Crowdsourced Object Segmentation with a Game

Amaia Salvador
IRIT-ENSEEIHT
University of Toulouse
Toulouse, France

Axel Carlier
IRIT-ENSEEIHT
University of Toulouse
Toulouse, France
axel.carlier@enseeiht.fr

Xavier Giro-i-Nieto
Universitat Politecnica de Catalunya
Barcelona, Catalonia
xavier.giro@upc.edu

Oge Marques
Florida Atlantic University
Boca Raton, Florida (USA)
omarques@fau.edu

Vincent Charvillat
IRIT-ENSEEIHT
University of Toulouse
Toulouse, France
vincent.charvillat@enseeiht.fr

ABSTRACT

We introduce a new algorithm for image segmentation based on crowdsourcing through a game : Ask'nSeek. The game provides information on the objects of an image, under the form of clicks that are either on the object, or on the background. These logs are then used in order to determine the best segmentation for an object among a set of candidates generated by the state-of-the-art CPMC algorithm. We also introduce a simulator that allows the generation of game logs and therefore gives insight about the number of games needed on an image to perform acceptable segmentation.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/machine SystemsHuman factors; I.4.6 [**Image Processing and Computer VisionI**]: SegmentationPixel classification

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Crowdsourcing, figure-ground segmentation, labeling game, human computing

1. MOTIVATION

Semantic annotation of visual content is a process that requires linking pixels within an image with the semantic concepts associated to each group of pixels. This is often done in a user-assisted way. There exist several levels of interaction between users and visual content, ranging from an intentional and accurate annotation targeted at generating high-quality labels (e.g., LabelMe [14]), to a completely unintentional process which is more bound to be noisy (e.g., textual contents surrounding images on web pages and multimedia documents).

This paper proposes a method for semantic object segmentation, at a pixel level, based on high-quality annotation data collected from an online game: Ask'nSeek [3]. Ask'nSeek is a two-player game where one player (the *master*) places a small square on the image to be segmented, a square which is invisible to the other player (the *seeker*). The goal of the seeker is to click inside the hidden square, i.e., to guess the square's location. The seeker is required to solve the problem by requesting clues to the master that consist of the relative position of the square with respect to a semantic object within the image. The name of the object is typed in by the seeker, thereby providing a semantic label for the clicks.

Our approach utilizes the information collected from game logs to seed a semi-supervised image segmentation algorithm, which will eventually extract the objects in the image from the surrounding background. Contrary to most existing human-assisted image segmentation solutions, the proposed process is unintentional from the user side, who is engaged in an online game that is scalable to crowds and whose objective is *not* image segmentation.

While the previous work on Ask'nSeek [3] aimed at solving the object detection problem with a bounding box, the work reported in this paper focuses on increasing the spatial accuracy of the results, i.e., performing pixel-based object segmentation. More specifically, this paper explores how the user interaction captured from a crowd- and gaming-based approach influences the quality of the object segmentation process. In particular, we investigate the relationship between the amount of crowdsourcing work and the quality of the corresponding segmentation results by providing an estimation of the minimum amount of clicks which are necessary to reach a certain segmentation accuracy.

This paper is structured as follows. Section 2 introduces previous works that are related to this paper. In section 3 we explain how we use game logs from Ask'nSeek in conjunction with a state-of-the-art segmentation algorithm, and also devise a method for simulating game logs. Experiments and results are presented in section 4. Section 5 presents concluding remarks and directions for future work.

2. RELATED WORK

CrowdMM '13 Barcelona, Catalonia

2.1 Semi-Supervised Object Segmentation

Previous works have explored how a minimal user interaction can seed a semi-supervised segmentation algorithm. Many techniques have been developed within the interactive segmentation domain [1], [2], [12] where a user typically draws scribbles and/or bounding boxes to indicate the location of an object. This initial interaction generates an overlaid object mask, which can be adjusted later by the user with additional feedback. There exist two fundamental differences between these works and ours. Firstly, in these works the intention of the user when interacting with the image is to produce a good object segmentation. In our game-based scenario, on the other hand, the goal of the user is to win the game which, in the case of Ask'nSeek, is not achieved according to the quality of the segmentation. In fact, the user is completely unaware that their feedback can be used for such purpose. Secondly, in these works the foreground and background traces follow a coherent temporal sequence that try to correct the result of the last mask estimation. In our game-based approach, user interactions from different games are combined independently from the moment of their acquisition.

