Assessment of crowdsourcing and gamification loss in user-assisted object segmentation

Axel Carlier, Amaia Salvador, Ferran Cabezas, Xavier Giro I Nieto, Vincent Charvillat, Oge Marques

Assessment of crowdsourcing and gamification loss in user-assisted object segmentation

Axel Carlier[1] · Amaia Salvador[2] · Ferran Cabezas[2] · Xavier Giro-i-Nieto[2] · Vincent Charvillat[1] · Oge Marques[3]

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 an 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

✉ Axel Carlier
Axel.Carlier@enseeiht.fr

Amaia Salvador
amaia.salvador@upc.edu

Ferran Cabezas
ferran.cabezas@upc.edu

Xavier Giro-i-Nieto
xavier.giro@upc.edu

Vincent Charvillat
vincent.charvillat@enseeiht.fr

Oge Marques
omarques@fau.edu

1 IRIT-ENSEEIHT, University of Toulouse, Toulouse, France

2 Universitat Politecnica de Catalunya (UPC), Barcelona, Catalonia, Spain

3 Florida Atlantic University (FAU), Boca Raton, FL, USA

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 crowd-workers, 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 assessment 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 on 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 [34], 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 as the image, 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, Fig. 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 (Fig. 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

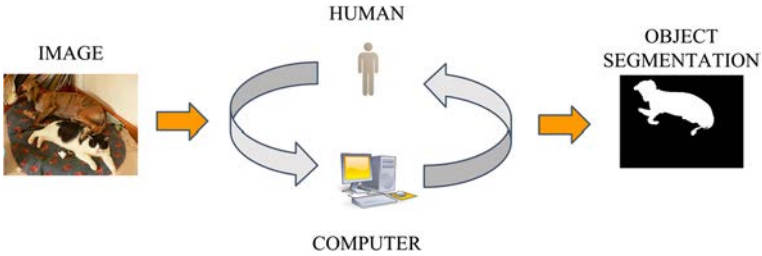


Fig. 1 Introducing the human "in the loop" in object segmentation

are even guided by these algorithms to produce clicks 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 feel 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.

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 describes the experiments performed using *Ask'nSeek* and *Click'n'Cut* and discusses the most relevant results. Finally, Section 5 suggests directions for future work and Section 6 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* [6]. 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 [27], and its annotation mode mode (bounding boxes, sloppy contours or tight polygon) predicted for each image on [15].

Superpixel-based solutions [16, 22, 24, 37] 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 [1, 2, 14, 31] use these data structures to propagate labels not through a flat graph partition but, instead, considering regions at multiple scales. The comparative study in [23] indicated similar accuracy labels for GrabCut [27] and hierarchical solutions [1, 31], but a faster response for the latter ones.

More recent solutions have used convolutional neural networks to extract object segments from minimal user interaction. A single user click is used in [26, 29] to condition the loss function of a network which had been previously trained for the semantic segmentation of an image [19]. These work though does not explore though an interactive framework, as only one click is collected by user and no visual feedback is provided to augment the informativeness of the click.

The solution adopted in our work generates segmentations by combining a precomputed set of object candidates (also referred to as "saliency detectors"). Instead of considering precomputed 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 [3, 10, 18] 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 [32], 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 [32], because it obtains solutions that can combine multiple candidates. Our *Click'n'Cut* system has been introduced in [9] and uses this combination of object candidates.

Crowd-based object annotation There have been several approaches to elicit object annotation data from the crowds in the literature. Some authors [32, 33] 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* [30], 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 [17]. 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 [5], an open source tool based on polygonal segmentation.

The authors of [11] 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 [15] 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 [23] 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 [4]. 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. The same active learning paradigm is followed in [28]. In this paper, the authors introduce a system relying on oversegmentations that requires clicks from users in order to obtain an image segmentation. Active learning has also been used in [13] for video segmentation. 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 [35] 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* [36]. This platform is the natural evolution of the popular ESP Game from the same authors [35], 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* [33] 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 [30]. 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* [20], 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* [32] 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.

