Interactive Segmentation from 1-Bit Feedback

Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang

Department of Computer Science, National Tsing Hua University, Taiwan

Abstract. This paper presents an efficient algorithm for interactive image segmentation that responds to 1-bit user feedback. The goal of this type of segmentation is to propose a sequence of yes-or-no questions to the user. Then, according to the 1-bit answers from the user, the segmentation algorithm progressively revises the questions and the segments, so that the segmentation result can approach the ideal region of interest (ROI) in the mind of the user. We define a question as an event that whether a chosen superpixel hits the ROI or not. In general, an interactive image segmentation algorithm is better to achieve high segmentation accuracy, low response time, and simple manipulation. We fulfill these demands by designing an efficient interactive segmentation algorithm from 1-bit user feedback. Our algorithm employs techniques from over-segmentation, entropy calculation, and transductive inference. Over-segmentation reduces the solution set of questions and the computational costs of transductive inference. Entropy calculation provides a way to characterize the query order of superpixels. Transductive inference is used to estimate the similarity between superpixels and to partition the superpixels into ROI and region of uninterest (ROU). Following the clues from the similarity between superpixels, we design the query-superpixel selection mechanism for human-machine interaction. Our key idea is to narrow down the solution set of questions, and then to propose the most informative question based on the clues of the similarities among the superpixels. We assess our method on four publicly available datasets. The experiments demonstrate that our method provides a plausible solution to the problem of interactive image segmentation with merely 1-bit user feedback.

1 Introduction

Image segmentation is a building block of many applications in computer vision and image editing. It is typically used to partition images into several regions and thus to enable the subsequent high-level processing about image structure. However, the regions of interest may have semantic significance, or just have certain homogeneity. Since it is hard to define the region of interest selected by a human user, interactive segmentation provides a solution to this dilemma by invoking the aid of the user. In interactive segmentation, the user marks areas of the image as ROI or ROU, then the segmentation algorithm updates the segmentation according to the new marked areas. By iteratively providing more new marked areas, the user can guide the segmentation toward the ROI she or he prefers.

In interactive image segmentation, there are several types of the manipulation mechanisms for human-machine interactions, containing varying degrees of complexity. To a machine, a user can guide the segmentation by providing seed points [3], region selections ([4, 14, 20]), line segments [5, 7, 11, 12], bounding boxes [6, 17], contours [13, 16], image set [18, 21] and so on. There is no doubt that the high segmentation accuracy is the basic criterion of a segmentation algorithm. However, the user may be more sensitive to the manipulation mechanism and the response time of a segmentation algorithm. The manipulation mechanism is the aforementioned inputs from the user. The response time is the time a segmentation algorithm takes to react to a given new input. An interactive segmentation algorithm with simpler manipulation and lower response time makes a user more willing to interact with the algorithm.

Recently, Rupprecht *et al.* [19] introduce one kind of interactive image segmentation with very simple manipulation. In this kind of segmentation, the user only needs to decide whether the pixel queried by the machine hits the ROI or not. Their idea is inspired from the classical *Twenty Questions game*[1]. Twenty questions game is a kind of deductive questioning game with multiple players. One player, called *oracle*, selects an object in mind; the other players, called the *inquirers*, try to infer the selected object with a limited amount of questions. Each kind of player obeys one rule to play the twenty questions game. The rule for the oracle is that only 'yes' or 'no' can be used to response to inquirer; the rule for an inquirer is that only the questions that can be answered with 'yes' or 'no' are allowed to be asked. The twenty-questions-style interactive image segmentation proposed by Rupprecht *et al.* can be thus defined as follows. A user, playing the role as the oracle, chooses an ROI in a given image. Then the segmentation algorithm, i.e. the inquirer, proposes one pixel each round to query the user's response to fact that the pixel hits the ROI or not. The algorithm's goal is to infer the ROI in the user's mind with the information from those queried pixels. Since the user only provides 'yes' or 'no' for each question, we call the interaction as *1-bit feedback*. This is the simplest manipulation mechanism for an interactive image segmentation. In this paper, we focus on addressing the interactive image segmentation with 1-bit user feedback.

The interactive segmentation from 1-bit feedback has potential to provide a hands-free segmentation mechanism, because the users do not need to provide any scribbles on specific image locations. For example, in sterilized operating room, the physically touched computer control for medical image segmentation is inappropriate. Or, for instance, tiny screens on the wearable computers have limited interface capabilities. In such scenarios, 1-bit user feedback are more adequate since any input device that receives binary signals can be used to collect the responses.

