is defined by underlying feature maps extracted from the image [9, 10, 17, 26–28]. One fundamental limitation is that existing methods solely rely on low-level image features such as image gradient or edge maps, which prevents leveraging high-level image semantics. As a result, users must control the curve carefully which demands significant user feedback for difficult cases. In this paper, we introduce an alternative approach which predicts the boundary map adaptively as users interacts. In our method, the appropriate boundary-related feature map is generated from a boundary map prediction model, leveraging the image and user interaction points as inputs.

Significant research has been conducted to better handle noisy low-level feature maps for boundary extraction [9, 10, 26, 27, 32]. The key principle is to leverage advanced energy models and minimal path finding methods that enable the incorporation of high-level priors and regularization such as curvature penalization [9, 10, 27], boundary simplicity [26], and high-order regularization [32]. Our work in this paper follows an orthogonal direction and can potentially benefit from the advances in this line of research. While those methods focus on developing new path solvers that work better with traditional image feature maps, we focus on obtaining better feature maps from which high-quality object boundaries can be computed using standard path solvers.

Our research is in part inspired by recent successes of deep neural networks in semantic edge detection [23, 35, 38]. It has been shown that high-level semantic edge and object contours can be predicted using convolutional neural networks trained end-to-end on segmentation data. While semantic edge maps can address the aforementioned lack of semantics in low-level feature maps, our work demonstrates that it is possible and more beneficial to go beyond pre-computed semantic edge maps. This paper is different from semantic edge detection in that we aim to predict the interaction-adaptive boundary with respect to not only the image information but also the user intention.

Our method determines the object boundary segments by connecting pairs of control points placed along the object boundary. In that regard, our system shares some similarities with the PolygonRNN framework proposed by Castrejon et al. [8]. There are two important differences between our method and PolygonRNN. First, our method takes arbitrary set of control points provided by the users while PolygonRNN predicts a set of optimal control points from an initial bounding box. More importantly, PolygonRNN mainly focuses on predicting the control points. They form the final segmentation simply by connecting those points with straight lines, which does not lead to highly accurate boundaries. Our method, on the other hand, focuses on predicting a boundary map from the

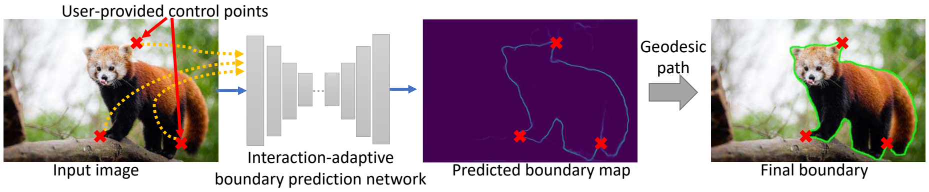


Fig. 3: Boundary extraction with interactive boundary map prediction. Given an image and a set of user provided control points, the boundary prediction network is used to predict a boundary map that reflects both high-level semantics in the image and user intention encoded in the control points to enable effective boundary extraction.

user provided control points. The predicted boundary map can then be used to extract high-quality object boundaries with a minimal path solver.

3 Interactive Boundary Prediction for Object Selection

We follow the user interaction paradigm proposed by recent works in boundary-based segmentation [9, 10, 26] to support boundary segmentation with sparse user inputs: given an image and a set of user provided control points along the desired object boundary, the boundary segments connecting each pair of consecutive points are computed as minimal-cost paths in which the path cost is accumulated based on an underlying image feature map. Different from existing works in which the feature maps are low-level and pre-computed before any user interaction, our method adapts the feature map to user interaction: the appropriate feature map (boundary map) is predicted on-the-fly during the user interaction process using our boundary prediction network. The resulting boundary prediction map is used as the input feature map for a minimal path solver [12] to extract the object boundary. Fig. 3 illustrates our overall framework.

3.1 Interaction-Adaptive Boundary Prediction Network

The core of our framework is the interaction-adaptive boundary map prediction network. Given an image and an ordered set of user provided control points as input, our network outputs a predicted boundary map.

