This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Content-Aware Multi-Level Guidance for Interactive Instance Segmentation

Soumajit Majumder
Institute of Computer Science II
University of Bonn, Germany
majumder@cs.uni-bonn.de

Angela Yao
School of Computing
National University of Singapore
yaoa@comp.nus.edu.sg

Abstract

In interactive instance segmentation, users give feedback to iteratively refine segmentation masks. The user-provided clicks are transformed into guidance maps which provide the network with necessary cues on the whereabouts of the object of interest. Guidance maps used in current systems are purely distance-based and are either too localized or non-informative. We propose a novel transformation of user clicks to generate content-aware guidance maps that leverage the hierarchical structural information present in an image. Using our guidance maps, even the most basic FCNs are able to outperform existing approaches that require state-of-the-art segmentation networks pre-trained on large scale segmentation datasets. We demonstrate the effectiveness of our proposed transformation strategy through comprehensive experimentation in which we significantly raise state-of-the-art on four standard interactive segmentation benchmarks.

1. Introduction

Interactive object selection and segmentation allows users to interactively select objects of interest down to the pixel level by providing inputs such as clicks, scribbles, or bounding boxes. The segmented results are useful for downstream applications such as image/video editing [6, 30], image-based medical diagnosis [50, 51], human-machine collaborative annotation [2], *etc*. GrabCut [45] is a pioneering example of interactive segmentation which segments objects from a user-provided bounding box by iteratively updating a colour-based Gaussian mixture model. Other methods include Graph Cuts [7], Random Walk [18] and GeoS [13] though more recent methods [32, 35, 36, 52, 53] approach the problem with deep learning architectures such as convolutional neural networks (CNNs).

In standard, non-interactive instance segmentation [4, 15, 21, 22, 23], the RGB image is given as input and segmentation masks for each object instance are predicted. In an interactive setting, however, the input consists of the RGB image as well as *'guidance'* maps based on user-provided supervision. The guidance map helps to select the specific instance to segment; when working in an iterative setting, it can also help correct errors from previous segmentations [6, 32, 35, 53].

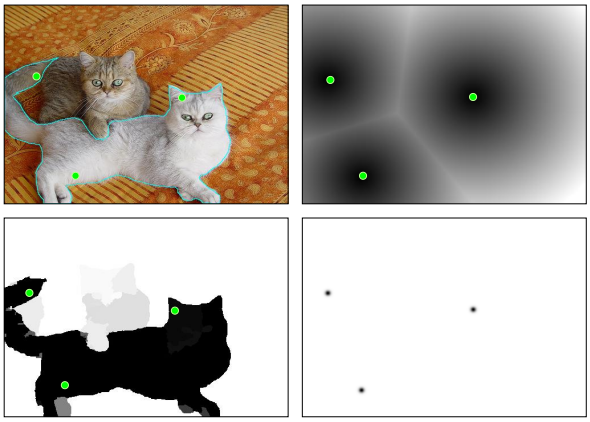


Figure 1. Existing interactive instance segmentation [26, 31, 32, 53] techniques do not utilize any image information when generating *guidance* maps (second column). In contrast, our proposed technique exploits image structures such as superpixels and object proposals, allowing us to generate more informative guidance maps (first column, bottom row).

User feedback is typically given as clicks [26, 31, 32, 35, 36, 53] or bounding boxes [52] and are then transformed into guidance signals fed as inputs into the CNN. By working with high-level representations encoded in pre-trained CNNs, the number of user interactions required to generate quality segments have been greatly reduced. However, there is still a large incongruence between the image encoding versus the guidance signal, as user interactions are transformed into simplistic primitives such as Euclidean [53, 31, 26] or Gaussian distance maps [6, 35, 36], the latter being the preferred choice in more recent works due to their ability to localize user clicks [35]. Examples of such guidance maps can be found in Fig. 1 second column, first row and second row respectively.

Our observation is that current guidance signals disregard even the most basic image consistencies present in the scene, such as colour, local contours, and textures. This of course also precludes even more sophisticated structures such as object hypotheses, all of which can be determined in an unsupervised way. As such, we are motivated to maximize the information which can be harnessed from user-provided clicks and generate more meaningful guidance maps for interactive instance segmentation.

To that end, we propose a simple yet effective transformation of user clicks which enables us to leverage a hierarchy of image information, starting from low-level cues such as appearance and texture, based on superpixels, to more high-level information such as class-independent object hypotheses (see Fig. 3). Ours is the first work to investigate the impact of guidance map generation for interactive segmentation. Our findings suggest that current Gaussian- and Euclidean distance based maps are too simple and do not fully leverage structures present in the image. A second and common drawback of current distance-based guidance maps is that they fail to account for the scale of the object during interaction. Object scale has a direct impact on the network performance when it comes to classification [41] or segmentation [40]. Gaussian- and Euclidean distance maps are primarily used for localizing the user clicks and do not account for the object scale. Our algorithm roughly estimates the object scale based on the user-provided clicks and refines the guidance maps accordingly.