This paper describes an efficient algorithm to address the problem of interactive image segmentation with merely 1-bit user feedback. There are three advantages of the proposed interactive image segmentation method. First, the proposed method has a very simple human-machine interaction mechanism. In

[1] An online Twenty Questions game web site: $http://www.20q.net/$

our implementation, user only needs to decide if the proposed superpixel hits the ROI or not. Second, the segmentation accuracy of our method is better than the competitor [19] in most datasets. Third, the average response time of our method is extreme short.

2 Related Work

The purpose of an interactive segmentation algorithm is to segment the region of interest in an image with the aid from the user. We briefly categorize some selected methods into two groups according to their interaction modes.

2.1 Passive Interaction Based Image Segmentation.

In passive interaction based image segmentation [3, 5–7, 11–13, 16–18, 21], an interaction is triggered by the inputs from the user. The user directly defines various inputs to guide the machine to approach the segmentation of ROI.

In general, these algorithms allow the user to specify scribbles via seed points [3], line segments [5, 7, 11, 12], bounding boxes [6, 17], or contours [13, 16]. Then, the segmentation algorithms minimize their energy functions using *level set*, *graph cuts*, *random walks*, or *geodesic distance* to segment the images. The segmentation results are updated according to the modified scribbles. Image co-segmentation [18, 21] is a special case of interactive segmentation, which provides a way to implicitly define the region of interest via multiple images. The segmentation results may be different according to the different image sets.

To sum up, the manipulation methods of the passive interaction based image segmentation usually contain varying degrees of complexity. Different scribbles may get very different segmentation results. The user needs to take full responsibility on how to specify good scribbles to guide the segmentation algorithm.

2.2 Active Interaction Based Image Segmentation.

Methods in the category of active interaction based image segmentation [4, 14, 20] usually propose several uncertain image regions to the user, and then the computer updates the segmentation results with the user selected regions.

To segment a lot of images simultaneously, Batra *et al.* [4] propose a co-segmentation algorithm which allows users to indicate where the ROI is. Based on the user indications, their system provides some uncertain image regions to ask for the real labels from the user. Kowdle *et al.* [14] propose an active-learning based segmentation algorithm for 3D reconstruction built on an energy minimization framework. They employ an active learning method to query whether the uncertain regions belong to ROI or not. Straehle *et al.* [20] provide non-local uncertainty measurements to suggest the uncertain regions for the user, and then apply the watershed cut to segment the large data sets guided by the user-selected locations. Rupprecht *et al.* [19] simulate the segmentation probability among the entire image by sampled segmentations. According to the probability

distribution over the set of sampled segmentations, they pick the region that has the highest uncertainty to ask for the real label.

Comparing with the passive interaction based image segmentation, the active interaction based image segmentation has simpler human-machine interaction but unsatisfactory segmentation accuracy. This kind of segmentation algorithm needs to estimate the segmentation uncertainty of the entire image, and based on the estimation, it can query the user for reducing the segmentation uncertainty.

2.3 Yes-or-no Interaction.

The most similar work to ours is the algorithm proposed by Rupprecht *et al.* [19]. Their segmentation model is based on the notion of the twenty questions game. In twenty questions game, the effective strategy for the inquirer is to come up with a question that can eliminate half the *all possible answers* at each iteration. In this way, a series of 20 questions can allow the inquirer to distinguish between 2^{20} potential answers. Based on this strategy, Rupprecht *et al.* use the MCMC sampling method to approximate the probability of *all possible segmentations*. Since the probability distributed among the entire image, they directly select the centroid pixel of the most uncertain region as the question.

In contrast to their method, we first reduce the solution set of all possible segmentations by over-segmentation. Then, we use transductive inference technique to directly explore the most informative superpixel for querying. Finally, we update the segmentation from the user feedback to approach the ROI in the user's mind. Fig. 1 illustrates the outline of our approach.

We do the experiments on four datasets: Berkeley segmentation dataset (BSDS300) [9], Stanford Background Dataset (SBD) [10], Microsoft Research Asia Salient Object Dataset (MSRA1000) [1], and The PASCAL Visual Object Classes Challenge 2007 (VOC2007) [8]. The experimental results show that our method performs better than the recent approach [19] in response time and segmentation accuracy.

3 Our Approach

The proposed segmentation algorithm includes three phases: *Initialization*, *Adaptive Querying*, and *Information Updating*. The initialization phase aims to reduce the solution set of all possible segmentations. In addition, this phase also prepares the needed graph structure and feature descriptors for the subsequent phases. The adaptive querying phase and the information updating phase carry out interactive image segmentation with the aid of 1-bit feedback from the user. The adaptive querying phase adopts the transductive inference technique to explore the most informative superpixel for querying. Based on the new feedback, the information updating phase improves the segmentation uncertainty and thus pushes the segmentation toward the ROI that the user prefers.