Our interactive boundary prediction network follows a convolutional encoder-decoder architecture. The encoder consists of five convolutional blocks, each contains a convolution-ReLU layer and a 2×2 Max-Pooling layer. All convolutional blocks use 3×3 kernels. The decoder consists of five up-convolutional blocks, with each up-convolutional layer followed by a ReLU activation. We use 3×3 kernels for the first two up-convolutional blocks, 5×5 kernels for the next two blocks, and 7×7 kernels for the last blocks. To avoid blurry boundary prediction results, we include three skip-connections from the output of the encoder's first

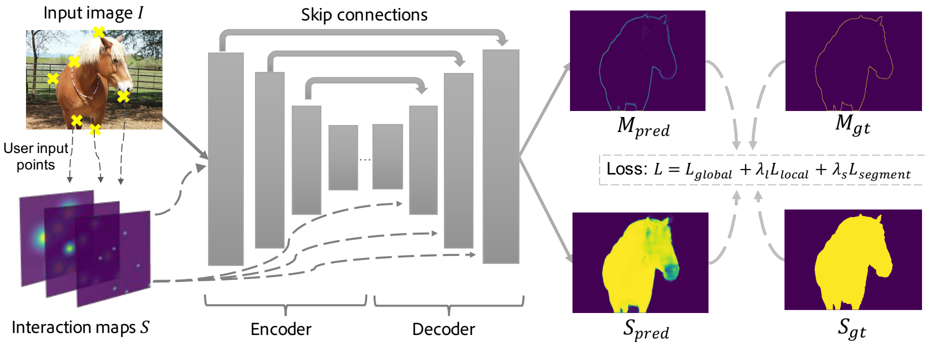


Fig. 4: Interactive boundary prediction network. The user-provided input points are converted to interaction maps S to use along with the image I as input channels for an encoder-decoder network. The predicted boundary map M_{pred} and segment map S_{pred} are used along with the corresponding ground-truth maps M_{gt}, S_{gt} to define the loss function during training.

three convolutional blocks to the decoder's last three deconvolutional blocks. The network outputs are passed through a sigmoid activation function to transform their values to the range $[0, 1]$. Fig. 4 illustrates our network model. It takes the concatenation of the RGB input image I and interaction maps as input. Its main output is the desired predicted boundary map. Additionally, the network also outputs a rough segmentation mask used for computing the loss function during training as described below.

Input Representation: To serve as the prediction network's input channels, we represent the user control points as 2-D maps which we call *interaction maps*. Formally, let $C = \{c_i | i = 1..N\}$ be spatial coordinates of the N user control points along the boundary. We compute a two-dimensional spatial map $S^{\sigma}_{c_i}$ for each point c_i as $S^{\sigma}_{c_i}(p) = \exp\left(\frac{-d(p,c_i)^2}{2(\sigma \cdot L)^2}\right)$ where $d(p, c_i)$ represents the Euclidean distance between pixel p and a control point c_i. L denotes the length of the smaller side of the image. Combining the interaction maps $S^{\sigma}_{c_i}$ from all individual control points c_i's with the pixel-wise max operator, the overall interaction map S for the control point set C is obtained.

The parameter σ controls the spatial extent of the control point in the interaction map. We observe that different values of σ offer different advantages. While a small σ value provides exact information about the location of selection, a larger σ value tends to encourage the network to learn features at larger scopes. In our implementation, we create three interaction maps with $\sigma \in \{0.02, 0.04, 0.08\}$ and concatenate them depth-wise to form the input for the network.

3.2 Loss Functions

During training, each data sample consists of an input image I and a set of control points $C = \{c_i\}$ sampled along the boundary of one object. Let θ denote the network parameters to be optimized during training. The per-sample loss function is defined as

$$L(I, \{c_i\}; \theta) = L_{global}(I, \{c_i\}; \theta) + \lambda_l L_{local}(I, \{c_i\}; \theta) + \lambda_s L_{seg}(I, \{c_i\}; \theta) \quad (1)$$

where L_{local}, L_{global}, and $L_{segment}$ are the three dedicated loss functions designed specifically to encourage the network to leverage the global image semantic and the local boundary patterns into the boundary prediction process. λ_l and λ_s are the weights to balance the contribution of the loss terms. In our experiment, λ_l and λ_s are chosen to be 0.25 and 1.0 respectively using cross validation.

