Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Assessment of Crowdsourcing and Gamification Loss in User-Assisted Object Segmentation

Axel Carlier · Amaia Salvador · Xavier Giro-i-Nieto · Vincent Charvillat · Oge Marques

Received: date / Accepted: date

Abstract There has been a growing interest in applying human computation – particularly crowdsourcing techniques – to assist in the solution of multimedia, image processing, and computer vision problems which are still too difficult to solve using fully automatic algorithms, and yet relatively easy for humans.

In this paper we focus on a specific problem – object segmentation within color images – and compare different solutions which combine color image segmentation algorithms with human efforts, either in the form of explicit interactive segmentation task or through an implicit collection of valuable human traces with a game.

We use Click'n'Cut, a friendly, web-based, interactive segmentation tool that allows segmentation tasks to be assigned to many users, and Ask'nSeek, a game with a purpose designed for object detection and segmentation.

The two main contributions of this paper are: (i) We use the results of Click'n'Cut campaigns with different groups of users to examine and quantify the *crowdsourcing loss* incurred when an interactive segmentation task is assigned to paid crowdworkers, comparing their results to the ones obtained when computer vision experts are asked to perform the same tasks. (ii) Since interactive segmentation tasks are inherently tedious and prone to fatigue, we compare the quality of the results obtained with Click'n'Cut with the ones obtained using a (fun, interactive, and potentially less tedious) game designed for the same purpose. We call this contribution the assess-

Axel Carlier and Vincent Charvillat
IRIT-ENSEEIHT, University of Toulouse
Toulouse, France
E-mail: {axel.carlier,vincent.charvillat}@enseeiht.fr

Amaia Salvador and Xavier Giro-i-Nieto
Universitat Politecnica de Catalunya (UPC)
Barcelona, Catalonia/Spain
E-mail: {amaia.salvador, xavier.giro}@upc.edu

Oge Marques
Florida Atlantic University (FAU)
Boca Raton, Florida, USA
E-mail: omarques@fau.edu

ment of the *gamification loss*, since it refers to how much quality of segmentation results may be lost when we switch to a game-based approach to the same task.

We demonstrate that the crowdsourcing loss is significant when using all the data points from workers, but decreases substantially (and becomes comparable to the quality of expert users performing similar tasks) after performing a modest amount of data analysis and filtering out of users whose data are clearly not useful. We also show that – on the other hand – the gamification loss is significantly more severe: the quality of the results drops roughly by half when switching from a focused (yet tedious) task to a more fun and relaxed game environment.

Keywords GWAP · Crowdsourcing · Serious Games · Object Detection · Object Segmentation

1 Motivation

The Semantic Gap between low-level visual features and high-level semantic concepts is still a problem in many computer vision tasks. The ability to make sense of pixel data in a way that is less dependent of their raw nature and more related to their high-level semantic interpretation (objects, labels, concepts) is something that humans do well but computer algorithms still do not (with the possible exception of a few selective tasks, e.g., face verification [1], where the performance of computer-based solutions has started to approach the levels of human performance).

In this paper, we focus on the object segmentation task, which could be described as follows: given a color image, create a binary mask of the same size, where all the pixels that belong to an object of interest are marked as *true (white)* and all the remaining pixels (other objects, background) are marked as *false (black)*. For example, Figure 1 depicts how a dog from a photo on the left can be extracted with the binary mask on the right.

Object segmentation applications include: image and video coding, semantic-based adaptive compression, visual-based hyperlinking, clickable video and photos, media mixing, product placement, Augmented Reality (AR), and inpainting.

We adopt a human computation paradigm to solve the object segmentation problem, which consists of designing software solutions that allow a user to interact with the image and produce the intended results (Figure 1). In this particular work, we look at two significantly different approaches for engaging the user's help to perform a computer vision task: (i) crowdsourcing through an interactive segmentation tool named *Click'n'Cut* and (ii) serious games, also called Games With A Purpose (GWAPs), through an online game called *Ask'nSeek*. In both cases, some of the most popular contemporary algorithms for object segmentation work behind the scenes to assist in the task. User annotations are used to select the most appropriate segmentation mask among a pool of candidates. In *Click'n'Cut* users are even guided by these algorithms to produce annotations in the most meaningful regions. The main difference between crowdsourcing and GWAP, of course, is that while the crowdsourcing approach is potentially tedious and depends on the attention to detail, patience, and understanding of the task by the workers, the game-based approach is potentially more relaxed and fun and does not look like a chore.

In this paper we are particularly interested in studying how much segmentation quality is lost when an interactive segmentation task is performed by either modestly paid workers with no expert knowledge (in the crowdsourcing case) or using data points from Ask'nSeek game logs. We use the data collected from a group of "expert users" (using Click'n'Cut) as a baseline for comparison. Throughout the paper we shall refer to the former type of loss as the *crowdsourcing loss* – the type of loss incurred when an interactive segmentation task is assigned to paid workers who are not segmentation experts, using the same tool – whereas the latter type will be called the *gamification loss*, which refers to how much quality of segmentation results may be lost when we switch to a game-based approach to the same task.

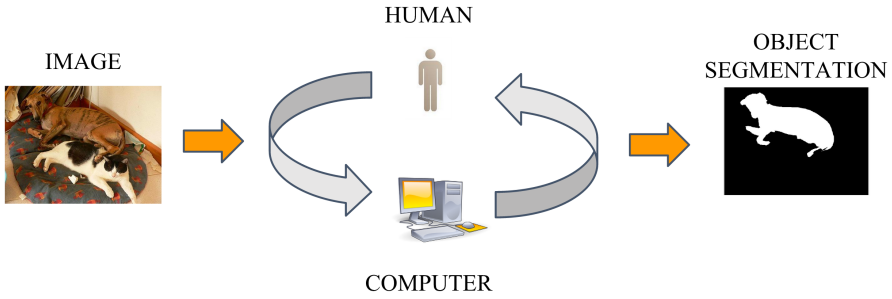

Fig. 1 Introducing the human "in the loop" in object segmentation.

The rest of the paper is organized as follows: Section 2 discusses related work in the fields of interactive object segmentation, crowd-based object annotation, and games with a purpose (GWAP). Section 3 describes *Click'n'Cut* and Ask'nSeek, the two online tools designed to collect human clicks for the object segmentation task. Section 4 discusses the object segmentation algorithm used to generate binary masks from the collected user clicks. Section 5 describes the experiments performed using Ask'nSeek and Click'n'Cut and discusses the most relevant results. Finally, Section 6 suggests directions for future work and Section 7 presents the final conclusions.

2 Related Work

Interactive Object Segmentation. The segmentation of objects by combining human interaction and image processing algorithms has been extensively explored in the literature. In such interactive setup, the graphical user interface responds to some sort of weak annotation (bounding box, scribbles, clicks...) from the user by generating and displaying a complete segmentation of the object. The typical workflow expects the user to interact with the proposed solution either by accepting it or by providing more traces that may allow the segmentation algorithm to converge to a satisfactory result. Most interactive segmentation techniques normally propagate the user-generated indications of which pixels belong to the *foreground* or *background*

through a graph-based representation of the image. Among these graph-based solutions, two main families of algorithms can be identified depending on whether the nodes of the graph correspond to a pixel or to a region (superpixel).