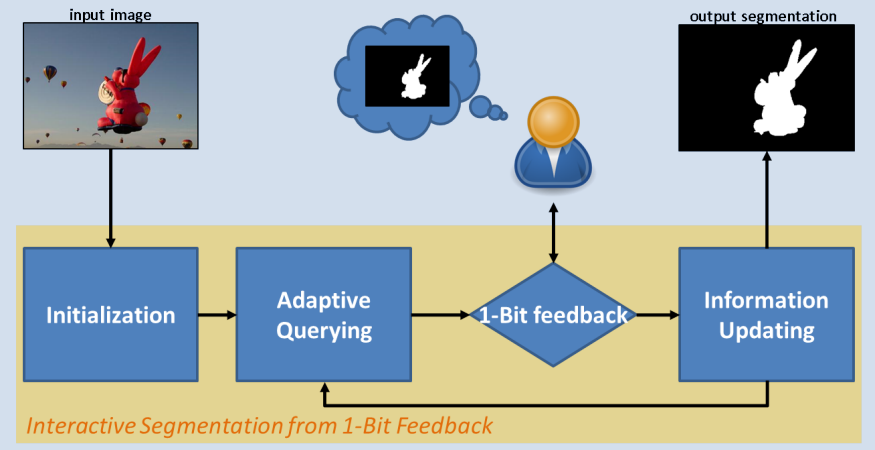


Fig. 1. The interaction pipeline of our method. The initialization phase aims to reduce the solution set of all possible segmentations. The adaptive querying phase and the information updating phase carry out interactive image segmentation with the aid of 1-bit feedback from the user. The adaptive querying phase adopts the transductive inference technique to explore the most informative superpixel for querying. Based on the user feedback, the information updating phase updates the needed information for next iteration.

3.1 Initialization

Given an image with height h and width w, the pixel-wise binary segmentation has $2^{h\times w}$ possible configurations. However, the discriminative power of twenty iterations in 1-bit feedback can only distinguish 2^{20} possible segmentations at most. Hence, our first goal is to reduce the solution set of all possible segmentations. One reasonable method is to over-segment the image into a superpixel set $\mathcal{S}$, thus the solution set could be greatly reduced to $2^{|\mathcal{S}|}$. That is the motivation we propose to consider superpixel-wise image segmentation in this work. In order to apply the transductive inference technique to estimate the similarity between any two superpixels, we need to model the given image as a graph for describing the neighborhood relationship among superpixels. Therefore, we first describe the steps of building the graph model of an image in this subsection, and then explain how to estimate the similarity between any two superpixels via the transductive inference technique.

Graph Model. For the given image, we use the efficient SLIC over-segmentation algorithm [2] to partition the image into a superpixel set $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ with N elements. For each superpixel, we use the 3-D mean color in CIE-Lab space as its feature representation.

Given a superpixel set $\mathcal{S}$, we define a weighted connected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \omega)$, where the vertex set $\mathcal{S}$ contains all image superpixels and the edge set $\mathcal{E}$ consists

all pairs of any two adjacent superpixels. Precisely, each vertex s_p denotes a single superpixel, and each edge $e_{pq} \in \mathcal{E}$ denotes the adjacent neighborhood of superpixels s_p and s_q. The weighting function $\omega : \mathcal{E} \rightarrow [0, 1]$ assigns the corresponding weight ω_{pq} to each edge e_{pq}, expressed in terms of mean color feature similarities. We can thus define the N-by-N weight matrix as $W = [\omega_{pq}]_{N \times N}$.

Similarity Estimation. The weight matrix W describes the similarity between any two *adjacent* superpixels. With the transductive inference method proposed by Zhou *et al.* [22], we can further estimate the transductive similarity between any two superpixels, no matter they are adjacent or not. The transductive similarity matrix T also has size N-by-N, and can be defined by

$$T = (D - \alpha W)^{-1} I \ , \tag{1}$$

where D is the diagonal matrix with each diagonal entry representing the row sum of W, α is a parameter in $(0, 1)$, and I is the N-by-N identity matrix.

3.2 Adaptive Querying

In the scenario of interactive image segmentation from 1-bit feedback, the goal of the segmentation algorithm is to guess the user's ROI. However, a region of interest may have semantic significance, or just have certain homogeneity. Fig. 2 shows the diverse ground-truth segments of different datasets. In this paper, we set the goal of the adaptive querying phase as to explore the most informative superpixel for querying. We also design two strategies to deal with two cases existing in this phase. The two cases are categorized according to whether we obtain i) only one of the ROI-superpixel or ROU-superpixel, or ii) both of the ROI-superpixel and ROU-superpixel.