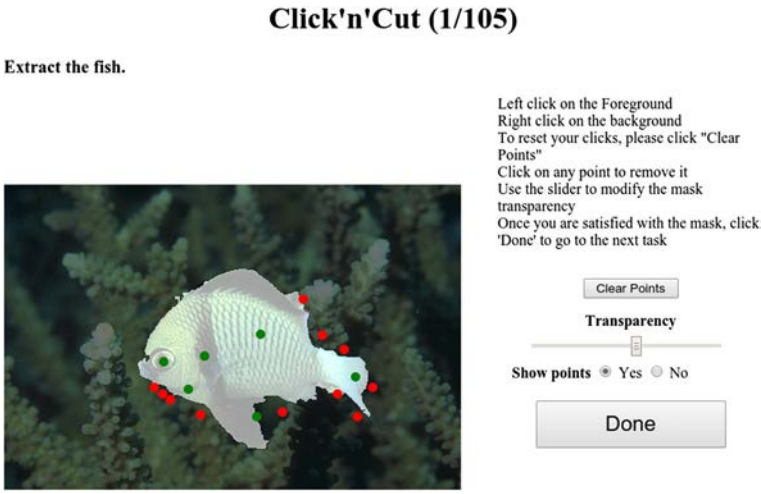


Fig. 2 Screenshot of the Click'n'Cut interface

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. *Intentional* interaction has been collected with *Click'n'Cut* [9], 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 [8], described in Section 3.2.

3.1 Click'n'Cut interactive segmentation tool

In *Click'n'Cut* [9], 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 using an algorithm based on object candidates, introduced in [3], and aims at guiding the user to provide information (i.e., meaningful clicks) that will actually help improving the quality of the final segmentation result.

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.

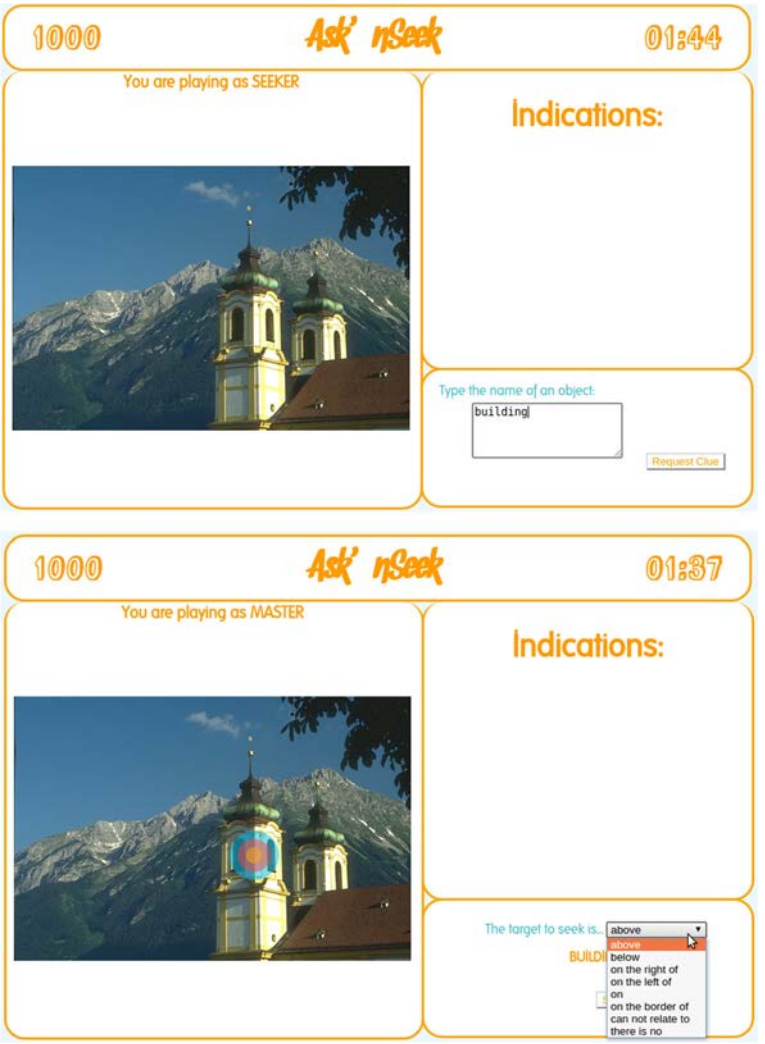


Fig. 3 Screenshots of the Ask'nSeek game: seeker's screen (*top*); master's screen (*bottom*)

3.2 The Ask'nSeek game

Ask'nSeek [8] 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* (Fig. 3, bottom) hides a target region somewhere within a randomly chosen image. The second player (*seeker*) (Fig. 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