Our approach is extremely flexible in that the generated guidance map can be paired with any method which accepts guidance as a new input channel [53, 32, 35, 6]. We demonstrate via experimentation that providing content-aware guidance by leveraging the structured information in an image leads to a significant improvement in performance when compared to the existing state-of-the-art, all the while using a simple, off-the-shelf, CNN architecture. The key contributions of our work are as follows :

- We propose a novel transformation of user-provided clicks which generates guidance maps by leveraging hierarchical information present in a scene.
- We propose a framework which can account for the scale of an object and generate the guidance map accordingly in a click-based user feedback scheme.
- We perform a systematic study of the impact of guidance maps on the interactive segmentation performance when generated based on features at different levels of the image hierarchy.
- We achieve state-of-the-art performance on four segmentation benchmarks; our proposed method significantly reduces the amount of user interaction required for accurate segmentation and uses the fewest number of average clicks per instance.

2. Related Works

Segmenting objects interactively using clicks, scribbles, or bounding boxes has always been a problem of interest in computer vision research, as it can solve some quality problems faced by fully-automated segmentation methods. Early variants of interactive image segmentation methods, such as the parametric active contour model [27] and intelligent scissors [39] mainly considered boundary properties when performing segmentation; as a result they tend to fare poorly on weak edges. More recent methods are based on graph cuts [7, 45, 49, 30], geodesics [5, 13], and or a combination of the two [19, 44]. However, all these algorithms try to estimate the foreground/background distributions from low-level features such as color and texture, which are unfortunately insufficient in several instances, *e.g.* in images with similar foreground and background appearances, intricate textures, and poor illumination.

As with many other areas of computer vision, deep learning-based methods have become popular also in interactive segmentation in the past few years. In the initial work of [53], user-provided clicks are converted to Euclidean distance transform maps which are concatenated with the color channels and fed as input to a FCN [34]. Clicks are then added iteratively based on the errors of the previous prediction. On arrival of each new click, the Euclidean distance transform maps are updated and inference is performed. The process is repeated until a satisfactory result is obtained. Subsequent works have focused primarily on making extensions with newer CNN architectures [35, 6] and iterative training procedures [35, 32]. In the majority of these works, user guidance has been provided in the form of point clicks [53, 35, 32, 36, 31] which are then transformed into a Euclidean-based distance map [53, 31]. One observation made in [6, 35, 36] was that encoding the clicks as Gaussians led to some performance improvement because it localizes the clicks better [35] and can encode both positive and negative click in a single channel [6]. In [9], the authors explore the use of superpixels to generate the guidance map.

However, in contrast to [9] which uses superpixels to maintain computational efficiency w.r.t. to their graph optimization, our guidance maps uses superpixels to leverage the local similarities contained within it. This is a general principle that we carry across image structures of varying levels for encoding user inputs. For the most part, there has been little attention paid to how user inputs should be incorporated as guidance; the main focus in interactive segmentation has been dedicated towards the training procedure and network architectures.

3. Proposed Approach

We follow previous interactive frameworks [53, 32, 35, 6] in which a user can provide both 'positive' and 'negative'

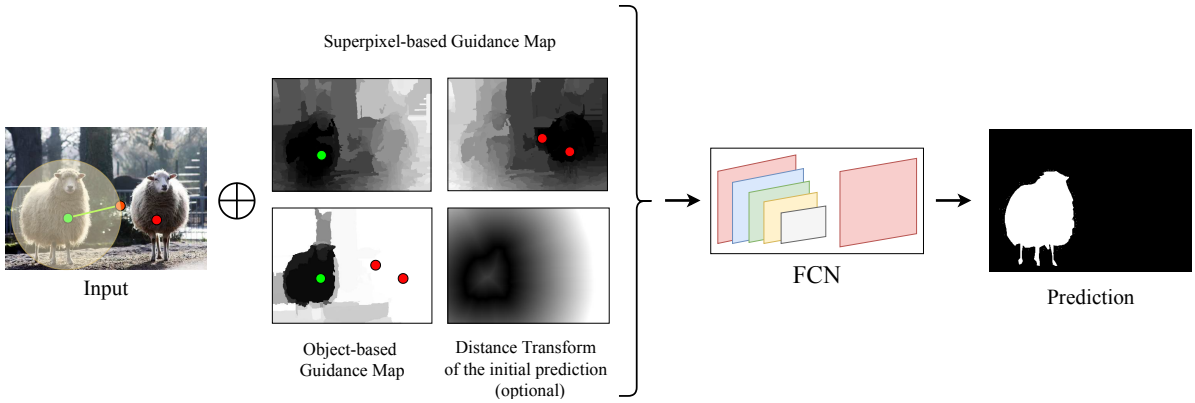


Figure 2. **Outline.** Given an input image and user interactions, we transform the positive and negative clicks (denoted by the green and red dots respectively) into three separate channels (2 channel superpixel-based and 1 object proposal-based guidance map), which are concatenated (denoted as $\oplus$) with the 3-channel image input and is fed to our network. Additionally, we concatenate the euclidean distance transform of the predicted mask from the previous iteration as our final non-color channel. The solid green line indicates our estimate of the object scale based on the initial pair of positive and negative click. The output is the ground truth map of the selected object.