The foundational proposal for interactive foreground segmentation was based on *graph cuts* [2]. The algorithm considers every pixel as a node in a graph, connected to their spatial neighbors by an edge whose weight depends on the visual similarity between pixels. In addition, every pixel node is also connected to two special terminal nodes, each of them representing an *foreground* or *background* label, with a weight associated to the similarity of each pixel to a model of the *foreground* or *background*. Segmenting an object is equivalent to finding the *min cut* of the graph, that is, those edges that once disconnected minimize a certain energy function defined on the two resulting sub-graphs. The *graph cuts* approach has also been expanded by other authors [3].

Superpixel-based solutions [4] [5] [6] [7] avoid the computational load of a pixel-by-pixel segmentation by working with unsupervised image segmentations performed offline. These solutions rely strongly on the region boundaries defined by the segmentation algorithm, which are assumed to correspond to the semantically meaningful regions in the image. These image partitions are usually configured to generate over-segmentations from the semantic point of view, which include all object contours as well as many additional ones, which are to be removed through the interaction process. The process of mapping user interaction to regions in the partitions, so that the labeled pixels are assigned to their corresponding region, is straightforward.

The adjacency information between regions coded in a partition can be further enriched by iteratively merging pairs of neighboring regions. As a result, *hierarchical partitions* contain additional information capable of capturing the multiscale semantic nature of an image. Interactive segmentation solutions based on hierarchies [8] [9] [10] [11] use these data structures to propagate labels not through a flat graph partition but, instead, considering regions at multiple scales. The comparative study in [12] indicated similar accuracy labels for GrabCut [3] and hierarchical solutions [8] [9], but a faster response for the latter ones.

The solution adopted in our work does not solve any labeling of a graph but generates segmentations by combining a precomputed set of object candidates (also referred to as "saliency detectors"). Instead of considering pre-computed regions which are normally generated considering only perceptual criteria, our basic processing unit are regions that have been defined by an algorithm that estimates the "objectness" of an automatically generated segment, i.e. how likely a segment is to correspond to a semantically meaningful object. Object candidate techniques [13] [14] [15] generate a ranked list of object candidates for the image, based on its visual features together with additional parameters learned from a collection of semantically meaningful regions. The presented approach is an extension of [16], where crowdsourced clicks labelled as *foreground* or *background* were mapped into a collection of object candidates to select the region which better matched the captured traces. However, our system is more flexible than [16], because it obtains solutions that can combine multiple candidates. Our *Click'n'Cut* system has been introduced in [17] and uses this combination of object candidates.

Crowd-based Object Annotation. The collection of object annotations from the crowds has been approached in different ways in the literature. Some authors [18] [16] have designed games so that users would unawarely generate segmentation traces. This way, the task would not become so tedious and the gaming incentive might eliminate or reduce the financial one. Most initiatives for object segmentation have adopted a *collaborative* approach where users are instructed on how to generate high quality segmentations. 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. *Click'n'Cut* relies on workers that were explicitly paid to generate an accurate segmentation of the objects.

One of the most popular initiatives in this direction is *LabelMe* [19], on online platform that has collected a large amount of local annotations by asking volunteers to draw a polygon around the object.

A very ambitious initiative is related to the Microsoft COCO (Common Objects in COntext) dataset [20]. This project uses workers from Amazon's Mechanical Turk to segment the objects in images at an estimated rate of 79 seconds per object instance. Only one worker segments each object, but this worker must first pass a training stage before being qualified to segment. This segmentation effort uses the *OpenSurfaces* interface [21], an open source tool based on polygonal segmentation.

The authors of [22] compare the segmentation results achieved from crowdsourced workers who draw polygons around cars with the results obtained by applying a computer vision algorithm in the bounding boxes provided as ground truth. Ground truth segmentations were generated by 9 in-lab annotators using a similar interface as workers, taking on average 60 seconds to label each car. The crowdsourced task was organized in two batches: a first one paid 1 cent per annotated car and a minimum 75% approval rate, and a second one paid 5 cents per car and 95% approval rate required. Results showed a small (1%) increase in the final quality of the segmentations for the higher priced case.

The crowd was also used in [23] to assess an interface aimed at choosing the best input modality among a bounding box, a sloppy contour or a tight polygon. The authors highlight that in crowdsourced campaigns the *annotation time* is the basic budget constraint, and that by automatically adapting the annotation mode to the image it is possible to optimize the quantity and quality of the segmentations. The selection is based on an estimation of the average time necessary for each modality: 7 sec (seconds) for bounding boxes, 20 sec for sloppy contours and 54 sec for tight polygons. Their study was performed on 101 workers and a dataset of 420 images, collecting a minimum of 5 responses for each modality per image.

In our work we have tried to adjust as much as possible to the experiment described in [12] to be able to compare the quality of a crowdsourced solution with respect to an campaign with expert annotators.

Finally on the related topic of object co-segmentation which consists in segmenting the same object in multiple images that feature this object, it is worth describing iCoseg [24]. In this paper, the authors allow users to draw scribbles on images to annotate background and foreground. The scribbles on one image are used to co-segment all images that show the same object. In addition, the authors use an active learning formulation that allows the system to automatically detect the areas

that would lead to the most informative scribbles, and propose it to the users. Our *Click'n'Cut* interface also displays feedback from the system to the users, to guide their interaction into the most relevant part of the image.

Games With A Purpose (GWAP). Boredom will limit the duration of the annotation sessions that users will be willing to accept. Collaborative campaigns tend to produce high-quality segmentations, but may result in tedious and boring tasks for the user. This limitation has been addressed in other works by designing Games With a Purpose (GWAP) capable of capturing valuable traces for object segmentations. In these cases, users (players) may be unaware that their feedback can be used for such purpose and still provide high quality traces.

There exist two fundamental differences between explicit and implicit collection of human traces. Firstly, in the work we just described where users directly interact with the segmentation resulting from their traces, their aim is explicitly the generation of high quality masks and the instant feedback guides them to generate the most informative traces. On the other hand, in a game-based scenario the goal of the user is to win the game which, in the case of Ask'nSeek (the GWAP that we propose), is completely unrelated to the quality of the segmentation coming out of the user traces. Secondly, interactive segmentation interfaces that collect 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.

A popular strategy for obtaining crowd-sourced annotations is through on-line GWAPs, which exploit the high motivational levels achieved by games in such a way that the user interaction produces some type of valuable outcome. The *Extra Sensory Perception (ESP)* game [25] collects textual labels at the global scale by showing the same image to a pair of players. Players are prompted to enter words related to the shown image and, when an agreement is obtained between different players, they are rewarded with points. The label is accepted as correct when different pairs of users agree about it.

The first game used for object detection at a local scale was *Peekaboom* [26]. This platform is the natural evolution of the popular ESP Game from the same authors [25], which generated pairs of images and labels at a global scale. *Peekaboom* is played in pairs, where one player reveals parts of an image so that the other can guess the textual label representing the object that is being discovered. The areas to uncover are indicated with clicks, which are supposed to be placed on the objects.