In this work, rather than adopting interactive segmentation algorithms to map the clicks by a player to a segment within the image, we have based our study on another family of segmentation algorithms that keep the user interaction to a minimum: category-independent object localization. These techniques try to estimate object candidates on the image [4], [6], [10] based only in the visual features contained in the image. They try to learn how objects and background are arranged in an image to try to discern them in a completely unsupervised approach. In particular, in this work we use the crowd-sourced collected data to select among the object candidates generated by Constrained Parametric Min-Cut problems (CPMC, [4]). This choice is based on the outstanding result of this technique in the Pascal VOC segmentation challenge, a benchmark that we also use to assess the quality of the results obtained using our approach.

2.2 Crowd-based Visual Annotation

The collection of unintentional user feedback from crowds for visual analysis has been approached in different ways.

The most popular approach is to design a collaborative effort aimed at a high quality annotation of a dataset. For example, LabelMe [14] has collected a large amount of local annotations by asking volunteers to draw a polygon around the object. Solutions in this family normally vary depending on the incentive, which can go from an abstract call to help science, to a very accurate pricing policy. These systems tend to produce high-quality segmentations, but may result in a tedious and boring tasks for the user.

The introduction of a crowd-based effort for image analysis has also been explored in the past. Previous works [5], [18] have used CAPTCHA human verification systems to actually annotate images. These systems combine data with available ground truth with other data aiming at being analyzed. In these cases, users will unintentionally annotate the unlabeled images while the verification step is based on the dataset with available ground truth. However, existing works have focused on textual annotation applied at the image global scale. In this work, we have focused on the user feedback based on clicks to generate accurate object segmentations at a local scale.

A popular strategy for obtaining crowd-sourced annotations is through on-line Games With a Purpose (GWAP), which exploit the high motivational levels achieved by games in such a way that the user interaction produces some type of valuable outcome.

The first game used for object detection at a local scale was *Peekaboom* [17]. This platform was the natural evolution of the popular ESP Game from the same authors [16], which generates pairs of images and labels at a global scale. The game is played in pairs, where one player reveals parts of an image so that the other can guess the textual label representing the object being discovered. The areas to be shown are indicated with clicks, which are supposed to be placed on the objects.

A similar approach is proposed in *Name-It-Game* [15], where objects were outlined by a *revealer* player and had to be predicted by a second *guesser* player upon a gradual appearance of the selected object. This interface combines freehand and polygonal segmentations, and the considered concepts were extracted from the WordNet ontology. This approach requires that the guesser must type the predicted labels, which in many cases will not be actually present in the image.

The two-role approach is simplified in *RecognizePicture* [11], where the gradual revealing of the image is automatically chosen following different patterns. Players must choose between four possible labels describing one of the semantics contained in the image. Such approach requires a previous stage where an annotation at a global scale must be previously available so that one of the four possible labels is indeed present in the image.

The presented work is based on Ask'nSeek [3], a GWAP which has been previously exploited for object detections with bounding boxes. In this paper we use the same framework to test its application to the segmentation of semantic objects. This platform presents some particularities that differentiate it with respect to other works. Firstly, it integrates in the same mechanism the annotation at the global and local scale, skipping the need of a previous system to generate labels at a global scale that are to be refined in the game. Secondly, the clicks collected on the image can also be associated to the *non-object* (background) class, in addition to the *object* (foreground) class. Thirdly, it avoids the generation of irrelevant textual tags as in [15], because the user introducing textual labels will always refer to semantic concepts which will indeed appear in the image.

3. CROWDSOURCED OBJECT SEGMENTATION

This work focuses on the potential of Ask'nSeek traces to help generating pixel-wise segmentation of the relevant objects within an image. In particular, the goal of this work is to estimate the minimum amount and type of user interaction necessary to achieve segmentation results of reasonable quality, comparable to the results of state-of-the-art segmentation algorithms.