Global Boundary Loss: This loss encourages the network to learn useful features to detect the pixels belonging to the appropriate boundary. We treat the boundary detection problem as pixel-wise binary classification. The boundary pixel detection loss is defined using the binary cross entropy loss [4, 14]

$$L_{global}(I, \{c_i\}; \theta) = \frac{-M_{gt} \cdot \log(M_{pred})^\top - (1 - M_{gt}) \cdot \log(1 - M_{pred})^\top}{|M_{gt}|} \quad (2)$$

where $M_{pred} = F_B(I, \{c_i\}; \theta)$ denotes the predicted boundary map straightened into a row vector. $|M_{gt}|$ denotes the total number of pixels in the ground-truth boundary mask M_{gt} (which has value 1 at pixels on the desired object boundary, and 0 otherwise). Minimizing this loss function encourages the network to be able to differentiate boundary and non-boundary pixels.

Local Selection-Sensitive Loss: We observe that a network trained with only L_{global} may perform poorly at difficult local boundary regions such as those with weak edges or complex patterns. Therefore, we design the local loss term L_{local} which penalizes low-quality boundary prediction near the user selection points.

Let G_i denote a spatial mask surrounding the control point c_i. Let $M_i = F_B(I, C_i; \theta)$ be the predicted boundary map generated with only one control point c_i. The local loss L_{local} is defined as a weighted cross entropy loss

$$L_{local}(I, \{c_i\}; \theta) = \frac{1}{|C|} \sum_{c_i \in C} \frac{-M_{gt} \odot G_i \cdot \log(M_i \odot G_i)^\top - (1 - M_{gt} \odot G_i) \cdot \log(1 - M_i \odot G_i)^\top}{|M_{gt}|} \quad (3)$$

where $\odot$ denotes the element-wise multiplication operation. This loss function is designed to explicitly encourage the network to leverage local information under the user selected area to make good localized predictions. To serve as the local mask, we use the interaction map component with $\sigma = 0.08$ at the corresponding location. Instead of aggregating individual interaction maps, we form a batch of inputs, each with the interaction map corresponding to one input control point. The network then produces a batch of corresponding predicted maps which are used to compute the loss value.

Segmentation-Aware Loss: While the boundary losses defined above encourage learning boundary-related features, it tends to lack the knowledge of what distinguishes foreground and background regions. Having some knowledge about whether neighboring pixels are likely foreground or background can provide useful information to complement the boundary detection process. We incorporate a segmentation prediction loss to encourage the network to encode knowledge of foreground and background. We augment our network with an additional decision layer to predict the segmentation map in addition to the boundary map.

Let $S_{pred} = F_S(I, \{c_i\}; \theta)$ denote the segmentation map predicted by the network. The loss function is defined in the form of binary cross entropy loss on the ground-truth binary segmentation map S_{gt} whose pixels have value 1 inside the object region, and 0 otherwise.

$$L_{segment}(I, \{c_i\}; \theta) = \frac{-S_{gt} \cdot \log(S_{pred})^\top - (1 - S_{gt}) \cdot \log(1 - S_{pred})^\top}{|S_{gt}|} \quad (4)$$

We note that all three loss terms are defined as differentiable functions over the network's output. The network parameters θ can hence be updated via back-propagation during training with standard gradient based methods [14].

3.3 Implementation Details

Our boundary prediction model is implemented in TensorFlow [1]. We train our network using the ADAM optimizer [20] with initial learning rate $\eta = 10^{-5}$. The network is trained for one million iterations, which takes roughly one day on an NVDIA GTX 1080 Ti GPU.

Network training with synthetic user inputs. To train our adaptive boundary prediction model, we collect samples from an image segmentation dataset [38] which consists of 2908 images from the PASCAL VOC dataset, post-processed for high-quality boundaries. Each training image is associated with multiple object masks. To create each data sample, we randomly select a subset of them to create the ground-truth boundary mask. We then randomly select k points along the ground-truth boundary to simulate user provided control points. Our training set includes data samples with k randomly selected in the range of 2 and 100 to simulate the effect of varying difficulty. We also use cropping, scaling, and blending for data augmentation.

Training with multi-scale prediction. To encourage the network to learn useful features to predict boundary at different scales, we incorporate multi-scale prediction into our method. Specifically, after encoding the input, each of the last three deconvolutional blocks of the decoder is trained to predict the boundary represented at the corresponding scale. The lower layers are encouraged to learn useful information to capture the large-scale boundary structure, while higher layers are trained to reconstruct the more fine-grained details. To encourage the network to take the user selection points into account, we also concatenate each decoder layer with the user selection map S described in Section 3.1.