The first case corresponds to the situation of all queried superpixels belonging to the ROI or the ROU. In this case, the boundary of ROI is very difficult to define by transductive inference method. The first priority for this case is to find out the other label so that the second case can be applied. The second case corresponds to the situation of some queried superpixels belonging to the ROI and some queried superpixels belonging to the ROU. In this case, the boundary of ROI can be roughly described by transductive inference method. Now the goal is to find out the boundary superpixel with the most uncertainty to refine the ROI.

Case 1: Only one label is available. In this case, we aim to find out the other label. The most informative superpixel is defined as the superpixel that has the highest entropy so far. In this work, we diversify our queries using the entropy and the transductive similarity matrix T from Eq. 1. In transductive similarity matrix T, the n-th row represents the similarity between superpixel s_n and all other superpixels. Here we normalize the n-th row of T to make it sum to one. We

observe that if a superpixel s_n has more similar superpixels, then the normalized n-th row of T is more flattened. Hence we adopt the entropy function of n-th row of T to represent the proportion of similar superpixels that the superpixel s_n has. The higher entropy that a superpixel has also indicates that the superpixel is more informative, because no matter which label the superpixel has, there are large proportion of similar superpixels contain that label. Therefore, we choose the superpixel with the highest entropy as the query-superpixel in this case.

In practice, we define the query-superpixel selection function Q_1 in Case 1 by

$$Q_1(\mathcal{S}) = \arg\max_{s_n \in \mathcal{S}} \epsilon(s_n) = \arg\max_{s_n \in \mathcal{S}} \epsilon(T_{s_n,\cdot}) \ , \tag{2}$$

where $\epsilon(\cdot)$ is the entropy function, $T_{s_n,\cdot}$ is the normalized n-th row of T.

Case 2: Both labels are available. In this case, we aim to refine the ROI boundary. The most informative superpixel is defined as the superpixel which has the most uncertainty. We simulate the segmentation uncertainty with transductive inference and the known labels, and thus select the most uncertain superpixel to form the query question.

In practice, we use the following N-by-2 transductive similarity matrix to describe the similarity between each superpixel to ROI or ROU:

$$\hat{T} = (D - \beta W)^{-1} Y \ , \tag{3}$$

where β is a parameter in $(0, 1)$, $Y = [y_{ROI}, y_{ROU}]$ is a label matrix, y_{ROI} and y_{ROU} are both N-by-1 indicator vectors, in which the n-th element is 1 if the n-th superpixel has label ROI or ROU. The first column of $\hat{T}$ indicates the similarity of each superpixel to all ROI-superpixels, the second column of $\hat{T}$ indicates the similarity of each superpixel to all ROU-superpixels. The superpixel that has the highest uncertainty will have the smallest difference between these two columns. Hence, we define the query-superpixel selection function Q_2 in Case 2 by

$$Q_2(\mathcal{S}) = \arg\min_{s_n \in \mathcal{S}} \delta(s_n) = \arg\min_{s_n \in \mathcal{S}} |\hat{T}_{s_n,1} - \hat{T}_{s_n,2}| \ , \tag{4}$$

where $\delta(\cdot)$ is the absolute difference function, $\hat{T}_{p,q}$ is the entry of $\hat{T}$ that locates on p-th row and q-th column.

3.3 Information Updating

In the scenario of interactive image segmentation from 1-bit feedback, we can receive the 1-bit feedback between adaptive querying phase and the information updating phase. With one more certain label of the query-superpixel, we first generate the corresponding segmentation, and then update the needed information $\epsilon(s)$ and $\hat{T}$ for next iteration. Note that if the new feedback defines the counterpart label of Case 1, then we go into Case 2 and thus only need to update $\hat{T}$.

Case 1: Still only one label is available. To get the corresponding segmentation from Eq. (2), we define the current maximum entropy value among all superpixels $\{\epsilon(s_1), \epsilon(s_2), \cdots, \epsilon(s_N)\}$ as $\mathfrak{m}$. A superpixel with larger entropy value than $\mathfrak{m}/2$ is treated as an ROI-superpixel.

For preventing from selecting the high entropy superpixel that is similar to the one in the previous iteration, we have to alter the entropy value among all superpixels with the latest queried superpixels s_z. Here we define the new entropy value of superpixel s_n as

$$\epsilon(s_n) = \epsilon(s_n) - \frac{T_{s_z,s_n}}{T_{s_z,s_z}}\epsilon(s_z) \ . \tag{5}$$

This updated $\epsilon(s_n)$ will trigger the query-superpixel selection function Q_1 in Eq. (2).