This study is based on traces obtained by Ask'nSeek. These traces are combined with a ranked list of object hypotheses to estimate an accurate segmentation of the object. We also propose a "traces simulator" to enable assessing the proposed solution on a large dataset.

3.1 Ask'nSeek Traces

Ask'nSeek asks players to guess the location of a hidden region within an image with the help of semantic and topological clues. One player, the *master*, hides a rectangular region somewhere within a randomly chosen image. The second player, the *seeker*, tries to guess the location of the hidden region through a series of successive guesses, expressed by clicking at some point in the image. What makes the game interesting is that, rather than just blindly clicking around, the seeker must ask the master for clues relative to some meaningful object within the image before each and every click. Once the master receives a request for a clue from the seeker containing a label, it is required to provide a spatial relation, which is selected from a list: *above*, *below*, *on the right of*, *on the left of*, *on*, *partially on* or none of the above. These indications - in the form of (spatial relation, label), e.g., "below the dog" - accumulate throughout the game and are expected to be jointly taken into account by the seeker during game play. Based on the previously selected points and the indications provided by the master, the seeker can refine their next guesses and – hopefully – find the hidden region after relatively few attempts.

Figure 1 shows the final screen that the seeker can see at the end of a (successful) game, with red pins indicating incorrect guesses and the green pin indicating the final (correct) guess and the associated hidden square.

The game is played cooperatively, which means that both players want the hidden region to be found by the seeker as quickly as possible and before a timer (set to 2 minutes) runs out. The score of both players decreases after each new click, which encourages the players to quickly find the hidden region.

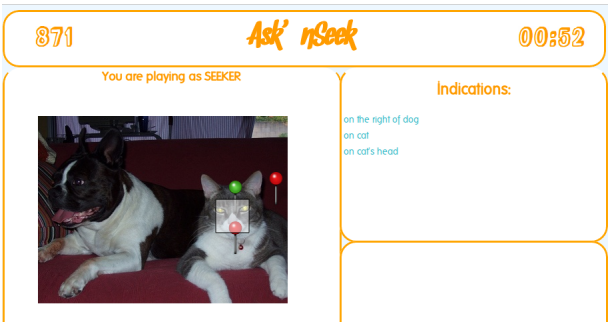

Figure 1: The end screen of a typical game of Ask'nSeek.

3.2 Consistency with Object Hypotheses

The traces collected with Ask'nSeek include valuable information regarding the labels and locations of the objects present in the image. Every object referred in the Ask'nSeek logs is associated to a set of foreground (fg) or background (bg) labeled pixels. These labels are generated from the players' clicks *on* the object (foreground labels) and the clicks *above*, *below*, *on the left* and *on the right* of the object (background labels). The *partially on* clicks also available in Ask'nSeek have been ignored in our set up.

In this work, these traces have been combined with the ranked set of segments generated by the Constrained Parametric Min-Cuts (CPMC) algorithm [4]. CPMC follows a two steps strategy: a first one that generates a set of candidate regions based on low level features (e.g., good alignment with image edges), and a second one that ranks these regions according to mid-level features, learned from a training dataset of segmented objects. As a result, the algorithm generates a ranked list of feasible segments within an image sorted according to their probability of being 'object-like'. CPMC has been adopted in our work because it was a basic block in the pipeline that obtained the best performance in Pascal VOC2009 and VOC2010 [7].

The ranked list of object hypothesis is compared with the collected foreground and background traces to select the best candidate. The proposed algorithm selects the first region in the ranked list of candidates with the maximum amount of coherent clicks. A click is considered coherent with respect to an object candidate when its label (fg/bg) matches its corresponding location in the image describing the object candidate.