Running time. Our system consists of two steps. The boundary map prediction step, running a single feed-forward pass, takes about 70 milliseconds. The shortest-path-finding step takes about 0.17 seconds to connect a pair of control points of length 300 pixels along the boundary.

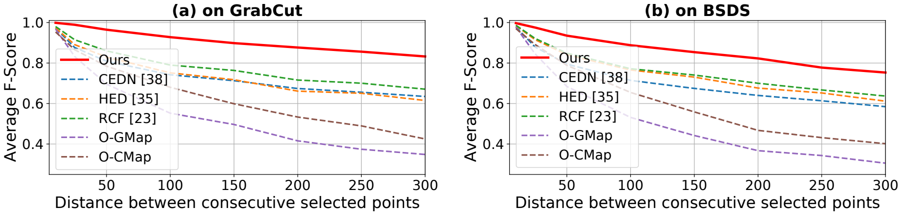


Fig. 5: Boundary quality at different boundary segment lengths. As expected, for all methods, the F-score quality decreases as l increases. Our adaptively predicted map consistently obtains higher F-score than non-adaptive feature maps. More importantly, our method performs significantly better with long boundary segments.

4 Experiments

We evaluate our method on two public interactive image segmentation benchmarks **GrabCut** [30] and **BSDS** [24] which consist of 50 and 96 images, respectively. Images in both datasets are associated with human annotated high-quality ground-truth object masks. For evaluation, we make use of two segmentation metrics proposed in [29]:

Intersection over Union (IU): This is a region-based metric which measures the intersection over the union between a predicted segmentation mask S_{pred} and the corresponding ground-truth mask S_{gt}.

Boundary-Based F-score: This metric is designed to specifically evaluate the boundary quality of the segmentation result [29]. Given the ground-truth boundary map B_{gt} and the predicted boundary map B_{pred} connecting the same two control points, the F-score quality of B_{pred} is measured as:

$$F(B_{pred}; B_{gt}) = \frac{2 \times P(B_{pred}; B_{gt}) \times R(B_{pred}; B_{gt})}{P(B_{pred}; B_{gt}) + R(B_{pred}; B_{gt})} \tag{5}$$

The P and R denote the precision and recall values, respectively computed as:

$$P(B_{pred}; B_{gt}) = \frac{|B_{pred} \odot dil(B_{gt}, w)|}{|B_{pred}|}; R(B_{pred}; B_{gt}) = \frac{|B_{gt} \odot dil(B_{pred}, w)|}{|B_{gt}|} \tag{6}$$

where $\odot$ represents the pixel-wise multiplication between maps. $dil(B, w)$ denotes the dilation operator expanding the map B by w pixels. In our evaluation, we use $w = 2$ to emphasize accurate boundary prediction.

4.1 Effectiveness of Adaptive Boundary Prediction

This paper proposes the idea of adaptively generating the boundary map along with the user interaction instead of using pre-computed low-level feature maps. Therefore, we test the effectiveness of our adaptively predicted boundary map

		CEDN [38]	**HED [35]**	**RCF [23]**	**O-GMap**	**O-CMap**	**Ours**
GrabCut	**F-score**	0.7649	0.7718	0.8027	0.5770	0.6628	***0.9134***
	IU	0.8866	0.8976	0.9084	0.8285	0.8458	***0.9158***
BSDS	**F-score**	0.6825	0.7199	0.7315	0.5210	0.6060	***0.7514***
	IU	0.7056	0.7241	0.7310	0.6439	0.7230	***0.7411***

Table 1: Average segmentation quality from different feature maps.

compared to non-adaptive feature maps in the context of path-based boundary extraction. To evaluate that quantitatively, we randomly sample the control points along the ground-truth boundary of each test image such that each pair of consecutive points are l pixels apart. We create multiple control point sets for each test image using different values of l ($l \in \{5, 10, 25, 50, 100, 150, 200, 250, 300\}$). We then evaluate each feature map by applying the same geodesic path solver [12] to extract the boundary-based segmentation results from the feature map and measure the quality of the result. We compare our predicted boundary map with two classes of non-adaptive feature maps:

Low-level Image Features. Low-level feature maps based on image gradient are widely used in existing boundary-based segmentation works [11, 18, 26, 28]. In this experiment, we consider two types of low-level feature maps: continuous image gradient maps and binary Canny edge maps [7]. We generate multiple of these maps from each test image using different edge sensitivity parameters ($\sigma \in 0.4, 0.6, 0.8, 1.0$). We evaluate results from all the gradient maps and edge maps and report the oracle best results among them which we named as **O-GMap** (for gradient maps) and **O-CMap** (for Canny edge maps).