The *Name-It-Game* [18] is played in pairs and collects both images and textual labels for the segmented objects. In that game, objects are outlined by a *revealer* player and their label must be predicted by a second *guesser* player upon a gradual appearance of the selected object. This interface combines freehand and polygonal segmentations, similar to LabelMe. The authors claim that by combining multiple traces obtained from games played using the same image, results are similar to the ones obtained by the LabelMe annotation campaign [19]. Our experiments have also fused traces from different users on the same image to clean out noise, but the task of our workers is not freehand, but assisted by a computer vision algorithm instead.

The two-role approach is simplified in *RecognizePicture* [27], 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 to make sure that at least one of the four possible labels is indeed present in the image. *Ask'nSeek* [16] also involved the participation of two players to collect the textual labels of the objects contained in an image, as well as selecting the best object candidate based on on clicks labeled as *above*, *below*, *on the left*, *on the right* or *on* the objects.

3 User Interfaces

Human computation requires the capture of user interaction; taken altogether, these human contributions act as a computer that assists in solving a problem. This section describes the two online tools used in this study to solve the object segmentation problem. The intentional interaction has been collected with *Click'n'Cut* [17], an interactive object segmentation tool described in Section 3.1. On the other hand, unintentional interaction has been captured by the *Ask'n'Seek* game [28], described in Section 3.2.

3.1 Click'n'Cut Interactive Segmentation Tool

In *Click'n'Cut* [17], users are asked to produce foreground and background clicks to perform a segmentation of the object that is indicated in a provided description. The fundamental interactions available to the user are the left and right clicks, which generate foreground and background points, respectively. Figure 2 shows the interface, which displays the image that we wish to segment, along with a set of basic interactions (on the bottom-right of the screen) and a reminder of how the interface works (on the top-right part of the screen). There is also a description of the object to segment on the top of the screen, right above the image. Every time a user generates a click, the segmentation result is updated and displayed over the image with an alpha value of 0.5 (which can be changed by the user using a *Transparency* slider). This segmentation is computed thanks to the algorithm described in section 4 and aims at guiding the user to provide information that will actually help improving the final segmentation.

Users can also correct a wrong click by just clicking on it again to make it disappear. The *Clear points* button removes the entire set of clicks that have been made by the user. Finally, once satisfied with the result, the user can move on to the next task by clicking the *Done* button.

3.2 The Ask'n'Seek game

Ask'n'Seek [28] is a two-player, web-based, game that can be played on a contemporary browser without any need for plug-ins. One player, the *master* (Figure 3, bottom) hides a target region somewhere within a randomly chosen image. The second player

Fig. 2 Screenshot of the Click'n'Cut interface.

(*seeker*) (Figure 3, top) 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 more 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, to the right of, to the left of, on, partially on, none of the above}*. These indications – in the form of (spatial relation, label), e.g., "on the church" – 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 – guess the hidden region after relatively few attempts.

Figure 4 illustrates a typical gameplay with Ask'nSeek. In the game featured on this example, the seeker first asked the master an indication relative to the dog, and the master answered that the region is "on the right of the dog". The seeker clicked on the image but not on the region, so he asked for a second clue, relative to the cat. The master answered that the region is "on the cat", and the seeker once again did not find the region. He finally got the indication that the region is "on the cat's head", and clicked on the right location. Once he clicks inside the region, the actual location of the region chosen by the master is prompted to the seeker (before finding it, he could

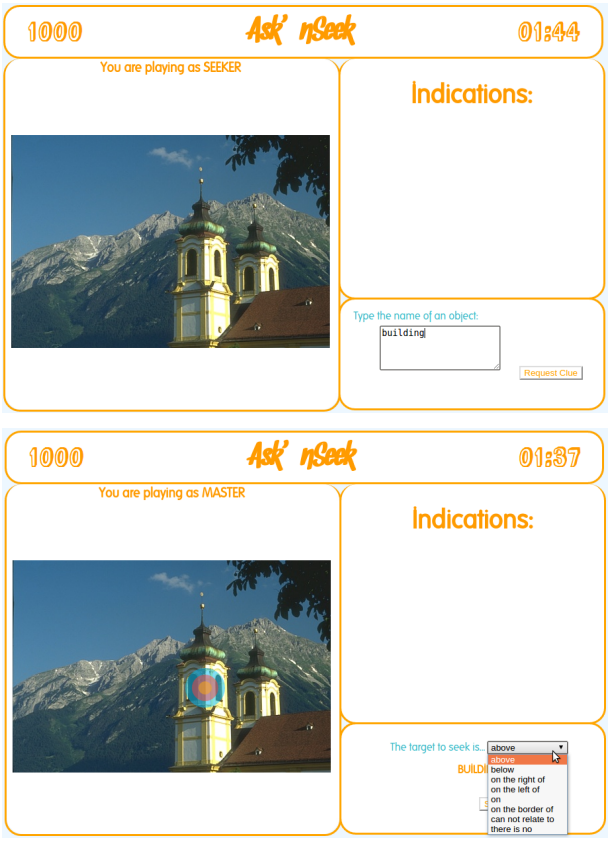

Fig. 3 Screenshots of the Ask'nSeek game: seeker's screen (top); master's screen (bottom).

not see it). In other words, the square only appears on the image after the seeker has managed to click inside it.

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.

Traces from *Ask'nSeek* can be processed in order to categorize a set of clicks labelled as *foreground* or *background*, as in the case of *Click'n'Cut*. For example, a click "on the building" is *foreground* relatively to the object building but *background* relatively to the object sky. In addition, a click "on the right of the building" is *background* relatively to the object building.

However, the motivation for the user is different in the *Ask'nSeek* case. Playing the game is the main goal, instead of obtaining an accurate segmentation. In fact, players are completely unaware that their interactions can be exploited to solve an object segmentation problem.

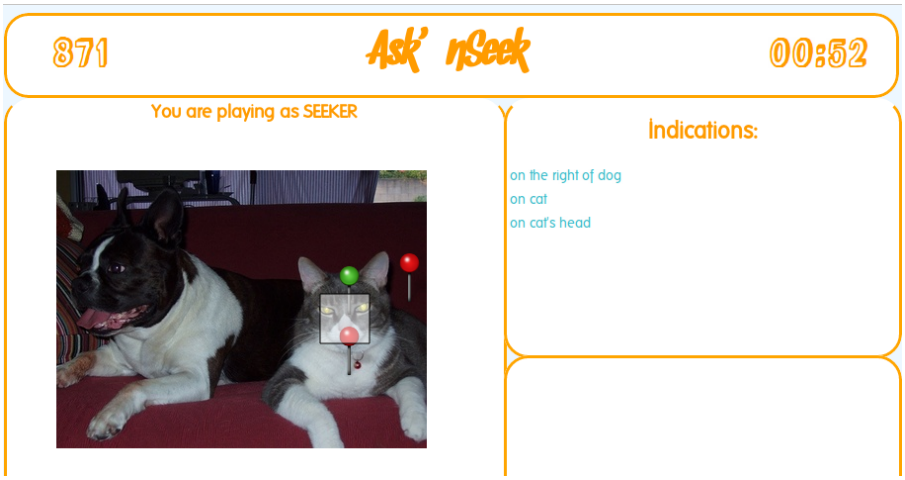

Fig. 4 Example of the game from the seeker's point of view.