3.3 Simulated Traces

The assessment of object segmentation techniques requires an extensive experimentation on large image datasets. On the other hand, one of the most limiting factors for research on crowdsourcing is the limitation on accessing large pools of users. These antagonistic scenarios can be managed by developing a simulation process to generate as many traces as desired, and validate the goodness of such simulator by collecting crowdsourcing data from a reduced portion of the whole dataset. This same approach has been adopted in previous works assessing algorithms for interactive segmentation [13], [9].

The goal of the simulator is to generate foreground and background clicks on the image. The simulator is supposed to have access to a segmentation ground truth of each test object, which codes the semantic interpretation that a human would infer by looking at it. The expected output is a collection of labeled clicks, as many as requested by the experiment. Two basic questions have been addressed during the design of the simulator: (a) the location of the clicks and (b) the ratio between the number of foreground and background clicks.

The location of a user's click on an image is influenced by its content. Humans pay more attention on some areas than others, being those more attractive also more feasible to attract more clicks. We propose saliency maps [8] to model these focuses of attention.

However, saliency maps alone are not a good estimator for clicks on Ask'nSeek. The main reason is that saliency maps tend to generate high values at the object contours, while our simulated user should be discouraged to click near these areas. Ask'nSeek seekers will only click near the object boundaries when the master generates a *partially on* indication, a possibility which is discarded in this paper. Taking into consideration that the box hidden by the master is 50×50 pixels, the proposed simulator applies a penalty to all those pixel locations which are 50 pixels or closer to an object boundary. The distance to the object boundary is obtained by computing the distance map from the foreground or background mask, depending on whether the simulated clicks are to be labeled as background or foreground, respectively. We compute the probability map as a simple product of the filtered distance map and the saliency map, the result of which can be seen on an example in figure 2.

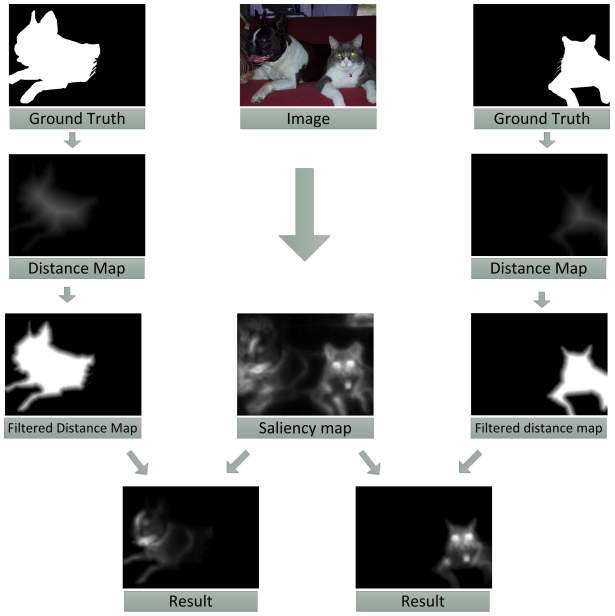

Figure 2: Generation of probability maps for foreground objects

In addition to avoid the object's contours, it has also been estimated that a player never clicks too close to the image borders, so an additional restriction of 25 pixels has been added to avoid the generation of clicks in these zones. It must be taken into account that these two restrictions with respect to object and image limits may, in the case of very large object, result in a lack of available locations for background clicks.

Once the method for deciding the location of the points is decided, the simulator still requires the specification of the proportion of clicks associated to the foreground or background. This work has tested two different options: the first one considers the relative size of the object with respect to the whole image as the probability of a foreground click; the second one considers the relationship between the sum of the saliency values for the foreground and background. Both proposals somehow link the ratio to the occupation of the object within the image, but the second one corrects this value with a measure of the human attention.

4. EXPERIMENTS

In this section, we present our experiments and the results using both real and simulated traces.

4.1 Protocol

We deployed a web-based version of the Ask'nSeek game, where players first encounter a tutorial video explaining them how to play the game. Players are then asked to log in, and wait to be paired with another player. The pairing algorithm is random.

For every game, we record the login of both the master and the seeker, along with the image they are playing over. The region hidden by the master is persisted, as well as every indication given by the master and every try of the seeker. At the end of the game, the final score and the remaining time (if any) are stored.