clicks to indicate foreground and background/other objects respectively (as shown in Fig. 2). We denote the set of click positions as $\{\boldsymbol{p}_0, \boldsymbol{p}_1\}$ with subscripts 0 and 1 for positive and negative clicks respectively. To date, guidance maps have been generated by as a function of the distance between each pixel of the image grid to the point of interaction. More formally, for each pixel position $\boldsymbol{p}$ on the image grid, the pair of distance-based guidance maps for positive and negative clicks can be computed as

$$\mathcal{G}_0^d(\boldsymbol{p}) = \min_{\boldsymbol{c}\in\{\boldsymbol{p}_0\}} d(\boldsymbol{p}, \boldsymbol{c}) \quad \text{and} \quad \mathcal{G}_1^d(\boldsymbol{p}) = \min_{\boldsymbol{c}\in\{\boldsymbol{p}_1\}} d(\boldsymbol{p}, \boldsymbol{c}). \quad (1)$$

In the case of Euclidean guidance maps [53], the function $d(\cdot,\cdot)$ is simply the Euclidean distance.

However, such guidance is image-agnostic and assumes that each pixel in the scene is independent. Our proposed approach eschews this assumption and proposes the generation of multiple guidance maps which align with both low-level and high-level image structures present in the scene. We represent low-level structures with superpixels and high-level ones with region-based object proposals and describe how we generate guidance maps from these structures in Sections 3.1 and 3.2.

3.1. Superpixel-based guidance map

We first consider a form of guidance based on non-overlapping regions; in our implementation, we use superpixels. Superpixels group together locally similarly coloured pixels while respecting object boundaries [1] and were the standard working unit of pre-CNN-based segmentation algorithms [42, 17]. Previous works have shown that most, if not all, pixels in a superpixel belong to the same category [17, 42, 25]. Based on this observation, we propagate user-provided clicks which are marked on single pixels to the entire superpixel. We then assign guidance values to each of the other superpixels in the scene based on the minimum Euclidean distance from the centroid of each superpixel to the centroid of a user-selected superpixel. One can think of the guidance as a discretized version of Eq. 1 based on low-level image structures.

More formally, let $\{\mathcal{S}\}$ represent the set of superpixels from an image and $f_{SP}(\boldsymbol{p})$ be a function which maps each pixel location $\boldsymbol{p}$ in the image to the corresponding superpixel in $\{\mathcal{S}\}$. We further define a positive and negative superpixel set based on the positive and negative clicks, *i.e.* $\{\boldsymbol{s}_0 = f_{SP}(\boldsymbol{p}_0)\}$ and $\{\boldsymbol{s}_1 = f_{SP}(\boldsymbol{p}_1)\}$ respectively. Similar to the distance-based guidance maps in Eq. 1, we generate a pair of guidance maps. However, rather than treating each pixel individually, we propagate the distances between superpixel centers to all pixels within each superpixel, *i.e.*

$$\mathcal{G}_t^{\text{sp}}(\boldsymbol{p}) = \min_{s\in\{\boldsymbol{s}_t\}} d_c\left(s, f_{SP}(\boldsymbol{p})\right), \quad \text{where} \quad t = \{0, 1\}, \quad (2)$$

and $d_c(s_i, s_j)$ is the Euclidean distance between the centers s_i^c and s_j^c of superpixels s_i and s_j respectively, where $s_i^c = (\sum_i x_i/|s_i|, \sum_i y_i/|s_i|)$ where $|s_i|$ denotes the number of pixels within s_i. For consistency across training images, the guidance maps values are scaled between $[0, 255]$. When the user provides no clicks, all pixel values are set to 255. Examples guidance maps are shown in the second and third column of Fig. 3 respectively.

3.2. Object-based guidance map

Superpixels can be grouped together perceptually into category-independent object proposals. We also generate guidance maps from higher-level image structures, specifically region-based object proposals [3, 29, 37, 43, 47]. Such proposals have been used in the past as weak supervision