Semantic Contour Maps. We also investigate replacing the low-level feature maps with semantic maps. In particular, we consider the semantic edge map produced by three state-of-the-art semantic edge detection methods [23, 35, 38], denoted as **CEDN**, **HED**, and **RCF** in our experiments.

Table 1 compares the overall segmentation result quality of our feature maps as well as the non-adaptive feature maps. The reported IU and F-score values are averaged over all testing data samples. This result indicates that in general the boundary extracted from our adaptive boundary map better matches the ground-truth boundary compared to those extracted from non-adaptive feature maps, especially in terms of the boundary-based quality metric F-score.

We further inspect the average F-score separately for different boundary segment lengths l. Intuitively, the larger the value of l the further the controls points are apart, making it more challenging to extract an accurate boundary. Fig. 5 shows how the F-scores quality varies for boundary segments with different lengths l. As expected, for all methods, the F-score quality decreases as l increases. Despite that, we can observe the quality of our adaptively predicted map is consistently higher than that of non-adaptive feature map. More importantly, our method performs significantly better with long boundary segments, which demonstrates the potential of our method to extract the full object boundary with far fewer user clicks.

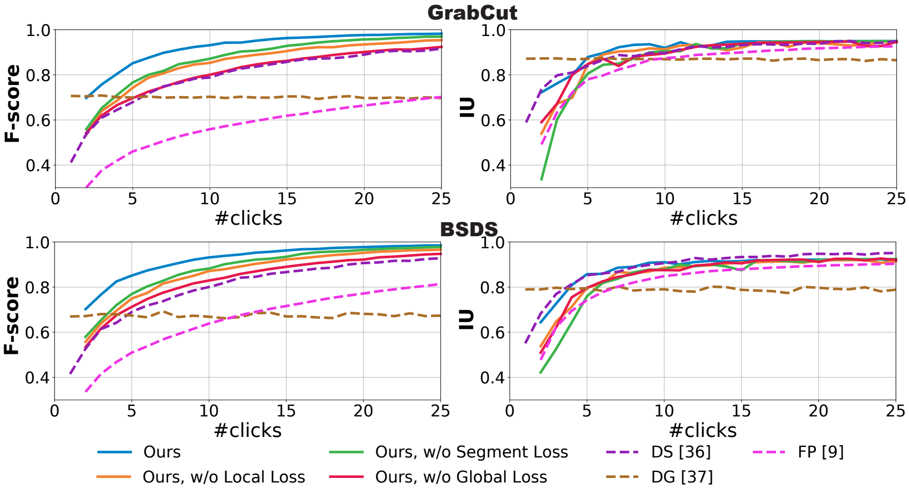


Fig. 6: Interactive segmentation quality. In terms of region-based metric IU, our method performs comparably with the state-of-the-art region-based method DS. Notably, our method significantly outperforms DS in terms of boundary F-score.

4.2 Interactive Segmentation Quality

The previous experiment evaluates the segmentation results generated when the set of control points are provided all at once. In this section, we evaluate our method in a more realistic interactive setting in which control points are provided sequentially during the segmentation process.

Evaluation with Synthetic User Inputs. Inspired by previous works on interactive segmentation [15, 36], we quantitatively evaluate the segmentation performance by simulating the way a real user sequentially adds control points to improve the segmentation result. In particular, each time a new control point is added, we update the interaction map (Section 3.1) and use our boundary prediction network to re-generate the boundary map which in turn is used to update the segmentation result. We mimic the way a real user often behaves when using our system: a boundary segment (between two existing consecutive control points) with lowest F-score values is selected. From the corresponding ground-truth boundary segment, the simulator selects the point farthest from the currently predicted segment to serve as the new control point. The process starts with two randomly selected control points and continues until the maximum number of iterations (chosen to be 25 in our experiment) is reached.