A total of 50 users (17 females and 33 males) played 255 games on 24 different images. The images were selected from the PASCAL VOC dataset. The players' age range between 18 and 62.

4.2 Game logs

We present in this section some statistics based on the analysis of game logs.

An average game lasts 59.15 seconds and consists of 2.05 indications given by the master to the seeker. 81% of the games are victories, which means the seeker managed to click inside the hidden region before the end of the timer (2 minutes).

The distribution of spatial relations (table 1) reveals that the *on* ('o') indication has been used the most (36% of all indications) and that the *partially on* ('p') indication was used the least (8%). Interestingly enough, in 80 % of the games, the master chose to hide the region in a semantically meaningful region (an object on the foreground). In those games, the percentage of *on* indications is even higher (44 % instead of 36 %). This percentage is dramatically lower (4 % instead of 36 %) in the games where the region is hidden on the background.

Set of Games	a	b	l	r	o	p
All	16%	8%	14%	18%	36%	8%
Region on fg	14%	5%	15%	15%	44%	7%
Region on bg	24%	22%	12%	28%	4%	10%

Table 1: Repartition of spatial relations among above, below, left, right, on and partially on points.

Regarding the number of indications per game, 40% of the games only require one indication, whereas 31% use two and 17% take three indications to converge.

We have also studied the precision of the game logs and observed that players often make mistakes while playing.

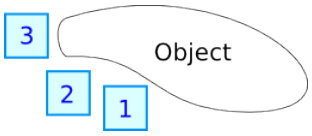

Figure 3: Different possible positions for a hidden region being *below* an object.

First, the master can give wrong indications, either because he misclicks or because he does not understand the name of the object requested by the seeker. We observed that 5% of the master indications are completely wrong, e.g., indicating the spatial relation *on the left of* instead of *on the right of*. The rest of the indications is valid, but has to be considered carefully. It is important to note that when a user states a hidden region is *below* an object, it does not necessarily mean that the region is mathematically below the object. Figure 3 shows examples of the regions the players consider being *below* an object. If regions 1 and 2 can indeed be considered below the object, it is important to know the proportion of such cases. Case 3 (the spatial relation is true everywhere in the hidden region) happens in 75 % of the cases where the spatial relation is either *above*,

below, *on the left of* or *on the right of*. Case 2 (the spatial relation is not true for the entire hidden region, but at least for its center) happens in 13 % of the cases, and case 1 (the spatial relation is not true for any pixel inside the hidden region, but is semantically meaningful) happens in 4% of the cases.

In the same spirit we can categorize the reliability of the seeker traces (their clicks after they were given an indication). 90.4% of the seeker's clicks are correct, whereas 6.8% of the clicks correspond to a seeker's mistake. The remaining 2.8% of wrong seeker's clicks are due to a bad indication of the master.

4.3 Object Segmentation

The potential of the presented framework in terms of object segmentation was assessed by comparing the obtained results with a ground truth. The Pascal VOC benchmark [7] provides a large collection of images that contain objects that belong to a diversity of semantic classes. We focused on the dataset from year 2010, to be consistent with the one used in the paper describing CPMC object candidates [4]. Every Pascal VOC image is provided with its corresponding pixel-wise ground truth, which allows the assessment of the object masks generated with our approach. The results are provided according to the Jaccard Index, sometimes referred as *overlap*.

Two types of experiments have been run. The first one aims at validating the simulator of traces by comparing its results to the ones obtained from real traces on a reduced dataset, while the second one uses this simulator to predict the behaviour of the system on a large volume on images.

4.3.1 Real vs Simulated Clicks

The simulated traces described in Section 3.3 have been validated on a reduced set of 10 objects extracted from the VOC2010 train and validation dataset. These 10 objects were selected as the ones with more clicks available on the game logs. The images have been actually used for playing Ask'nSeek and the collected traces contain at least 15 clicks for each of the considered objects. The final goal of this experiment is to assess that the simulator provides data that will generate similar segmentation results than real traces. The real traces generated a mean Jaccard index for the 10 considered objects of 0.4878. The comparison has been performed according to the Jaccard indexes obtained for the real traces and an average of 5 simulated runs.