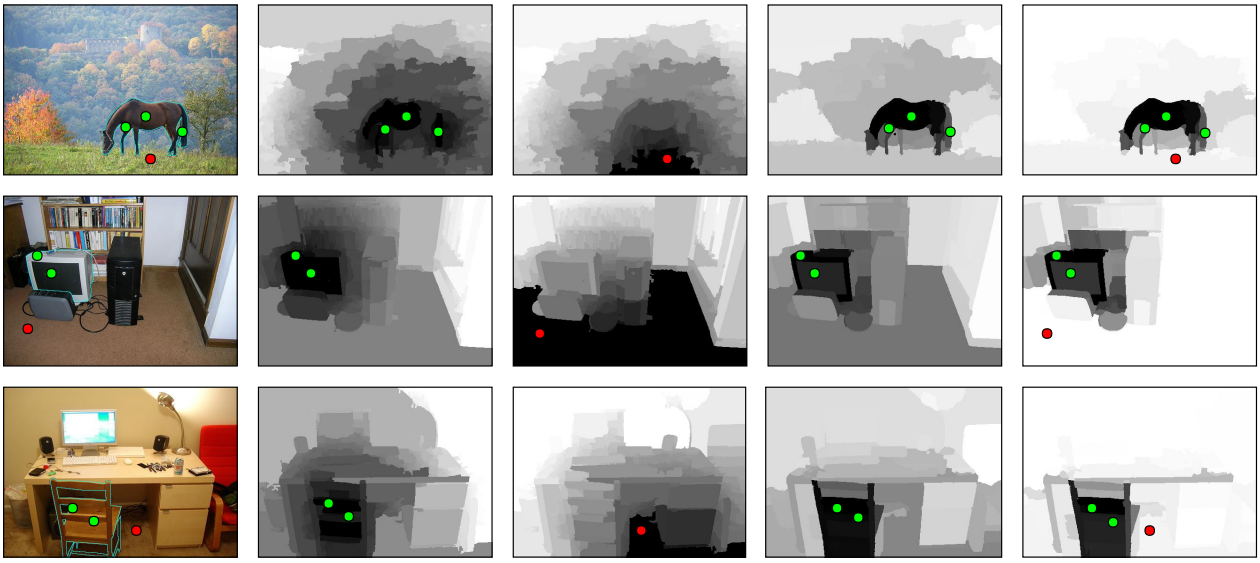


Figure 3. **Example of guidance maps**. We transform the user-provided positive (shown as green dots) and negative (shown as red dots) clicks into guidance maps for the instance segmentation network (columns 2 to 5). The second and third column correspond to the positive and negative superpixel based guidance map respectively. Examples of the object based guidance map and the scale-aware guidance map are shown in columns 4 and 5 respectively. For the clarity of visualization, we inverted the values of the object-based guidance map and the scale-aware guidance map (Best viewed in color).

for semantic segmentation [28, 14] and allow us to incorporate a weak object-related prior to the guidance map, even if the instance is not explicitly specified by the user-provided clicks. To do so, we begin with a set of object proposals [43], which have positive clicks its pixel support. For each pixel in the guidance map, we count the number of proposals from this set to which the pixel belongs. Pixels belonging to same object proposals are more likely to belong in the same object category and the number of proposals to which pixels belong incorporates a co-occurrence prior with respect to the current positive clicks.

More formally, let $\{\mathcal{L}_p\}$ be the set of object proposals for an image with support of pixel location $\boldsymbol{p}$. The object-based guidance map can be generated as follows:

$$\mathcal{G}^{\text{o}}(\boldsymbol{p}) = \sum_{\boldsymbol{p'} \in \{\boldsymbol{p}_0\}} \sum_{\mathcal{L} \in \{\mathcal{L}_{p'}\}} \mathbf{1}[\boldsymbol{p} \subset \mathcal{L}] \tag{3}$$

where $\mathbf{1}[\boldsymbol{p} \subset \mathcal{L}]$ is an indicator function which returns 1 if object proposal $\mathcal{L}$ has in its support or contains pixel $\boldsymbol{p}$. Similar to the superpixel-base guidance map, the object-based guidance is also re-scaled to $[0, 255]$. In the absence of user-provided clicks, all pixels are set to the value of 0. Examples are shown in the fourth column of Fig. 3.

3.3. Scale-aware guidance

Within an image, object instances can exhibit a large variation in their spatial extent [46]. While deep CNNs are known for their ability to handle objects at different scales [10], specifying the scale explicitly leads to an improvement in performance [41]. Interactive instance segmentation methods [36] which isolate the object tend to have a superior performance. For segmenting object instances, it is thus desirable to construct guidance maps which exhibit spatial extents consistent with the object.

A common limitation of most click-based interactive approaches is that the provided guidance is non-informative about scaling of the intended object instance. The commonly used forms of guidance are either too localized [35] (guidance map values are clipped to 0 at a distance of 20 pixels from the clicks) or non-informative [53].

Suppose now that we have some rough estimate of an object's scale in pixels, either in width or length. A convenient way to make our guidance maps scale-aware is to incorporate contributions of superpixels and object proposals which are in agreement with this scale. More specifically, we can apply this to the superpixel guidance map by truncating distances exceeding some factor f of our scale measure s, *i.e.*

$$\mathcal{G}_t^{\text{sp-sc}}(\boldsymbol{p}) = \min\left[\mathcal{G}_t^{\text{sp}}(\boldsymbol{p}), fs\right]. \tag{4}$$

We can apply similar constraints to the object-proposal based guidance by considering only the proposals within an accepted size range bounded by tolerance factors f_1 and f_2:

$$\mathcal{G}^{\text{o-sc}}(\boldsymbol{p}) = \sum_{\boldsymbol{p'} \in \{\boldsymbol{p}_0\}} \sum_{\mathcal{L} \in \{\mathcal{L}_{p'}\}} \mathbf{1}[\boldsymbol{p} \subset \mathcal{L}] \cdot \mathbf{1}[f_1 \leq |\mathcal{L}|/s^2 \leq f_2]. \tag{5}$$

3.4. Simulating user interactions

Even when selecting the same object instance, it is unlikely that different users will provide the same interactions inputs. For the model to fully capture expected behaviour across different users, one would need significant amounts of interaction training data. Rather than obtaining these clicks from actual users for training, we simply simulate user clicks and generate guidance maps accordingly.