We compare our method with three state-of-the-art interactive segmentation algorithms, including two region-based methods Deep Object Selection (DS) [36], Deep GrabCut (DG) [37] and one advanced boundary-based method Finsler-based Path Solver (FP) [9]. Note that FP uses the same user interaction mode as ours. Therefore, we evaluate those methods using the same simulation process as ours. For DS, we follow the simulation procedure described in [36] using the

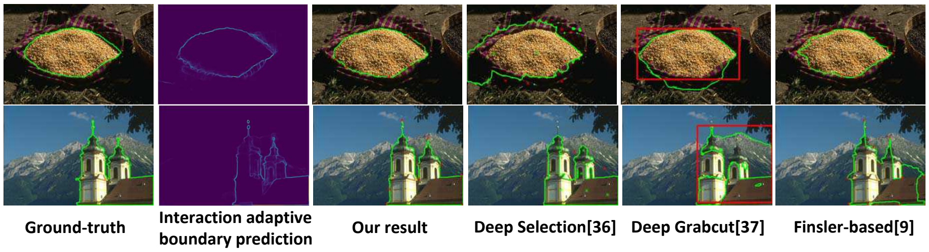


Fig. 7: Visual comparison of segmentation results. We compare the segmentation results of our method to three state-of-the-art interaction segmentation methods.

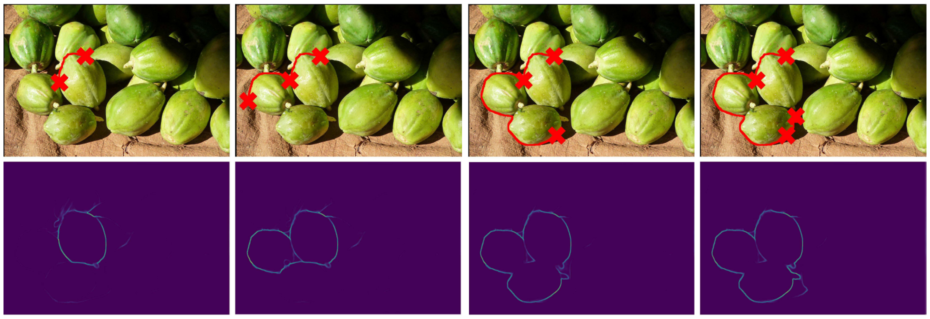


Fig. 8: Adaptivity analysis. By learning to predict the object boundary using both image content and user input, the boundary map produced by our network can evolve adaptively to reflect user intention as more input points are provided.

author provided implementation. For DG, we use the following simulation strategy: at the k^{th} simulation step, k bounding boxes surrounding the ground-truth mask are randomly sampled. We always additionally include the tightest bounding box. From those bounding boxes, we use DG to generate k segmentation results and the highest-score one is selected as the result for that iteration.

Fig. 6 shows the average F-score and IU of each method for differing numbers of simulation steps on the GrabCut and the BSDS datasets. In terms of the region-based metric IU, our method performs as well as the state-of-the-art region-based method DS. Notably, our method significantly outperforms DS in terms of boundary F-score, which confirms the advantage of our method as a boundary-based method. This result demonstrates that our method can achieve superior boundary prediction even with fewer user interactions. We also perform an ablation study, evaluating the quality of the results generated with different variants of our boundary prediction network trained with different combinations of the loss functions. Removing each loss term during the network training tends to decrease the boundary-based quality of the resulting predicted map.

Fig. 7 shows a visual comparison of our segmentation results and other methods after 15 iterations. These examples consist of objects with highly textured and low-contrast regions which are challenging for region-based segmentation as they rely on boundary optimization process such as graph-cut [36] or dense-CRF [37]. Our model, in contrast, learns to predict the boundary directly from both the input image and the user inputs to better handle these cases.

To further understand the advantage of our adaptively predicted map, we visually inspect the boundary maps predicted by our network as input points are added (Fig. 8). We observe that initially when the number of input points are too few to depict the boundary, the predicted boundary map tends to focus its confidence value at the local boundary regions surrounding the selected points and may generate some fuzzy regions. As more input points are provided, our model leverages the information from the additional points to update its prediction which can accurately highlight the desired boundary regions and converge to the correct boundary with a sufficient number of control points.