4 Object Segmentation

In this section we describe the algorithm for performing image segmentation that we use in our work. The algorithm is based on object candidates and is used in our Click'n'Cut implementation to display a feedback mask to the user (see Fig. 2).

To compute the best mask with respect to a set of f foreground points and b background points, we adopt the following algorithm.

A set of object candidates *MCG* is generated using the *Multiscale Combinatorial Grouping* introduced in [15]. Figure 5 shows how these binary masks can be of various sizes and natures. In this case the masks are computed on the same image that is used in figure 2.

For each mask $m \in MCG$, we start by computing two scores fg_m and bg_m. fg_m (resp. bg_m) is the number of foreground (resp. background) points that are correct with respect to m. For example on Figure 5 the foreground point (in green) is correct with respect to masks 1 and 4.

Then if there exists a mask m^* for which $fg_{m^*} = f$ and $bg_{m^*} = b$ then m^* is the best possible mask and this is the mask that will be shown to the worker.

Else, it means that no mask fits perfectly with all collected clicks. In that case, we build a novel binary mask as the union of all masks that belong to $M^* = \{m \in MCG, bg_m = b \text{ and } fg_m > 0\}$. This means that M^* aggregates all those binary masks that have not received any background click and for which there is at least one foreground point.

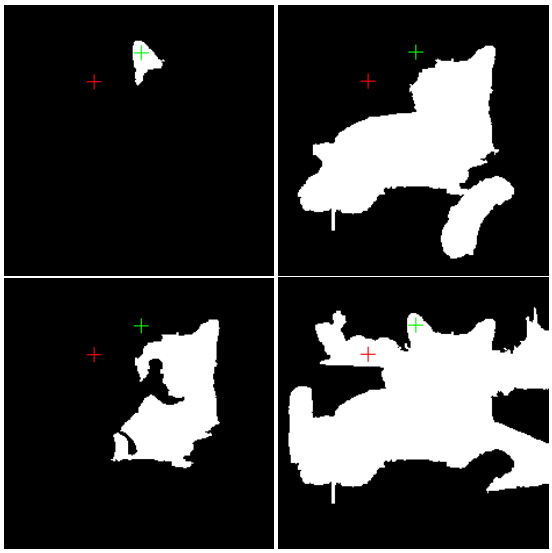

Fig. 5 Examples of MCG object candidates.

5 Experiments and Results

In this section we introduce our experiments for which we use the interfaces described in Section 3, and whose data have been processed with the techniques presented in Section 4.

Using the data collected during these experiments we try to estimate a crowsourcing loss, i.e. the eventual loss in information collected and results obtained when going from an experiment using expert users to an experiment employing paid workers recruited on a crowdsourcing platform.

We also study the gamification loss that can occur when instead of incentivizing users or workers with money, we use a GWAP to have gamers (only motivated by fun) produce meaningful information through their game logs.

Our object segmentation experiments have been conducted on a dataset proposed by [12] in their work on interactive segmentation which makes our results comparable with the resutls from that study. The collection contains 96 images selected from the larger Berkeley Segmentation Dataset [29], and also includes the ground truth binary masks for 100 objects (2 images have three associated objects each), and a textual description for the users of the object to segment. In order to study and control the quality of our user traces, we have added 5 images from the PASCAL VOC [30] segmentation dataset and have written a textual description of one object per image. Therefore our image set is composed of 101 images, and there are 105 tasks (objects to segment) to perform.

Our experiments were conducted on three different user profiles:

– **Click'n'Cut - Experts**: 15 computer vision researchers from academia, both students and professors.

– **Click'n'Cut - Workers**: 20 paid workers from the platform `microworkers.com`, 4 Females and 16 Males with ages ranging from 20 to 40 (average: 25.6). Workers were all from South-East Asia, with a large majority (17 out of 20) of users originating from India and Sri Lanka. Each worker was paid 4 USD for annotating 105 images.
– **Ask'n'Seek - Players**: 162 players (mostly students) played the Ask'n'Seek game on the number of images they wanted to.

5.1 Preliminary figures

Table 1 presents a comparison of figures for the three experiments. The first main comment is that the workers produced a lot of clicks. In average, workers clicked more that twice as many times as the experts on the same images (they were 20 against 15), and ten times more than the players.

	Click'n'Cut	Click'n'Cut	Ask'n'Seek
	Experts	Workers	Players
# Users	15	20	162
# Clicks (per image, all users included)	234.4 168 FG 66.4 BG	544.6 345.8 FG 198.8 BG	51.4 29 FG 21 BG 1.4 Part. On
# Errors	4%	35%	7%

Table 1 Comparison of the number of clicks and error rates in the different setups.

Another interesting difference between the groups of users is the ratio between foreground and background clicks. Expert users mostly produce foreground clicks (72% of the times). Workers also use more foreground clicks, but the ratio is 63%/37%. Finally, players tend to produce 57% of foreground clicks.

The most remarkable value in Table 1 is the percentage of erroneous clicks, defined as the number of clicks that are badly categorized, i.e. foreground clicks that are in fact on the background and vice versa. We did not consider the *Partially On* clicks from Ask'n'Seek in this percentage, as 1) they represent a minority of clicks, and 2) they are a specificity of Ask'n'Seek, therefore not comparable to Click'n'Cut.

The first interesting finding is the difference in the amount of clicks per user that are collected with the two interfaces. There are several reasons that explain this difference. First, it takes 2 players to produce a click in Ask'n'Seek (the master and the seeker) whereas only one user is necessary in Click'n'Cut. Second, the users in Ask'n'Seek played an average of 30 games (i.e. images) each whereas Click'n'Cut users performed the entire set of 105 tasks. Also, Click'n'Cut users were given the possibility to do as many clicks as they wanted to. Ask'n'Seek players are limited by a 2 minutes timer (which includes the time to type labels, and exchange indications). The game stops when the seeker finds the target, which occurs after an average of 1.6 indications per game. Finally, Click'n'Cut users are focused on one object for each image: the object they have to segment. In Ask'n'Seek, the players use all the objects

they can see in the image. In other words it takes two players to play an average of 30 games that usually produce 1.6 clicks each, which, due to the game's nature, do not necessarily have to be related to a single object.

It is also interesting to take a look at the number of errors performed by workers on Click'n'Cut with respect to the Ask'n'Seek players. As we will see in the next subsection, the high error rate of the workers on Click'n'Cut is partly due to a specific subset of workers who performed particularly bad. Ask'n'Seek players error rate is much more homogeneous. The fact that Ask'n'Seek is a game naturally limits the impact of some of the usual sources of errors in crowdsourcing. The major sources of errors were, as listed in [31], the spammers, the incompetent workers and the insufficient attention from the workers. Being a game, Ask'n'Seek is relatively safe from spammers (the game has nothing to offer except for enjoyment; if players do not like the game, they are free to leave), and workers' attention is kept at a certain level by the non-repetitiveness of the task. Unlike Click'n'Cut where the task to perform is always the same, Ask'n'Seek players regularly switch roles (from master to seeker) and since the players' pairing is random, players interact with different people over time. The major source of errors in Ask'n'Seek is the misunderstanding between the master and the seeker. Misunderstanding can arise from an imprecise requested object from the seeker (e.g. "fish" in an image where there are three fishes), or from the master not understanding a word used by the seeker.