We follow the sampling strategies proposed in [53]. For each object instance, we sample N_{pos} positive clicks within the object maintaining a distance d_1^{in} pixels from the object boundary and d_2^{in} pixels from each other. For negative clicks, we test the first two of the three sampling strategies outlined in [53], one in which N_{neg}^1 clicks are sampled randomly from the background, ensuring a distance of d_1^{out} pixels away from the object boundary and d_2^{out} pixels from each other and one in which N_{neg}^2 clicks on each of the negative objects (objects not of interest).

The above click-sampling strategy helps the network to understand notions such as negative objects and background but cannot train the network to identify and correct errors made during the prediction [35]. To this end, we also randomly sample N_{iter} clicks based on the segmentation errors. After an initial prediction is obtained, positive or negative clicks are randomly sampled from the error. Existing set of clicks are then replaced with the newly sampled clicks with a probability of 0.3. To mimic a typical user's behavior [35], the error-correction clicks are placed closest to the center of the largest misclassified region.

To estimate the scale measure s, we reserve the first two clicks, one positive and one negative, and assume that the Euclidean distance between the two is a roughly proportional measure; f, f_1 and f_2 are then set accordingly.

4. Experimental Validation

4.1. Datasets & Evaluation

We apply our proposed guidance maps and evaluate the resulting instance segmentations on four publicly available datasets: PASCAL VOC 2012 [16], GrabCut [45], Berkeley [38], and MS COCO [33].

PASCAL VOC 2012 consists of 1464 training images and 1449 validation images spread across 20 object classes.

GrabCut consists of 50 images with the corresponding ground truth segmentation masks and is a used as a common benchmark for most interactive segmentation methods. Typically, the images have a very distinct foreground and background distribution.

Berkeley consists of 100 images with a single foreground object. The images in this dataset represent the various challenges encountered in an interactive segmentation setting such as low contrast between the foreground and the background, highly textured background etc.

MS COCO is a large-scale image segmentation dataset with 80 different object categories, 20 of which are from the Pascal VOC 2012 dataset. For fair comparison with [53, 32], we randomly sample 10 images per category for evaluation and splitting the evaluation for the 20 Pascal categories versus the 60 additional categories.

Evaluation Fully automated instance segmentation is usually evaluated with mean intersection over union (mIoU) between the ground truth and predicted segmentation mask. Interactive instance segmentation is differently evaluated because a user can always add more positive and negative clicks to improve the segmentation and thereby increase the mIoU. As such, the established way of evaluating an interactive system is according to the number of clicks required for each object instance to achieve a fixed mIoU. Like [53, 32, 35, 6], we limit the maximum number of clicks per instance to 20. Note that unlike [53, 32], we do not apply any post-processing with a conditional random field and directly use the segmentation output from the FCN.

4.2. Implementation Details

Training As our base segmentation network, we adopt the FCN [34] pre-trained on PASCAL VOC 2012 dataset [16] as provided by MatConvNet [48]. The output layer is replaced with a two-class softmax layer to produce binary segmentations of the specified object instance. We fine-tune the network on the 1464 training images with instance-level segmentation masks of PASCAL VOC 2012 segmentation dataset [16] together with the 10582 masks of SBD [20]. We further augment the training samples with random scaling and flipping operations. We use zero initialization for the extra channels of the first convolutional layer (`conv1_1`). Following [53], we fine-tune first the stride-32 FCN variant and then the stride-16 and stride-8 variants. The network is trained to minimize the average binary cross-entropy loss.For optimization, we use a learning rate of 0.01 and stochastic gradient descent with Nesterov momentum with the default value of 0.9 is used.

Click Sampling We generate training images with a variety of click numbers and locations; sometimes, clicks end up being sampled from the same superpixel, which reduces training data variation. To prevent this and also make the network more robust to the click number and location for training, we sample randomly from the following hyperparameters rather than fixing them to single values: $N_{pos} = \{2, 3, 4, 5\}$, $N_{neg}^1 = \{5, 10\}$, $N_{neg}^2 = \{3, 5\}$, $d_1^{in} = \{15, 20, 40\}$, $d_2^{in} = \{7, 10, 20\}$, $d_1^{out} = \{15, 40, 60\}$, $d_2^{out} = \{10, 15, 25\}$. The randomness in the number of clicks and their relative distances prevents the network from over-fitting during training.

Guidance Dropout Since the FCNs are pre-trained on PASCAL VOC 2012, we expect the network to return a good initial prediction for images with object instances from one of its 20 classes. Thus, during training, when the network receives images without any instance ambiguity (*i.e*. an image with single object), we zero the guidance maps (value of 0 for object guidance map and 255 for the superpixel based guidance map) with a probability of 0.2 to encourage good segmentations without any guidance. We further increase robustness by resetting the positive or negative superpixel-based guidance with a probability of 0.4.