Case 2: Both labels are available. To get the corresponding segmentation from Eq. (3), a superpixel with positive value of $\hat{T}_{s_n,1} - \hat{T}_{s_n,2}$ is treated as an ROI-superpixel.

Since we have one more new label, we have new indicator vector y'_{ROI} or y'_{ROU}. Hence we obtain new label matrix $Y' = [y'_{ROI}, y_{ROU}]$ or $Y' = [y_{ROI}, y'_{ROU}]$. This update, $Y = Y'$, will trigger the new transductive similarity matrix $\hat{T}$ in Eq. (3) and the new query-superpixel in Eq. (4).

4 Experimental Results

Since we deal with the interactive image segmentation problem, we aim to achieve faster response time, higher segmentation accuracy, and fewer queries. Hence we conduct the evaluations with respect to the response time and the qualitative and quantitative results. The experiments are performed on four datasets. The parameter settings are the same for the four datasets, we set the number of superpixels $N = 350$, the parameters $\alpha = 0.999$ and $\beta = 0.001$.

To evaluate our method on these datasets, every segment in each individual ground-truth annotation is selected as a region of interest (ROI). To measure the segmentation quality, we employ the median and the mean Dice scores as used in Rupprecht *et al.* [19], which measure the overlap between the segmentation and the ground truth. We separately compare the proposed interactive image segmentation with its several variants and the method of Rupprecht *et al.* [19] in following experiments.

Berkeley Segmentation DataSet (BSDS300) [9]: The BSDS300 dataset contains 300 natural images. Each image has several hand-labeled segmentations as the ground-truth human annotations. The example ground-truth of BSDS300 is shown in Fig. 2a. Note that we only show one human annotation for better visualization. This dataset contains various regions with several human annotators, thus provides difficult tasks of identifying regions of interest of a interactive segmentation from 1-Bit Feedback algorithm. The average number of regions in each individual ground-truth is 20.37.

(a) BSDS300 (b) SBD (c) MSRA1000 (d) VOC2007

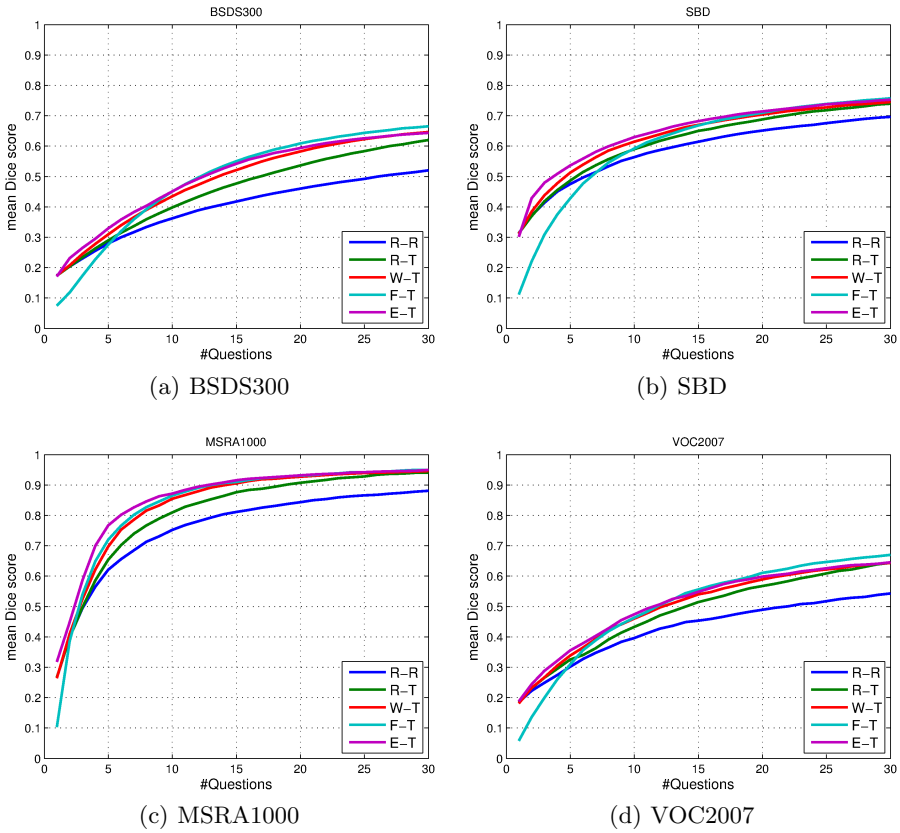
Fig. 2. Testing examples from each dataset. Each color denotes an ground-truth segment. Black color in (b), (c), and (d) and creamy-white color in (d) are ignored segments in the experiments.