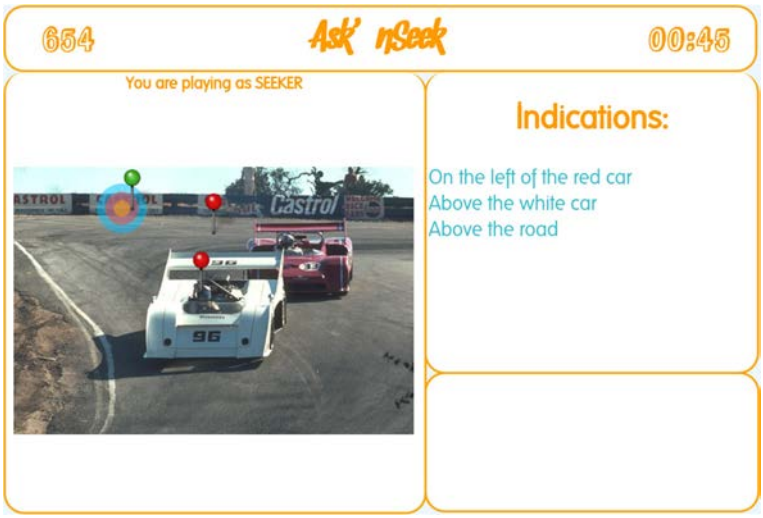


Fig. 4 Example of the game from the seeker's point of view

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 red car, and the master answered that the region is "on the left of the red car". The seeker clicked on the image but not on the region, so he asked for a second clue, relative to the white car. The master answered that the region is "above the white car", and the seeker once again did not find the region. He finally got the indication that the region is "above the road", 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 not see it). In other words, the target only appears on the image after the seeker has managed to click inside it.

The game is played *cooperatively*, which provides a way to encourage both players to locate the hidden region as quickly, and with as few clicks, as possible. To control for time, a timer (set to 2 minutes) is provided, which requires that the hidden object be found within this interval in order to win. Moreover, the score of both players decreases after each new click, which encourages the players to be precise in their request for clues (and associated responses) to minimize the number of clicks (and, indirectly, shorten the duration of the game).

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.

4 Experiments and results

In this section we discuss the results of a series of experiments using the interfaces described in Section 3. From the data collected during these experiments, we examine and quantify the *crowdsourcing loss* incurred when an interactive segmentation task is assigned to paid workers recruited on a crowdsourcing platform, comparing their results to the ones obtained when computer vision experts are asked to perform the same tasks. 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 [23] in their work on interactive segmentation which makes our results comparable with the results from that study. The collection contains 96 images selected from the larger Berkeley Segmentation Dataset [21], and also includes the ground truth binary masks for 100 objects (two 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 five images from the PASCAL VOC [12] segmentation dataset and have written a textual description of one object per image (see Appendix A) . 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 Southeast 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'nSeek - Players**: 162 players (mostly students) played the Ask'nSeek game on any number of images they wanted to.

4.1 Preliminary figures

Table 1 presents a preliminary comparison of relevant figures for the three groups of users. It is worth mentioning that paid workers produced a substantial number 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.

Another interesting difference among 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'nSeek

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

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

in this percentage, as 1) they represent a minority of clicks, and 2) they are a specificity of Ask'nSeek, therefore not comparable to Click'n'Cut. A very important finding derived from these early results 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 two players to produce a click in Ask'nSeek (the master and the seeker) whereas only one user is necessary in Click'n'Cut. Secondly, the users in Ask'nSeek played an average of 30 games (i.e. images) each, whereas Click'n'Cut users performed the entire set of 105 tasks. Moreover, Click'n'Cut users were given the possibility to produce as many clicks as they wanted to, whereas Ask'nSeek players are limited by a 2-minute 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'nSeek, 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.

This preliminary set of results also highlights the difference in number of errors performed by workers on Click'n'Cut when compared to Ask'nSeek players. As we will see in Section 4.2, the high error rate of the workers using Click'n'Cut is partly caused by a specific subset of workers who performed particularly poorly. Ask'nSeek players error rate is much more evenly spread across users (game players). The fact that Ask'nSeek is a game naturally limits the impact of some of the usual sources of errors in crowdsourcing, namely: the spammers, the incompetent workers, and workers with insufficient attention [25]. Being a game, Ask'nSeek 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'nSeek 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'nSeek is the misunderstanding between the master and the seeker. Misunderstanding can arise from an imprecise requested object from the seeker (e.g. "dog" in an image where there are three dogs), or from the master not understanding a word used by the seeker.

4.2 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 in order to explore the details behind the *crowdsourcing loss* suffered between the results of the experts and the paid workers.