Interaction Loop During evaluation, a user provides positive and negative clicks sequentially to segment the object of interest. After each click is added, the guidance maps are recomputed; in addition the a distance transform of predicted mask from the previous iteration is provided as an extra channel [35]. The newly generated guidance map is concatenated with the image and given as input to the FCN-8s network which produces an updated segmentation map.

Superpixels & Object Proposals We use the implementation provided in [43] for generating superpixels; on average, each frame has $500 - 1000$ superpixels. For comparison, we also try other superpixelling variants *e.g*. SLIC [1] and CTF [54]. Although several object proposal algorithms exist [47, 8, 43], we use only MCG [43] as it has been shown to have higher quality proposals [14]. The final stage of MCG returns a ranking which we disregard. We use the pre-computed object proposals for PASCAL VOC 2012 and MS COCO provided by the authors of [43]. For GrabCut and Berkeley, we run MCG [43] on the *'accurate'* setting to obtain our set of object proposals.

4.3. Impact of Structure-Based Guidance

We begin by looking at the impact of superpixel based guidance. As a baseline, we compare with [53], which uses a standard Euclidean distance-based guidance as given in Eq. 1 (see examples in second row of Fig. 1). Similar to [53], we concatenate our positive and negative superpixel-based guidance maps with the three color channels and feed it as an input to the FCN-8s [34]. We use the superpixels computed using MCG [43]. For a fair comparison, we train our network non-iteratively, *i.e*., during training, we do not generate click samples based on the error in the prediction and do not append the distance transform of the current predicted mask as an extra channel. Looking at Table. 1, we see that our superpixel based guidance maps significantly reduce the number of clicks required to reach the standard mIoU threshold.

The object-based guidance provides the network with a weak localization prior of the object of interest. adding the object-based guidance with the superpixel based guidance leads to further improvements in performance (see third row of Table. 1). The impact is more prominent for datasets with a single distinct foreground object (*e.g*. 9.3% and 14% relative improvement for the Berkeley and GrabCut dataset). Finally, by making the feedback iterative, *i.e*. based on previous segmentation errors, we can further reduce the number of clicks. Overall, our structure-based guidance maps can reduce the number of clicks by 35% to 47% and unequivocally proves that having structural information in the guidance map is highly beneficial.

	GrabCut @90%	Berkeley @90%	VOC 2012 @85%
Euclidean ([53])	6.04	8.65	6.88
SP	4.44	6.67	4.23
SP + Obj.	3.82	6.05	4.02
SP + Obj. + Iter	**3.58**	**5.60**	**3.62**

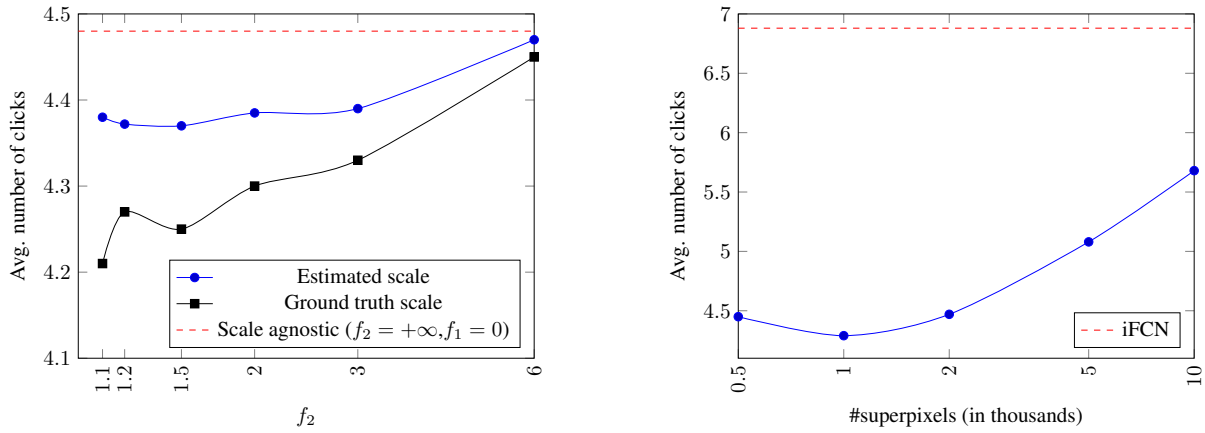
Table 1. Clicks required for different types of guidance. Guidance maps leveraging structural information require significantly less clicks than Euclidean distance-based guidance. *SP* refers to the superpixel guidance maps and *Obj* refers to the obect based guidance map and *Iter* refers to iterative training.

4.4. Impact of Scale-Aware Guidance

Due to fixed-size receptive field, FCNs experience difficulty when segmenting small objects [40]. The benefits of our scale-aware guidance map is most pronounced for segmenting small objects; for large objects (¿ 32×32 pixels), it does not seem to much effect. To highlight the impact of our guidance on small object instances, we pick the subset of 621 objects (from PASCAL VOC 2012) which are smaller than 32×32; objects smaller than this size are harder to identify [46].