Stanford Background Dataset (SBD) [10]: The SBD dataset contains 715 natural images collected from other datasets. This dataset contains three types of ground-truth annotations. We select the *region* annotation as Rupprecht *et al.*[19] for comparison. Each image in the SBD dataset has one ground-truth human annotation. This dataset contains some semantic regions. Precisely, each image has eight possible semantic labels: building, foreground object, grass, mountain, road, sky, tree, and water. The example ground-truth of SBD is shown in Fig. 2b. The average number of semantic regions in each individual ground-truth is 4.22.

Microsoft Research Asia Salient Object Dataset (MSRA1000): The MSRA1000 dataset contains 1000 natural images collected from other datasets. The ground-truth human annotations are provided by Achanta *et al.* [1][2]. The natural images are provided by Liu *et al.* [15][3]. Each image has only one ground-truth human annotations. The example ground-truth of MSRA1000 is shown in Fig. 2c. This dataset contains only the ROI region, thus provides clearly-defined region of interest. The average number of regions in each individual ground truth is thus 1.0.

The PASCAL Visual Object Classes Challenge 2007 (VOC2007) [8]: From the VOC2007 dataset, we use the *trainval* data in segmentation subset for evaluation. The *trainval* image set in segmentation subset contains 422 natural images. This dataset also contains some semantic regions. Precisely, each image can has twenty possible semantic labels. Each image is partitioned into independent instances. The example ground-truth of VOC2007 is shown in Fig. 2d. The average number of regions in each individual ground-truth is 2.87.

[2] *http://ivrlwww.epfl.ch/supplementary_material/RK_CVPR09/index.html.*
[3] *http://research.microsoft.com/en−us/um/people/jiansun/SalientObject/salient_object.htm.*



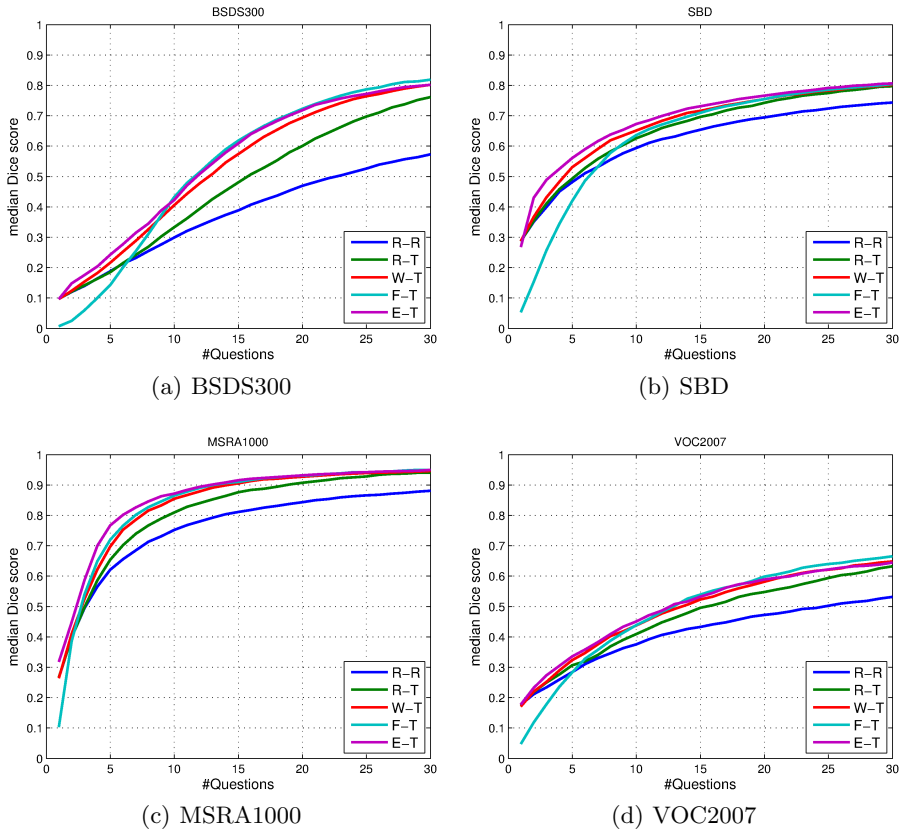
(a) BSDS300 (b) SBD

(c) MSRA1000 (d) VOC2007

Fig. 3. Performance comparison of the variant versions of our method on the mean Dice score against the number of questions.