The first feature to assess from the simulator corresponds to the location of the clicks. The proposed solution, combining saliency and distance maps, has been compared with a random distribution of points. In this case, the amount of simulated foreground and background points corresponds to the figures obtained from the traces. Table 2 shows the results, where the Jaccard Index for each generated segmentation is compared between the two types of simulation and real traces through the 10 considered objects. In addition, the results for the simulated cases were obtained by averaging the results of the 5 runs. The two simulation options are compared to the Jaccard indexes obtained with the real traces through the Mean Square Error (MSE). These results support the *Saliency* option as a better alternative to the random generation of positions, because its average MSE is much lower, as well as its variance.

Once the method for deciding the location of the points is decided, the simulator still requires the proportion of clicks associated to the foreground or background. Three options were considered: (a) *Fixed*: use a fixed proportion for all object by learning from traces (1/3 in this experiment), (b) *Area*: estimate from the object occupation; and (c) *Saliency*: estimate from the saliency values associated to foreground and background.

	μ_{MSE}	σ^2_{MSE}	$\min_{MSE}$
Saliency	0.0319	0.0007	0.0143
Random	0.0545	0.0073	0.0117

Table 2: Evaluation of the segmentations obtained with different strategies for generating spatial location of simulated clicks (Saliency-based, and random).

To perform this experiment, we generated the same amount of clicks for the simulated data than we got from the real traces.

The results in Table 3 clearly discard the option of using the same proportion for every object, with a mean MSE significantly larger than in the other two options. The *Saliency* solution seems to provide a better result because it both presents the lowest μ_{MSE} and $\min_{MSE}$. However, its higher variance with respect to the *Area* does not clearly discard this latter option. Nevertheless, the *Saliency* approach was adopted which, combined with the saliency-based estimation of the clicks location, support the subsequent experiments with simulated data.

	μ_{MSE}	σ^2_{MSE}	$\min_{MSE}$
Fixed	0.0703	0.0040	0.0277
Area	0.0472	0.0005	0.0299
Saliency	0.0453	0.0008	0.0219

Table 3: Estimators for the fg-bg clicks ratio

4.3.2 Amount of clicks

The main question addressed in this paper is an estimation of the amount of necessary clicks on a GWAP like Ask'nSeek to achieve a certain quality. Once the clicks simulator has been validated with real traces, it is possible to run extensive experimentation on a large dataset. The goal of these experiments is to study the evolution of the segmentation quality as the amount of clicks increase, as well as assess the influence that the semantic class of the object in the expected accuracy for a given amount of clicks. This experiment was run on the validation dataset of the Pascal VOC2010, which contains 964 images with object from 20 different classes.

Figure 4 presents the averaged Jaccard Index obtained for each of the 20 considered classes in the 5 runs with respect to the amount of simulated clicks. The plotted Jaccard index is an averaged value, which offers an estimation of the necessary clicks to obtain a certain quality. As expected, this averaged curve presents a growing trend with the amount of clicks, with a faster increase during the first 15 clicks. The last simulated value, for which 30 clicks have been used, corresponds to an average Jaccard Index of 0.5836. This value outperforms the winners of the Pascal VOC 2012 challenge who obtained 0.473, but on a different test dataset with the same 20 classes.

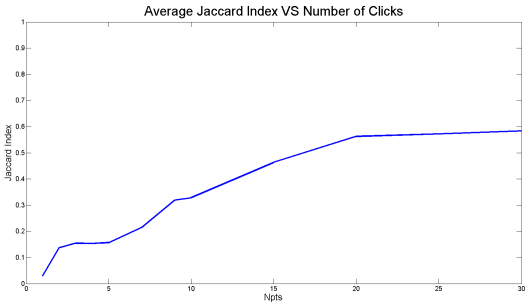

Figure 4: Average Jaccard Index vs Number of clicks.