In the scale agnostic setting, we consider all object proposals which has the click in its pixel support for generating the object-based guidance map, *i.e*. (as shown in Equation. 3; note that this is equivalent to having $f_1 = 0, f_2 = \infty$). Since the lower bound on scale has little effect, we set $f_1 = 0$. Looking at the average number of clicks required per instance to reach 85% mIoU for the subset of small objects (see Fig. 4 (a)), we find that having a soft scale estimate improves the network performance when it comes to segmenting smaller objects. This is primarily because the guidance map disregards object proposals which are not consistent in scale and can degrade the network performance by inducing a misleading co-occurrence prior.

When the scale s is based on ground truth (as the square root of the number of pixels in the mask foreground, see black curve in Fig. 4 (a)), the average clicks required per instance is consistently lower than the scale-agnostic case, even when as we relax f_2 up to 6, *i.e*. allowing for object



(a) Scale-Aware Guidance

(b) Number of superpixels

Figure 4. **(a) Scale-Aware Guidance.** The figure shows the average number of clicks required for segmenting small object instances (smaller than 32×32 pixels [46]) for varying degrees of tolerance till which we accept object proposals for generating our guidance map based on our estimated object scale and the ground truth object scale (computed as the square root of the number of pixels in the object mask). **(b) Number of superpixels.** The figure shows the average number of clicks required for segmenting object instances in PASCAL VOC 12 *val* set for different number of superpixels.

proposals which are 6 times larger than the actual object scale. Estimating the scale from the clicks is of course much less accurate than when it is take from the ground truth masks (compare black curve vs blue curve in Fig. 4 (a)). Nevertheless, even with such a coarse estimate, we find improvements in the number of clicks required as compared to the scale-agnostic scenario (compare red dashed line in Fig. 4 (a)). Given the first pair of positive and negative clicks, our estimated object scale is $\sqrt{\pi}d$ where d is the euclidean distance between the positive and the negative click. In our experiments, we observed that our estimated scale varies between $50 - 300\%$ from the ground truth scale). In comparison to a scale-agnostic setting, over the PASCAL VOC 2012 *val* set, we observe an improvement of 0.1 clicks on the small objects subset and an improvement of 0.032 clicks per instance for objects larger than 32×32 pixels.

Segmenting small objects with CNNs can be problematic [40]; we observed similar difficulties in preliminary experiments. For objects smaller than 32×32 pixels from PASCAL VOC 2012 *val* set, we require an average of 4.33 clicks which is significantly higher than our dataset average of 3.62 clicks.

4.5. Superpixels

Type of Superpixels To study the impact of the superpixeling algorithm, we consider the two variants SLIC [1] and CTF [54] and use only the superpixel based guidance map. On an average, MCG [43] generates $500 - 1000$ superpixels for each image in its default setting. For a fair comparison, we generate 500 and 1000 superpixels using SLIC and CTF. We observe that using 1000 SLIC superpixels results in performance similar to the MCG. However, irrespective of the superpixeling method, we found an overall improvement when the guidance maps are generated based on superpixels instead of pixel-based distances.

#superpixels	SLIC [1]	CTF [54]	MCG [43]
500	4.45	4.82	4.23
1000	4.29	4.58	

Table 2. Choice of superpixel algorithm

Number of Superpixels For this study, we consider only the superpixel-based map as guidance and use SLIC [1] as the superpixel algorithm. In the extreme case, all superpixels will have one pixel in its support and the guidance map degenerates to the Euclidean distance transform commonly used in existing interactive methods [53, 31]. We use the reported results in iFCN [53] on PASCAL VOC 2012 *val* set as our degenerate case (as shown by the red curve in Fig. 4 (b)). In addition to the reported results for 500 and 1000 superpixels on PASCAL VOC 2012 *val* set (as shown in Table 4 of the paper), we generate $2000, 5000$ and 10000 superpixels using SLIC [1]. We notice an initial gain in performance, but with increase in the number of superpixels, the performance drops as our network requires more and more clicks to segment the object of interest. As the number of superpixels increase, the benefits of local structure based grouping is lost as each superpixel is segmented into similar and redundant superpixels.

Method	Base Network	GrabCut @90%	Berkeley @90%	PascalVOC12 @85%	MS-COCO seen@85%	MS-COCO unseen@85%
iFCN [53]	FCN-8s [34]	6.04	8.65	6.88	8.31	7.82
RIS-Net [32]	DeepLab-LargeFOV [10]	5.00	6.03	5.12	5.98	6.44
ITIS [35]	DeepLabV3+ [12]	5.60	-	3.80	-	-
DEXTR [36]	DeepLabV2 [11]	4.00	-	4.00	-	-
VOS-Wild [6]	ResNet-101 [24]	3.80	-	5.60	-	-
FCTSFN [26]	Custom	3.76	6.49	4.58	9.62	9.62
IIS-LD [31]	CAN [55]	4.79	-	-	12.45	12.45
Ours	FCN-8s [34]	**3.58**	**5.60**	**3.62**	**5.40**	**6.10**

Table 3. The average number of clicks required to achieve a particular mIoU. The best results are indicated in **bold**.