4.1 Variants

This experiment compares some variant versions of our method. Fig. 3 and Fig. 4 depict the mean Dice score and the median Dice score against the number of questions. In the legend blocks of these two figures, we use strategy1-strategy2 to represent the variant versions of our method. Specifically, strategy1 denotes the strategy used in Case 1 and strategy2 denotes the strategy in used Case 2. In addition, 'R' means selecting a random superpixel; 'T' means selecting the most uncertain superpixel according to the transductive inference similarity; 'W' means selecting a random superpixel weighted by its entropy value; 'F' means selecting the superpixel which has the most different feature to all previous selected superpixels; 'E' means selecting a superpixel with the highest entropy value. A dataset with a larger difference between Fig. 3 and Fig. 4 means the dataset is more difficult to do the interactive segmentation, because the median is helpful in ignoring several bad segmentations.



(a) BSDS300 (b) SBD

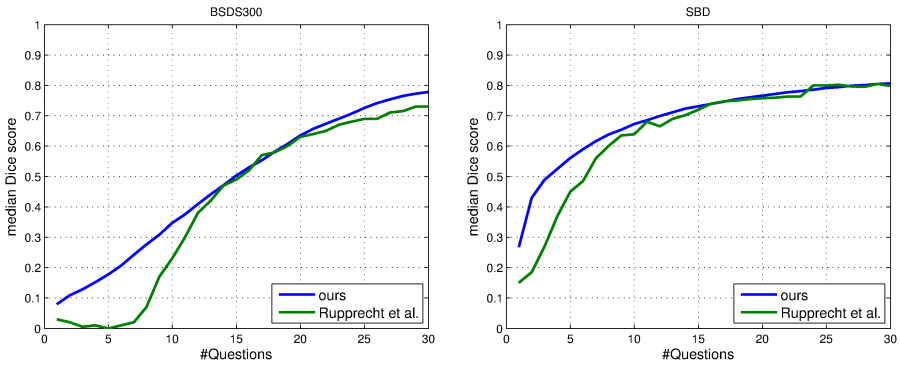
(c) MSRA1000 (d) VOC2007

Fig. 4. Performance comparison the variant versions of our method on the median Dice score against the number of questions.

It is better to select the superpixel with the highest entropy value as the query-superpixel in the first case of the adaptive querying phase, because it contains the most information. On the other hand, it is better to select the most uncertain superpixel according to the transductive inference similarity in the second case of the adaptive querying phase, because it can find out the superpixel that locates on the object's boundary, and thus is better for segmentation refinement. Therefore, we select the 'E-T' version as our representative method in following experiments.

4.2 Comparison

Fig. 5 illustrates the median Dice score against the number of questions. Since there is no released code of the method of Rupprecht *et al.*, we reproduce their results on Fig. 5 according to their experiments in [19]. It can bee seen that our method performs significantly better than the method of Rupprecht *et al.*



(a) BSDS300 (b) SBD

Fig. 5. Performance comparison on the median Dice score against the number of questions.

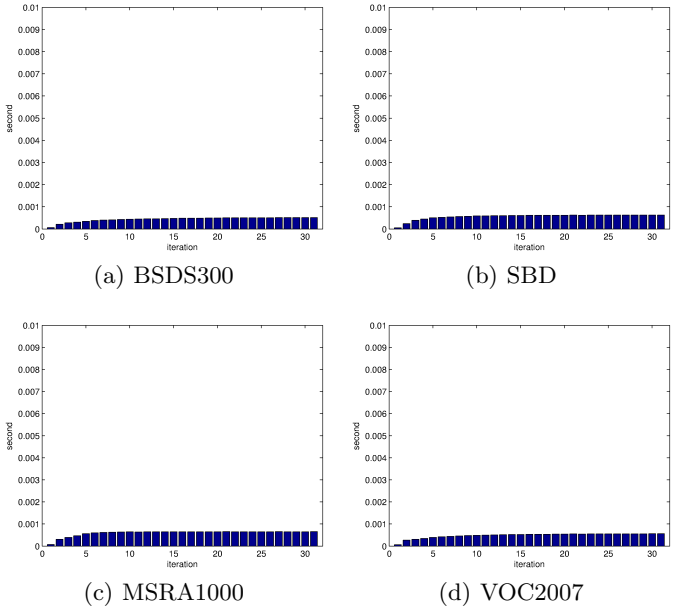
Table 1. Quantitative comparison: The mean and median Dice scores (%).