However, a detailed study of the curves by class points shows that each of them has a particular behavior with faster or slower growths. The dependency of the results from the object classes is pictured in Figure 5. The average Jaccard index has been computed individually for each class for four amounts of clicks: 1, 10, 20 and 30. This graph indicates a clear correlation between the object class and the expected segmentation quality for a given amount of clicks. Object classes like *bus* or *aeroplane* obtain higher qualities with the same amount of clicks than more challenging classes like *bicycle* or *chair*. While the first classes represent compact objects which tend to occupy large portions of the image, the latter are complex objects made of thin structures that difficult the segmentation task. Notice that, in a few cases (*person* or *train*), a larger amount of clicks may slightly reduce the obtained accuracy. This is because, in a few cases, new clicks can discard object candidates that fitted better to the object than the newly selected candidate.

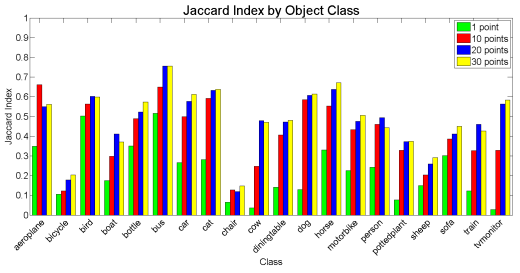

Figure 5: Jaccard Index by Object Class.

5. CONCLUSION

In this paper we have shown how we could use game logs from players' interactions in the Ask'nSeek game to feed a segmentation algorithm and improve the quality of state-of-the-art unsupervised segmentation results . We have also presented how to design and validate a simulator to generate such game logs. Our experiments point that a total of 20 clicks related to an object will provide an accuracy quality that will not increase significantly on further clicks. However, this is an average estimation, with a high dependency on the object category. In our future work we want to conduct a larger-scale user study in order to obtain results on a full dataset, namely the BSDS dataset which has been used in other interactive segmentation works. In addition, we also plan to introduce additional factors (eg. visual entropy, object category...) on the estimation of the amount of clicks. The simulator and evaluation software, as well the collected datasets, have been made publicly available from UPC website [1].

6. ACKNOWLEDGMENTS

The authors would like to thank Jordi Pont-Tuset for his valuable contributions. This work has been partially funded by the Camomile CHIST-ERA project, and by the Spanish project TEC2010-18094 MuViPro.

7. REFERENCES

[1] P. Arbelaez and L. Cohen. Constrained image segmentation from hierarchical boundaries. In *CVPR'08*, pages 1–8, 2008.

[2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary map; region segmentation of objects in n-d images. In *ICCV'01*, pages 105–112 vol.1, 2001.

[3] A. Carlier, O. Marques, and V. Charvillat. Ask'nseek: A new game for object detection and labeling. In *ECCV'12 Workshops*, pages 249–258. 2012.

[4] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR'10*, pages 3241–3248, 2010.

[5] R. Datta, J. Li, and J. Z. Wang. Imagination: a robust image-based captcha generation system. In *ACM MM'05*, pages 331–334, 2005.

[6] I. Endres and D. Hoiem. Category independent object proposals. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV'10*, pages 575–588. 2010.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[8] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *PAMI*, 34(10):1915–1926, 2012.

[9] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR'10*, pages 3129–3136, 2010.

[10] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV'12*, volume 7578, pages 444–458. 2012.

[11] M. Lux, A. Müller, and M. Guggenberger. Finding Image Regions with Human Computation and Games with a Purpose. In *AIIDE'12*, pages 41–43, 2012.

[12] K. McGuinness and N. E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434 – 444, 2010.

[13] K. McGuinness and N. E. O'Connor. Toward automated evaluation of interactive segmentation. *CVIU*, 115(6):868–884, 2011.

[1] https://imatge.upc.edu/web/publications/crowdsourced-object-segmentation-game-0

[14] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[15] J. Steggink and C. Snoek. Adding semantics to image-region annotations with the name-it-game. *Multimedia Systems*, 17:367–378, 2011.

[16] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI'04*, pages 319–326, 2004.

[17] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *CHI'06*, pages 55–64, 2006.

[18] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition captchas. In *CCS '10*, pages 187–200, 2010.