4.3 Evaluation with Human Users

We examine our method when used by human users with a preliminary user study. In this study, we compare our method with Intelligent Scissors (IS) [28] which is one of the most popular object selection tool in practice [25, 33]. We utilize a publicly available implementation of IS[3]. In addition, we also experiment with a commercial version of IS known as Adobe Photoshop Magnetic Lasso (ML) which has been well optimized for efficiency and user interaction. Finally, we also include the state-of-the-art region-based system Deep Selection (DS) [36] in this study.

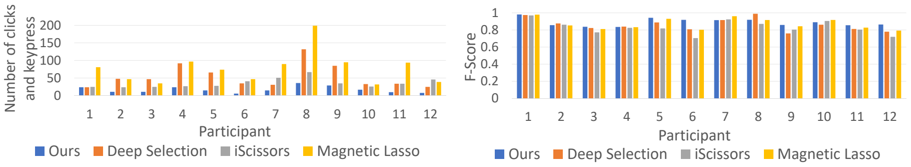
We recruit 12 participants for the user study. Given an input image and the expected segmentation result, each participant is asked to sequentially use each of the four tools to segment the object in the image to reproduce the expected result. Participants are instructed to use each tool as best as they can to obtain the best results possible. Prior to the study, each participant is provided a comprehensive training session to help them familiarize with the tasks and the segmentation tools. To represent challenging examples encountered in real-world tasks, we select eight real-world examples from the online image editing forum Reddit Photoshop Requests[4] by browsing with the keywords “isolate”, “crop”, and “silhouette” and picked the images that have a valid result accepted by the requester. Each image is randomly assigned to the participants. To reduce the order effect, we counter-balance the order of the tools used among participants.

Fig. 9 shows the amount of interaction (represented as number of mouse clicks) that each participant used with each methods and the corresponding segmentation quality. We observe that in most cases, the results obtained from our method are visually better or comparable with competing methods while needing much fewer user interactions.

Robustness against imperfect user inputs. To examine our method’s robustness with respect to noisy user inputs, we re-run the experiment in Section

[3] github.com/AzureViolin

[4] www.reddit.com/r/PhotoshopRequest



(a) Number of user clicks

(b) Segmentation quality

Fig. 9: Evaluation with real user inputs. In general, our method enables users to obtain segmentation results with better or comparable quality to state-of-the-art methods while using fewer interactions.

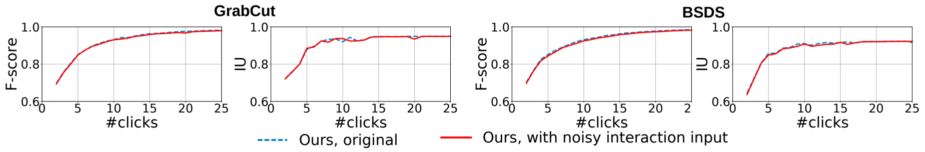


Fig. 10: Our method is robust against noisy interaction inputs

4.2 with randomly perturbed simulated input points. Each simulated control point $c_i = (x_i, y_i)$ is now replaced by its noisy version $c'_i = (x_i + \delta_x, y_i + \delta_y)$. δ_x and δ_y are sampled from the real noise distribution gathered from our user study data (Section 4.3). For each user input point obtained in the user study, we identify the closest boundary point from it and measure the corresponding δ_x and δ_y. We collect the user input noise over all user study sessions to obtain the empirical noise distribution and use it to sample δ_x, δ_y. Fig. 10 shows that our method is robust against the noise added to the input control points.

5 Conclusion

In this paper, we introduce a novel boundary-based segmentation method based on interaction-aware boundary prediction. We develop an adaptive boundary prediction model predicting a boundary map that is not only semantically meaningful but also relevant to the user intention. The predicted boundary can be used with an off-the-shelf minimal path finding algorithm to extract high-quality segmentation boundaries. Evaluations on two interactive segmentation benchmarks show that our method significantly improves the segmentation boundary quality compared to state-of-the-art methods while requiring fewer user interactions. In future work, we plan to further extend our algorithm and jointly optimize both the boundary map prediction and the path finding in a unified framework.