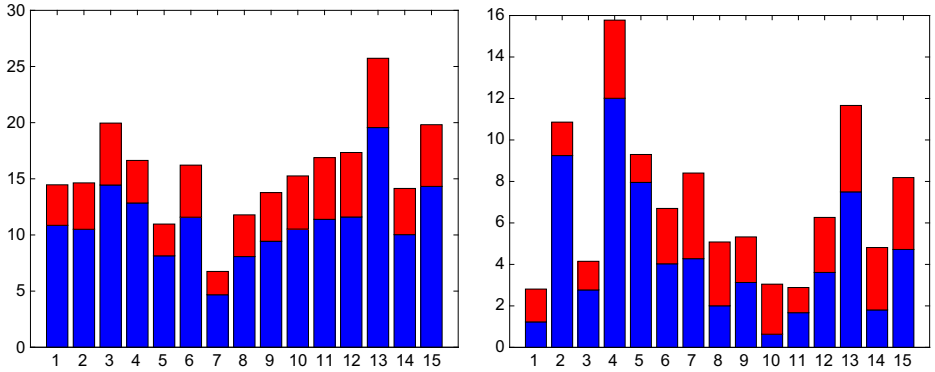


Fig. 5 On the *left*, number of foreground (*blue*) /background (*red*) clicks on the left and on the right, percentage of foreground/background errors per expert user on Click'n'Cut

Figure 5 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; interestingly enough, the proportion of foreground (blue)/background (red) clicks for each expert user is fairly similar, regardless of the image.

The right graph of Fig. 5 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 worth noting that the expert's highest source of errors seems to be the foreground clicks.

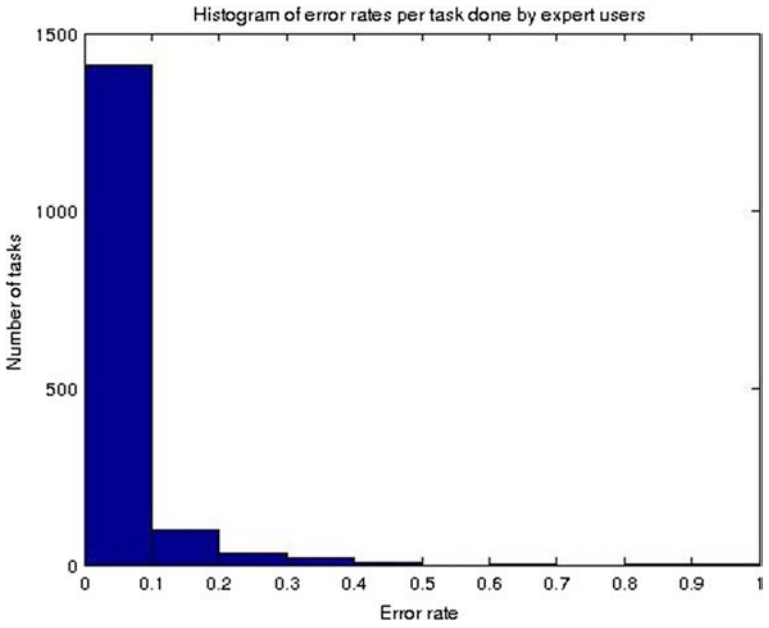


Fig. 6 Histogram of the error rates per task

Fig. 7 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*)

To further understand this phenomenon, 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 Fig. 6, on which we can see that a very large majority of the tasks had a very low error rate.

Figure 7 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, which suggests that 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).

Figure 8 shows the same type of plot as in Fig. 5 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

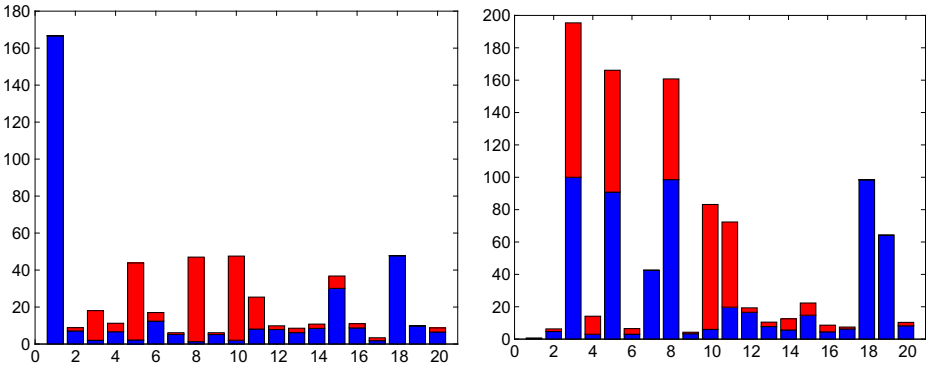


Fig. 8 On the left, number of foreground (*left*) /background (*right*) clicks; on the right, percentage of foreground/background errors (*right*) per paid worker on Click'n'Cut