5.2 User profiling based on the types and correctness of the clicks

In this section, we take a closer look at user traces on Click'n'Cut to try to understand the **Crowdsourcing loss** suffered between the results of the experts and the workers.

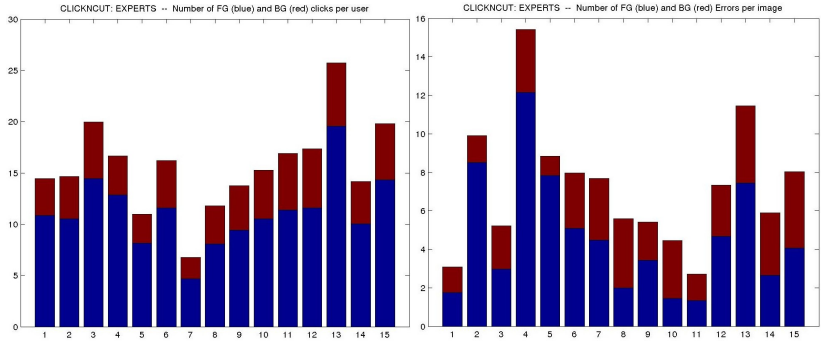

Fig. 6 Number of foreground/background clicks (left) and percentage of foreground/background errors (right) per expert user on Click'n'Cut

Figure 6 shows an analysis of the experts' clicks and errors. The numbers are averaged on all tasks. Expert users produce in average from 7 to 26 clicks per image and it is interesting to note that for each expert user, the proportion of foreground (blue)/background (red) clicks is fairly similar.

The right graph of Figure 6 presents the percentage of errors on foreground clicks (in blue) and on background clicks (in red). Note that these percentages do not take into account the number of foreground and background clicks, which means that the mean of the two percentages is not equivalent to the total error rate. It is interesting to note that the expert's highest source of errors seems to be the foreground clicks.

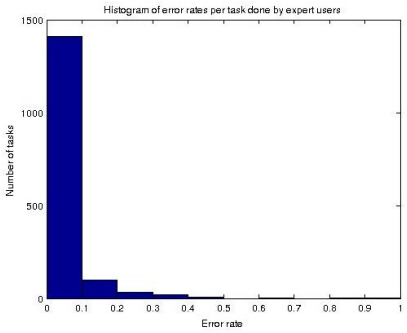

Fig. 7 Histogram of the error rates per task.

To further understand this phenomenom, let us consider the following numbers. Expert users have produced 24,611 clicks on $15 * 105 = 1,575$ tasks, and among those clicks there were 1,042 wrong ones. The 10 tasks (out of 1,575) for which the most errors were made account for a total of 372 errors, i.e. more than one third of the errors. This error rate distribution is shown on figure 7, on which we can see that a very large majority of the tasks had a very low error rate.

Fig. 8 Two of the tasks that produced many errors. Descriptions associated to the tasks are: 'Extract just the man's hat. Do not include the rest of the man or any or any other objects.' (left) and 'Extract the topmost fish on the center-right of the image.' (right)

Figure 8 presents two tasks that have created a large number of errors from the experts. On the left, only the man's hat should have been segmented. Two experts segmented the man, which created 100 errors (one tenth of the total number). On the

right, the description of the fish to segment (“topmost fish on the center-right”) was also misunderstood by two experts.

It is interesting to note that these errors are due to insufficient attention from the experts. This suggests we should always have more than one expert performing a task (typically in our traces, there were never more than two experts who misunderstood a task at the same time).

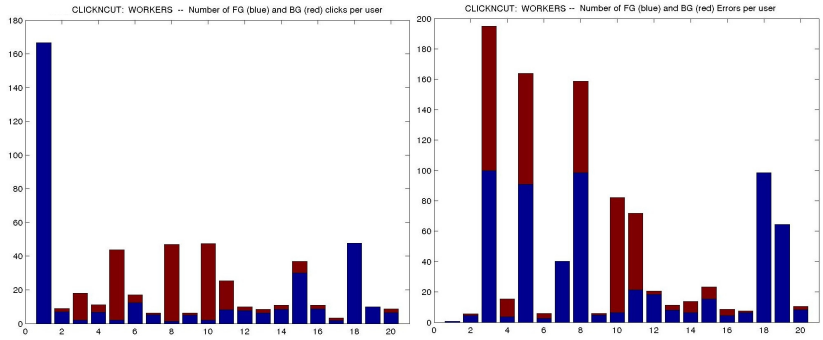

Fig. 9 Number of foreground/background clicks (left) and percentage of foreground/background errors (right) per paid worker on Click’n’Cut

Figure 9 shows the same type of plot as in Figure 6 but this time for paid workers. The first very obvious fact is that, unlike expert users, workers have a very heterogeneous way of interacting with Click’n’Cut. Five workers out of 20 produced a majority of background clicks, whereas we previously observed that all expert users clicked a higher number of foreground clicks. The distribution of the number of clicks is also clearly biased by one user (user # 1), who produced an average of 160 clicks per image (only foreground clicks). The right plot of Figure 9 also shows that this particular user (user # 1) made very few errors. We should be careful with the data from this user since it can affect our results a lot without being statistically significant.

The biggest difference between experts (Figure 6) and workers (Figure 9) is the diversity of user profiles in the case of the workers. This heterogeneity is visible when comparing the graphs of error percentages, depicted on the graphs on the right. This observation has led us to propose a categorization of the workers, inspired by [31], and exemplified in the cases contained in Figure 10):

- Worker # 1, a.k.a. "The painter" produced only foreground clicks, with an exceptional amount of clicks and an error rate almost equal to 0%. In fact we suspect that this user misunderstood the task and believed he had to entirely paint the object with green clicks.
- Workers # 3 and 5, a.k.a. "The mirrors", have such a high error rate that by inverting their contributions (considering their background clicks as foreground, and vice versa), they would actually display a very low error rate. We can only assume that they misunderstood the instructions, confusing foreground and background clicks.

- Worker # 8 and 10, a.k.a. "The border guards" produced almost exclusively background clicks located on the border of the objects.
- Worker # 18, a.k.a. "The surrounder" produced only foreground clicks, and has almost 100% errors. He tried to surround the object with foreground clicks, in as similar way as requested in LabelMe [19].
- Worker # 19, a.k.a. "The spammer", randomly placed foreground clicks over the image so that he would get paid. This worker did the entire set of tasks in less than 5 minutes, whereas it takes from 30 to 60 minutes to a honest user.
- Remaining workers, a.k.a. "The expert workers", only placed a few well-positioned clicks, and made a few mistakes due to insufficient attention. These workers exhibit statistics (in number of clicks and error rate) that are comparable to expert users.

Fig. 10 6 sorts of workers: "The painter", "The mirror", "The border guard", "The surrounder", "The spammer" and "The expert".

The main lesson of this categorization of user profiles is that, except for worker # 19 who was just a spammer, the highest number of errors come from users who did not understand the job properly. This could have been avoided, or at least limited, with a better tutorial on gold standard images that would have taught workers what is a good click and what is a bad click. Nevertheless the collected data was processed to address a realistic scenario in which crowdsourced workers do not understand or even try to understand the provided instructions.