4.6. Comparison to State of the Art

We compare the average number of clicks required to reach some required mIoU (see Table 3) against other methods reported in the literature. The methods vary in the base segmentation network from the basic FCNs to the highly sophisticated DeepLabV3 and also make use of additional CRF post-processing. We achieve the lowest number of clicks required for all datasets across the board, again proving the benefits of applying guidance maps based on existing image structures. We report results for our best trained SP+Obj+Iter network. To reach the mIoU threshold of 90% on GrabCut and Berkeley, our full model needs the fewest number of clicks as shown in Table 3 with a relative improvement of 5.79% and 7.13% over the current benchmark. For PASCAL VOC 2012 *val* set, we observe a relative improvement of 4.7%. For MS COCO, we observe a larger improvement for the 20 seen categories from PASCAL VOC 2012, as our networks were trained heavily on these object categories. Overall, we achieve an improvement of 9.7% and 5.28% over the 20 seen and 60 unseen object categories. We note that such an improvement is achieved despite the fact that our base network is the most primitive of the methods compared, *i.e*. an FCN-8s, in comparison to the others who use much deeper (ResNet-101) and more complex (DeepLabV3) network architectures. It should be noted that FCTSFN [26] and IIS-LD [31] report their result over all the 80 classes of MS COCO and not separately for 20 seen and 60 unseen classes.

We also compare our approach to that of [9]. [9] targets images with only a single foreground objects. To be comparable, we consider only our results with a single positive (foreground) click. We find that for the GrabCut and Berkeley dataset, our mIoU is higher by 4% and 8% respectively.

5. Discussion & Conclusion

In this work, we investigated the impact of the guidance maps for interactive object segmentation. Conventional methods use distance transform based approaches for generating guidance maps which disregard the inherent image structure. We proposed a scale aware guidance map generated using hierarchical image information which leads to significant reduction in the average number of clicks required to obtain a desirable object mask.

During experimentation, we observed that the object instances within the datasets varied greatly in difficulty. For instance, on PASCAL VOC 2012, the base network, without *any* user guidance, is able to meet the $\geq 85\%$ mIoU criteria for 433 of the 697 instances. Similarly observations were made for GrabCut ($\geq 90\%$ mIoU, 13 out of 50) and Berkeley ($\geq 90\%$ mIoU, 15 out of 100). On the other hand, we encountered instances where our algorithm repeatedly exhausted the 20 click budget regardless of sampled click locations and iterative feedback based on prediction errors. This is especially true for objects with very fine detailing, such as such as spokes in bicycle wheels, partially occluded chairs, *etc*. Based on these two extreme cases, we conclude that interactive segmentation is perhaps not so relevant for single object instances featuring prominently at the center of the scene and should feature more challenging scenarios. On the other hand, we need to design better algorithms which can handle objects that are not contiguous in region, *i.e*. has holes and are able to handle scenarios of occlusion. Depending on the target application, dedicated base architectures may be necessary to efficiently handle these cases.

Acknowledgement Research in this paper was partly supported by the Singapore Ministry of Education Academic Research Fund Tier 1.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012.

[2] Mykhaylo Andriluka, Jasper RR Uijlings, and Vittorio Ferrari. Fluid annotation: A human-machine collaboration interface for full image annotation. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1957–1966. ACM, 2018.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[4] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[5] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82(2):113–132, 2009.

[6] Arnaud Benard and Michael Gygli. Interactive video object segmentation in the wild. *arXiv preprint:1801.00269*, 2017.

[7] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001.

[8] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, (7):1312–1328, 2011.

[9] Ding-Jie Chen, Jui-Ting Chien, Hwann-Tzong Chen, and Long-Wen Chang. Tap and shoot segmentation. In *AAAI*, 2018.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2018.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2018.

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[13] Antonio Criminisi, Toby Sharp, and Andrew Blake. Geos: Geodesic image segmentation. In *ECCV*, 2008.

[14] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015.

[15] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[17] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.

[18] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In *MICCAI*, 2005.

[19] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010.

[20] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[21] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[22] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[25] Xuming He, Richard S Zemel, and Debajyoti Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006.

[26] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 2018.

[27] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[28] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[29] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *ECCV*, 2014.

[30] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 303–308. ACM, 2004.

[31] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, pages 577–585, 2018.

[32] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, 2017.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[35] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *BMVC*, 2018.

[36] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018.

[37] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *ECCV*, 2016.

[38] Kevin McGuinness and Noel E Oconnor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010.

[39] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *SIGGRAPH*. ACM, 1995.

[40] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[41] George Papandreou, Iasonas Kokkinos, and Pierre-André Savalle. Modeling local and global deformations in deep

learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *CVPR*, 2015.

[42] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[43] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *TPAMI*, 39(1):128–140, 2017.

[44] Brian L Price, Bryan Morse, and Scott Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR*. IEEE, 2010.

[45] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[46] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection–snip. In *CVPR*, 2018.

[47] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[48] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *MM*. ACM, 2015.

[49] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *Graphicon*, 2005.

[50] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine-tuning. *IEEE Transactions on Medical Imaging*, 2018.

[51] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L Divid, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *TPAMI*, 2018.

[52] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *BMVC*, 2017.

[53] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016.

[54] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *CVPR*, 2015.

[55] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.