BSDS300	10Q		20Q		30Q	
	mean	median	mean	median	mean	median
Rupprecht *et al.* [19]	34.7	23.8	48.8	62.0	56.1	73.2
Ours	**40.3**	**34.7**	**55.6**	**63.5**	**63.3**	**77.8**
SBD	10Q		20Q		30Q	
	mean	median	mean	median	mean	median
Rupprecht *et al.* [19]	52.6	63.9	63.9	75.8	67.9	79.8
Ours	**61.4**	**65.4**	**70.9**	**76.1**	**75.0**	**80.5**

in BSDS300. In SBD, our method is much better than their method in the first fifteen rounds. Table. 1 also shows that our method performs quite well. This experiment shows that our method is very competitive with respect to the criterion of segmentation accuracy.

4.3 Response Time

This experiment shows the efficiency of our method. In general, the average response time per iteration for our method is less than 1 millisecond on an Intel Core i7-4770 3.40 GHz CPU. For comparison, the average response time of the method of Rupprecht *et al.* is about 1 second on Intel Core i7-4820 3.70 GHz CPU. Their computation bottleneck is the step of MCMC sampling, which is used to approximate the segmentation probability among the entire image. In contrast, we directly use the transductive inference technique to explore the most informative superpixel for querying, thus prevent the complex sampling process. Another reason is because we use superpixels as the building blocks of our algorithm. Using superpixels greatly reduces the number of graph nodes and speeds up the computation in transductive inference. Notice that it takes



(a) BSDS300 (b) SBD

(c) MSRA1000 (d) VOC2007

Fig. 6. Average response time per iteration of our approach on the four datasets.

about 0.2 second in the initialization phase, which contains over-segmentation, feature extraction, and transductive inference. However, the initialization phase only needs to be done once. Fig. 6 shows that our average time per iteration on the four datasets.

5 Conclusion

We have shown that the proposed method can efficiently solve the problem of interactive image segmentation from 1-bit feedback. Our interactive image segmentation algorithm achieves the preferable properties of high segmentation accuracy, low response time, and simple manipulation. We fulfill these requirements by designing an efficient algorithm that consists of over-segmentation, entropy estimation, and transductive inference. The experimental results show the good performance of our method, in particular, extremely short response time. Our key idea is to prune the solution space of possible segmentations, and then to propose the most informative question based on the clues of the similarity among the superpixels. The method helps to increase the probability of finding out the most uncertain image region to obtain its real label from the 1-bit user feedback.

Acknowledgement. This work was support in part by MOST Grants 103-2221-E-007-045-MY3 and 103-2218-E-007-017-MY3 in Taiwan.

References

1. Achanta, R., Hemami, S.S., Estrada, F.J., Süsstrunk, S.: Frequency-tuned salient region detection. In: CVPR. (2009) 1597–1604
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34** (2012) 2274–2282
3. Adams, R., Bischof, L.: Seeded region growing. IEEE Trans. Pattern Anal. Mach. Intell. **16** (1994) 641–647
4. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: icoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR. (2010) 3169–3176
5. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV. (2001) 105–112
6. Cheng, M., Prisacariu, V.A., Zheng, S., Torr, P.H.S., Rother, C.: Densecut: Densely connected crfs for realtime grabcut. Comput. Graph. Forum **34** (2015) 193–201
7. Dong, X., Shen, J., Shao, L., Yang, M.: Interactive cosegmentation using global and local energy optimization. IEEE Trans. Image Processing **24** (2015) 3966–3977
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
9. Fowlkes, C.C., Martin, D.R., Malik, J.: Local figure-ground cues are valid for natural images. Journal of Vision **7** (2007) 1–9
10. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV. (2009) 1–8
11. Grady, L.: Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 1768–1783
12. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: CVPR. (2010) 3129–3136
13. Kass, M., Witkin, A.P., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1988) 321–331
14. Kowdle, A., Chang, Y., Gallagher, A.C., Chen, T.: Active learning for piecewise planar 3d reconstruction. In: CVPR. (2011) 929–936
15. Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H.: Learning to detect A salient object. In: CVPR. (2007)
16. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: SIGGRAPH. (1995) 191–198
17. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23** (2004) 309–314
18. Rother, C., Minka, T.P., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In: CVPR. (2006) 993–1000
19. Rupprecht, C., Peter, L., Navab, N.: Image segmentation in twenty questions. In: CVPR. (2015) 3314–3322
20. Straehle, C.N., Köthe, U., Knott, G., Briggman, K.L., Denk, W., Hamprecht, F.A.: Seeded watershed cut uncertainty estimators for guided interactive segmentation. In: CVPR. (2012) 765–772
21. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: CVPR. (2011) 2217–2224
22. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS. (2003) 321–328