5.3 Filtering low quality workers with gold standard tasks

Section 5.2 has shown that while experts present in general a uniform and acceptable error rate on generated clicks, workers tend to offer a much more heterogeneous performance, resulting in some cases in completely misleading interactions. For this reason, this current section presents strategies to detect and discard these low quality workers.

The only data we can use to filter workers are the traces on the gold standard images, i.e., the five PASCAL images introduced to serve as a control dataset. Figure 11 displays the error rate per user on the gold standard dataset (in blue) and on the test dataset (in red).

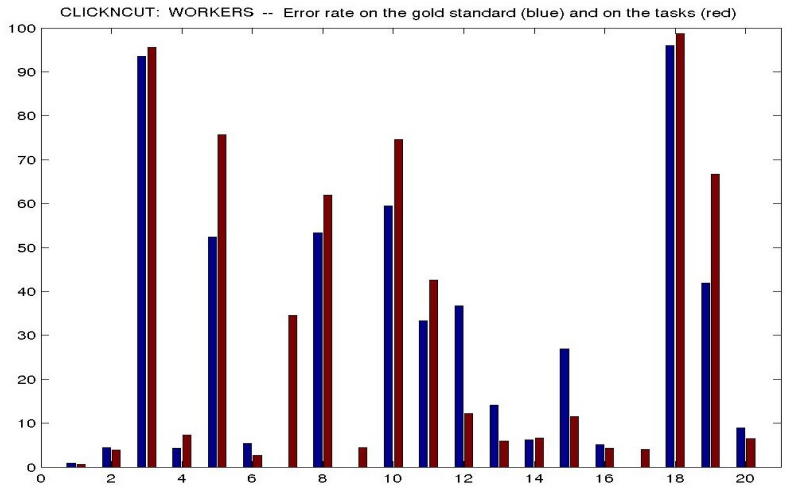

Fig. 11 Error rate on the gold standard image (in blue) and on the tasks (in red) for each worker

The good news is that there is an obvious correlation between error rates on the gold standard and on the test set. Of course the correlation is not perfect; for example worker # 7 made no mistakes on the gold standard set, but on the contrary made more than 30% of mistakes on the test set.

Nevertheless, we can filter a considerable amount of errors by just removing the workers that are above a threshold of error rate on the gold standard. In Figure 12, we vary the threshold that we use to filter users based on the gold standard images. The blue curve (resp. red curve) represents the error rate of the remaining users on the gold standard (resp. on the test set).

In the next section we will therefore consider two thresholds based on this graph. First we will keep only the users who did less than 50% errors on the gold standard: that makes the total error rate less around 20% on the gold standard. We will also try the smaller threshold of 20%, which makes the total error rate less than 10% on the gold standard (and on the test data as well).

5.4 Object segmentation results

In this section we compute the segmentation results in our experiments. We study the importance of filtering the data to improve the results.

Table 2 presents the results on the three experiments: Click'n'Cut with experts, Click'n'Cut with workers, and finally Ask'n'Seek. We use the Jaccard index as a

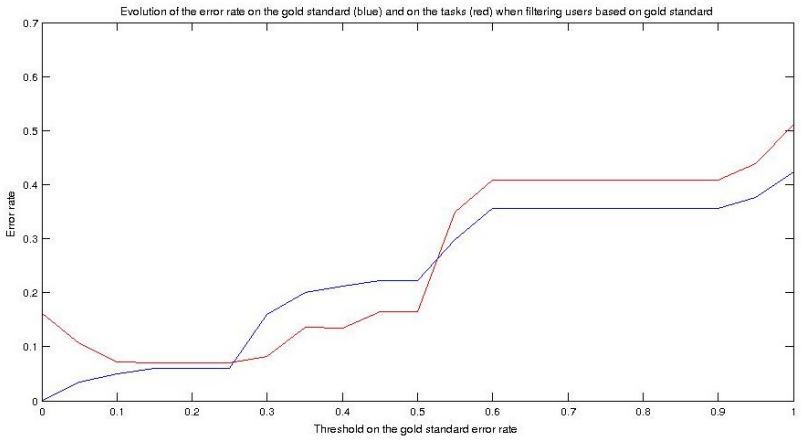

Fig. 12 Evolution of the overall error rate on gold standard images (in blue) and on the tasks (in red) when filtering users based on a threshold on the gold standard error rate.

	Click'n'Cut Experts	Click'n'Cut Workers	Ask'n'Seek Players
All users	0.89	0.14	**0.44**
Users with less than 50% errors on GS	0.89	0.63	0.43
Users with less than 20% errors on GS	0.89	**0.82**	0.40

Table 2 Average Jaccard Index on the test dataset in the three experiments.

measure of the segmentation precision (the Jaccard Index is defined as $J = \frac{P \cap GT}{P \cup GT}$ between the Predicted (P) and Ground Truth (GT) masks).

The experts results are computed for each expert separately, and then averaged over all experts. The results thus mean that one expert will obtain an average of 0.89 Jaccard on each task. Note that filtering experts based on their gold standard performances does not make a lot of sense, since all experts have an error rate below 10% on the gold standard.

There are many things to be said about these numbers. First and not surprisingly, the experts obtain the best segmentation score. This is understandable because they are fully aware that they are performing segmentation (unlike Ask'n'Seek players), and they already know what is a good segmentation and what are the main difficulties to obtain it. In other words, their experience help them to focus on more meaningful regions to click on than workers for example.

Workers' results are very dependent on the filtering based on the gold standard images. The results range from 0.14 without filtering (which is very bad) to 0.82 when considering only users with a low error rate on gold standard images. On the other hand, Ask'n'Seek results are very low compared to Click'n'Cut experiments.

Figure 13 shows the Jaccard index that can be obtained with different amounts of clicks from the three experiments. It can be observed that Click'n'Cut users are able to achieve a significantly better object segmentation result compared to Ask'n'Seek players. What is interesting from this graph is the evolution of the Jaccard index

for the Ask'n'Seek traces, which does not seem to improve with the amount of clicks. This is due to the big difference in the distribution of clicks in Ask'n'Seek and Click'n'Cut, which will be discussed later in this paper.

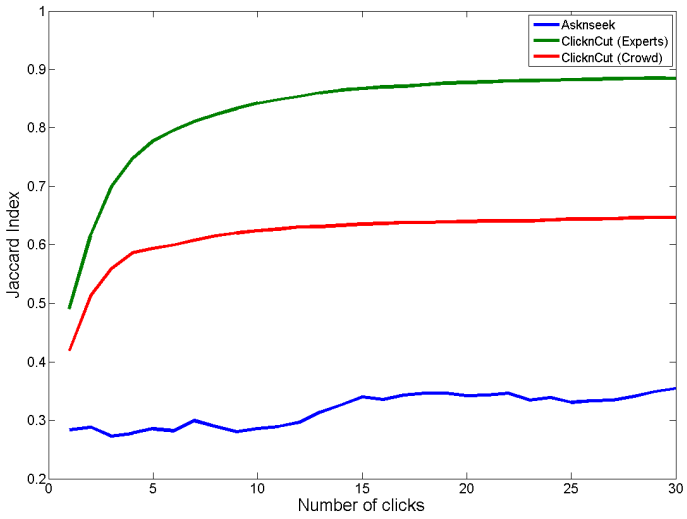

Fig. 13 Jaccard Index vs. the number of clicks used for segmentation.