Acknowledgments This work was partially done when the first author was an intern at Adobe Research. Fig. 2 uses images from Flickr user Liz West and Laura Wolf, Fig. 3 uses an image from Flickr user Mathias Appel, and Fig. 8 uses an image from Flickr user GlobalHort Image Library/ Imagetheque under a Creative Commons license.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016)
2. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(6), 641–647 (1994)
3. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE International Conference on Computer Vision. pp. 1–8 (2007)
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag (2006)
5. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision **70**(2), 109–131 (2006)
6. Brinkmann, R.: The Art and Science of Digital Compositing. Morgan Kaufmann Publishers Inc., 2 edn. (2008)
7. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**(6), 679–698 (1986)
8. Castrejon, L., Kundu, K., Urtasun, R., Fidler, S.: Annotating object instances with a polygon-rnn. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4485–4493 (2017)
9. Chen, D., Mirebeau, J.M., Cohen, L.D.: A new finsler minimal path model with curvature penalization for image segmentation and closed contour detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 355–363 (2016)
10. Chen, D., Mirebeau, J.M., Cohen, L.D.: Global minimum for a finsler elastica minimal path approach. International Journal of Computer Vision **122**(3), 458–483 (2017)
11. Cohen, L.: Minimal Paths and Fast Marching Methods for Image Analysis, pp. 97–111. Springer US, Boston, MA (2006)
12. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: a minimal path approach. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 666–673 (1996)
13. Criminisi, A., Sharp, T., Blake, A.: Geos: Geodesic image segmentation. In: European Conference on Computer Vision. pp. 99–112 (2008)
14. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
15. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3129–3136 (2010)
16. He, J., Kim, C.S., Kuo, C.C.J.: Interactive Image Segmentation Techniques, pp. 17–62 (2014)
17. Jung, M., Peyré, G., Cohen, L.D.: Non-local Active Contours, pp. 255–266 (2012)
18. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**(4), 321–331 (1988)
19. Kim, T.H., Lee, K.M., Lee, S.U.: Generative image segmentation using random walks with restart. In: European Conference on Computer Vision. pp. 264–275. Springer (2008)

20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
21. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: IEEE International Conference on Computer Vision. pp. 277–284 (2009)
22. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Transactions on Graphics **23**(3), 303–308 (2004)
23. Liu, Y., Cheng, M.M., Hu, X., Wang, K., Bai, X.: Richer convolutional features for edge detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5872–5881 (2017)
24. McGuinness, K., Oconnor, N.E.: A comparative evaluation of interactive segmentation algorithms. Pattern Recognition **43**(2), 434–444 (2010)
25. McIntyre, C.: Visual Alchemy: The Fine Art of Digital Montage. Taylor & Francis (2014)
26. Mille, J., Bougleux, S., Cohen, L.D.: Combination of piecewise-geodesic paths for interactive segmentation. International Journal of Computer Vision **112**(1), 1–22 (2015)
27. Mirebeau, J.M.: Fast-marching methods for curvature penalized shortest paths. Journal of Mathematical Imaging and Vision (Dec 2017)
28. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: Annual Conference on Computer Graphics and Interactive Techniques. pp. 191–198. SIGGRAPH '95, ACM, New York, NY, USA (1995)
29. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 724–732 (2016)
30. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23**(3) (2004)
31. Santner, J., Pock, T., Bischof, H.: Interactive multi-label segmentation. In: Asian Conference on Computer Vision. pp. 397–410. Springer (2010)
32. Ulen, J., Strandmark, P., Kahl, F.: Shortest paths with higher-order regularization. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(12), 2588–2600 (2015)
33. Whalley, R.: Photoshop Layers: Professional Strength Image Editing:. Lenscraft Photography (2015)
34. Wu, J., Zhao, Y., Zhu, J., Luo, S., Tu, Z.: Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 256–263 (2014)
35. Xie, S., Tu, Z.: Holistically-nested edge detection. In: IEEE International Conference on Computer Vision. pp. 1395–1403 (2015)
36. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 373–381 (2016)
37. Xu, N., Price, B.L., Cohen, S., Yang, J., Huang, T.S.: Deep grabcut for object selection. In: British Machine Vision Conference (2017)
38. Yang, J., Price, B., Cohen, S., Lee, H., Yang, M.H.: Object contour detection with a fully convolutional encoder-decoder network. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 193–202 (2016)