5.5 The crowdsourcing loss

The results introduced earlier reveal several facts that could be characterized as crowdsourcing loss, i.e. a loss induced by having a task performed by workers instead of experts.

First, the data is more noisy with workers than with experts. This can be visualized in Figure 14: green and red points are mingled in the central row (workers) and much more sparse on the top row (experts). This is caused by many factors: spammers, workers who do not understand the task, a lower attention level, etc. This already underlines a key message: crowdsourcing tasks must be carefully designed and include quality controls to detect errors.

Second, most of the best workers from the crowd still perform worse than average experts. The bottom row of Figure 14 depicts the collected clicks collected from the filtered crowdworkers and we can see that, though the clicks are free of noise, they are also sparser than the expert clicks.

In fact when we compute the jaccard index of each crowd worker, we observe that the workers we categorize as experts in section 5.2 perform worse than the average expert. Nine "expert workers" reach a jaccard index between 0.74 and 0.86 which is

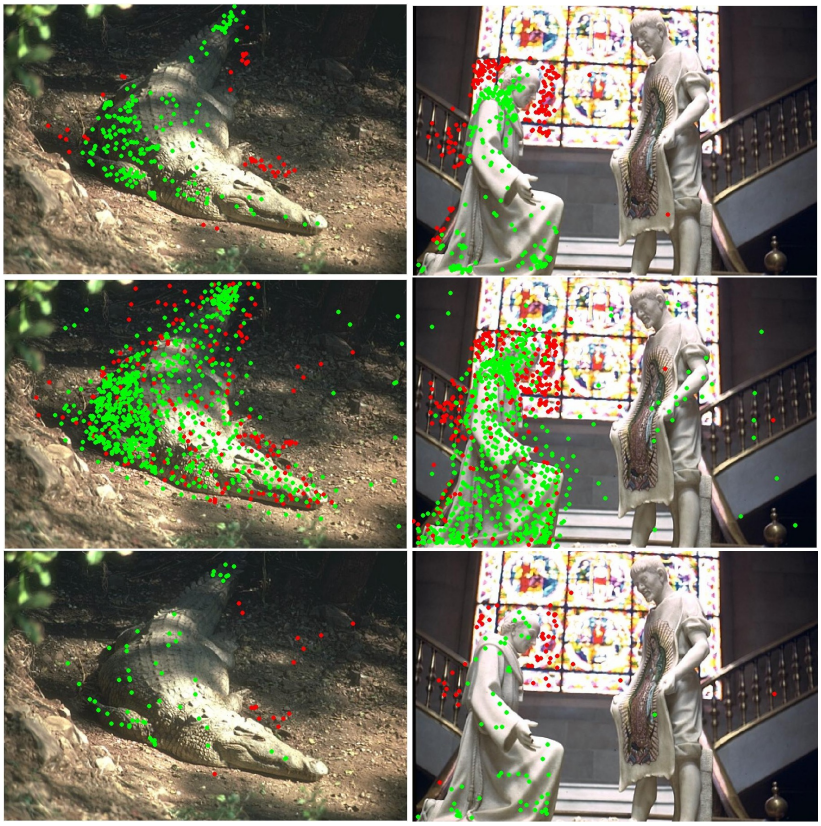

Fig. 14 Spatial distribution of the clicks (foreground in green, background in red) from the experts (top), from the workers (middle), and from the workers categorized as experts according to Section 5.2 (bottom).

below the average (0.89, see table 2) of the real experts. Only two workers can really compare to the real experts: they reached a 0.89 and 0.90 jaccard index.

Figure 14 also shows that there are areas in which users (expert and crowd) tend to click more. In our experience, these high density areas are located on regions where the underlying superpixels segmentation fails to follow the object boundaries. As such, high density areas convey a lot of information: they can help improving the superpixel segmentation for example. Filtering users, as we can see on the figure, makes these high density areas disappear.

5.6 The gamification loss

This section aims at discussing why the segmentation results obtained with the Ask'n'Seek game are poorer than the ones obtained with the Click'n'Cut interactive segmentation tool.

A first very simple reason for Ask'n'Seek's performance is the number of clicks gathered through the game, as discussed in Section 5.1. It is important to state here that one of the limitations of our current approach is the difficulty of processing the free text entered by the seeker when asking for a clue (see Section 3.2). Even by doing it manually there are many labels that remain hard to categorize, either because they are not precise enough or because they are not understandable enough. It would be interesting to study simple ways (e.g., autocompletion, limited vocabulary, and so forth) to make the natural language processing more straightforward.

Fig. 15 Spatial distribution of the clicks (foreground in green, background in red) from the Ask'n'Seek players.

The second reason of Ask'n'Seek's poor performances is the spatial distribution of the seekers' clicks. Figure 15 shows the foreground and background clicks gathered through Ask'n'Seek on two images from our dataset. On the left image, the entire soldier that stands on the right should be segmented. We can see that all the foreground clicks are focused on the soldier's head. We can make a similar observation on the right image: the foreground clicks are concentrated into the duck's head whereas the entire duck should be segmented in our case.

What is even more interesting is the spatial distribution of the background clicks. We can see that on Ask'n'Seek traces, the background clicks are mostly located on other objects, which is understandable: the seekers know that the target region is often located on a salient object, so their clicks are focused on objects (other soldiers, or other ducks in Figure 15). When we look at Figure 14, we can see that background points obtained through Click'n'Cut are almost always located near the object's boundaries. In other words, the gamification loss is a direct consequence of the nature of the game itself: the players know that the most efficient strategy to win in Ask'n'Seek is to place the target region on an object, and preferably on a salient part of the object (e.g., human's face). This is probably the key reason that explains the gamification loss in Ask'n'Seek.

6 Future Work

The obtained results provide several opportunities for research in order to reduce the crowsourcing and gamification losses for object segmentation introduced by our tools Click'n'Cut and Ask'n'Seek.

A common aspect to improve is a more accurate user categorization in order to improve any collected feedback. As discussed in Section 5.2, erroneous feedback can have different sources and discarding users completely may be a too drastic solution. For this reason, we foresee an automatic categorization of users to feed different object segmentation algorithms depending on the type of collected feedback.

Regarding Click'n'Cut, we have seen that most errors from paid workers are due to a misunderstanding of the job. Even though we see the potential of using the variety of traces produced by these workers, we believe that adding a tutorial at the beginning of the experiment would allow users to understand the task better and it would simplify the challenge of filtering errors.

Regarding Ask'nSeek, our analysis on Section 5.6 encourages the definition of an active learning setup for the game. Given that Ask'nSeek traces are biased towards the most prominent objects (and object parts) on the images, a saliency estimator could determine where to place the target region on an image in order to gain as much information as possible from the game logs. This may allow biasing the master's choice of the target region (e.g., by granting more points if the master follows the system's advice) in order to gather more informative traces through the Ask'nSeek game.

In addition, one of the major limitations of Ask'nSeek is related to the text processing of the collected tags. In this work these tags have been manually selected and clustered, while a more realistic approach should be able to perform this task automatically by resolving disambiguation problems and identifying synonyms or related terms.

Another possible avenue for improvements would be to question the choice of using MCG candidates in our system. We could potentially include any object candidate algorithm (many more exist), as well as any segmentation algorithm to generate the candidates.

7 Conclusion

In this paper, we have presented and studied the crowdsourcing and gamification losses in a computer vision problem such as object segmentation. This work has been carried on by using the Click'n'Cut interactive segmentation tool and the Ask'n'Seek game, both accessed online by users. The study has considered three different groups of users: experts and workers on Click'n'Cut, and players on Ask'n'Seek. Not surprisingly, the experts who use Click'n'Cut produced the best results. The crowd of workers produce very noisy inputs, but we have shown how a simple filtering method based on gold standard images can bring acceptable results. Finally, results obtained through Ask'nSeek are poor and significantly worse than the results obtained through Click'n'Cut.

We have put special attention in analyzing the loss induced by having a crowd of paid workers to perform an object segmentation task, when this crowd is compared to computer vision experts. We have seen that the workers are less efficient than expert users in positioning their clicks in meaningful areas. However, this loss could be compensated by a better of use of the diversity of workers' profiles that actually produce a high amount of clicks that are wrong, but that are still informative.

Conversely, Ask'nSeek clicks are not very informative because they are very redundant. Due to the nature of the games, players are biased towards positioning their clicks on objects which is not the best strategy for our segmentation algorithm. The challenge we will have to face in future work will be to modify the gameplay in order to encourage a higher diversity of clicks' positions.

To end this paper, we would like to put the emphasis once more on the workers categorization depicted in figure 10. Among the different categories of workers we have extracted, it is interesting to note that the traces that were really useful to our algorithm are the one provided by the "expert" workers. On the other hand the traces from the "painter", the "surrounder" and the "border guard" have not beed used properly, though they carry a lot of useful information for the segmentation task. Coming up with a smart way of using this information could help overcoming the current best performances in interactive segmentation. This indicates an interesting research avenue that, to our knowledge, may have not been explored enough : introducing several different types of interaction to bring complementary information that would work together to achieve the perfect segmentation.

Acknowledgements This work has been developed in the framework of the project TEC2013-43935-R, financed by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF).

A Images used in our experiments

We used the five images taken from the PASCAL VOC dataset (see Fig. 16) as Gold Standard in our experiments.

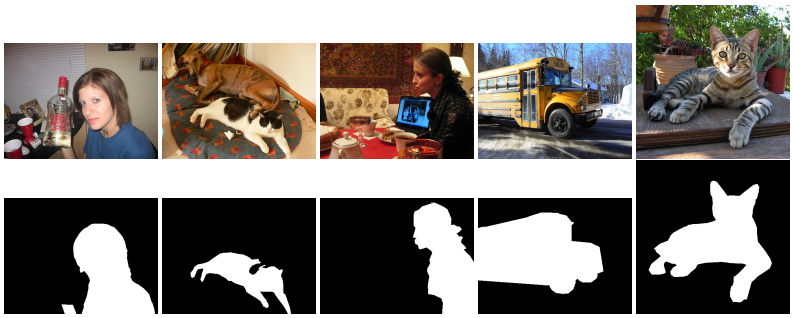

Fig. 16 Original images (top) and ground truth segmentation masks (bottom) of the 5 images we used in our exeperiments as Gold Standard.

The textual descriptions that were provided to the users during our experiments were :

- Extract the woman from the image. Include her hair, her clothes, and the part of her arm that holds the bottle.
- Extract the cat from the image. Try to discard the dog's paw laying on the cat.
- Extract the woman from the image. Include her hair.
- Extract the bus from the image. Do not include the mirrors on the front of the bus.
- Extract the cat from the image.

References

1. Y. Sun, Y. CHen, W. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2014.
2. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary map; region segmentation of objects in n-d images," in *ICCV*, 2001.
3. C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, Aug. 2004.
4. J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, vol. 24, no. 3, Jul. 2005.
5. A. Noma, A. B. V. Graciano, R. M. Cesar Jr, L. A. Consularo, and I. Bloch, "Interactive image segmentation by matching attributed relational graphs," *Pattern Recogn.*, vol. 45, no. 3, Mar. 2012.
6. K. McGuinness and N. O'Connor, "Improved graph cut segmentation by learning a contrast model on the fly," in *ICIP*, 2013.
7. H. S. Lee, J. Kim, S. J. Park, and J. Kim, "Interactive segmentation as supervised classification with superpixels," in *WCVPR 2014-W. on Computer Vision and Human Computation*, 2014.
8. P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE T. on Image Processing*, vol. 9, no. 4, 2000.
9. T. Adamek, "Using contour information and segmentation for object registration, modeling and retrieval," Ph.D. dissertation, Dublin City University, 2006.
10. P. Arbelaez and L. Cohen, "Constrained image segmentation from hierarchical boundaries," in *CVPR*, 2008.
11. X. Giró-i Nieto, M. Martos, E. Mohedano, and J. Pont-Tuset, "From global image annotation to interactive object segmentation," *MTAP*, vol. 70, no. 1, 2014.
12. K. McGuinness and N. E. O'connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.
13. J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *CVPR*, 2010.
14. T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *PAMI*, vol. 33, no. 2, 2011.
15. P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014.
16. A. Salvador, A. Carlier, X. Giro-i Nieto, O. Marques, and V. Charvillat, "Crowdsourced object segmentation with a game," in *ACM CrowdMM*, 2013.
17. A. Carlier, V. Charvillat, A. Salvador, X. Giro-i Nieto, and O. Marques, "Click'n'cut: Crowdsourced interactive segmentation with object candidates," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, ser. CrowdMM '14. New York, NY, USA: ACM, 2014, pp. 53–56. [Online]. Available: http://doi.acm.org/10.1145/2660114.2660125
18. J. Steggink and C. Snoek, "Adding semantics to image-region annotations with the name-it-game," *Multimedia Systems*, vol. 17, 2011.
19. B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *IJCV*, vol. 77, no. 1-3, 2008.
20. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *CoRR*, 2014.
21. S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Opensurfaces: A richly annotated catalog of surface appearance," *ACM TOG*, vol. 32, no. 4, 2013.
22. L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun, "Beat the mturkers: Automatic image labeling from weak 3d supervision," in *CVPR*, 2014.

23. S. D. Jain and K. Grauman, "Predicting sufficient annotation strength for interactive foreground segmentation," in *ICCV*, 2013.
24. D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proceedings of CVPR'10*, 2010, pp. 3169–3176.
25. L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *ACM CHI*, 2004.
26. L. von Ahn, R. Liu, and M. Blum, "Peekaboom: a game for locating objects in images," in *ACM CHI*, 2006.
27. M. Lux, A. Müller, and M. Guggenberger, "Finding Image Regions with Human Computation and Games with a Purpose," in *AIIDE*, 2012.
28. A. Carlier, O. Marques, and V. Charvillat, "Ask'nseek: A new game for object detection and labeling," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 249–258.
29. D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.
30. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, jun 2010.
31. D. Oleson, A. Sorokin, G. P. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing." *Human computation*, vol. 11, p. 11, 2011.