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 Fig. 8 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 (Fig. 5) and workers (Fig. 8) 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-hand-side of each figure. This observation has led us to propose a categorization of the workers, inspired by [25], and exemplified in the cases contained in Fig. 9:

- Worker # 1, a.k.a. the "painter", only used 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/she had to entirely paint the object with green clicks.
- Worker # 3, a.k.a. the "mirror", has such a high error rate that by inverting his/her contributions (i.e. considering his/her background clicks as foreground, and vice versa), he/she would actually display a very low error rate. We can only assume that he/she misunderstood the instructions, confusing foreground and background clicks.
- Worker # 5 and # 8, 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 a similar way as requested in LabelMe [30].
- Worker # 19, a.k.a. the "spammer", randomly placed foreground clicks over the image so that he would get paid. This worker completed the entire set of tasks in less than 5 minutes, whereas it took 30 to 60 minutes to a honest user.
- Workers # 7, # 9 and # 17, a.k.a. the "tired", had used a very limited number of clicks but, unlike the "spammer", most of the clicks are correct.
- Workers # 2, # 4, # 6, # 12, # 13, # 14, # 15, # 16, # 20, a.k.a. "the expert workers", understood the task and completed it carefully. These workers exhibit statistics (in number of clicks and error rate) that are comparable to expert users.
- Workers # 10 and # 11 can not be categorized as only one type of users. They changed their pattern of interaction during their session. For example, worker # 11 started as "a border guard", and probably understood the interface after a while and ended up interacting as an expert.

Building up on the observations we have made to establish our categorization, we manually defined some criteria to automatically categorize users based on the following statistics:

- Average number of clicks per task `#cli`
- Percentage of foreground clicks (i.e. left clicks) `%FG`
- Median percentage of errors per task `%err` (We chose to use the median because, as we have shown in Fig. 6, a small amount of tasks can have a drastic effect on the average error rate. The median is more robust to this small amount of outlying tasks).
- Average Jaccard index per task `J`, described below.
- Average time spent per task `t`
- Percentage of clicks that lie near the contour of the object `%Cont`

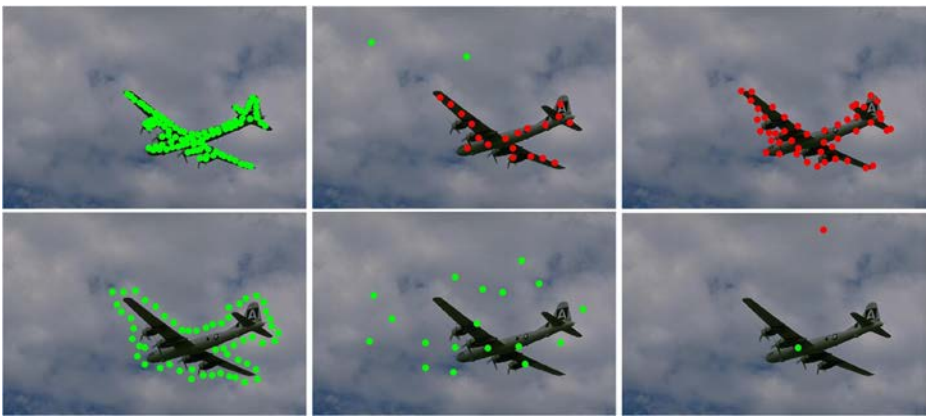


Fig. 9 Six types of workers: (*top*) the "painter", the "mirror", and the "border guard"; (*bottom*) the "surrounder", the "spammer" and the "tired"

We use the Jaccard index as a 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.

We then define the set of following rules that will help us categorize users:

- **Rule 1**: A "painter" is a worker who clicked more than 100 times per task with more than 90 % foreground clicks (`#cli` $>= 100 \cap$ `%FG` $>= 90$).
- **Rule 2**: A "mirror" is a worker who has a median error rate greater than 90 % with less than 50 % clicks on the contour (`%err` $>= 90 \cap$ `%Cont` $<= 50$).
- **Rule 3**: A "border guard" is a worker who clicked more than 50 % of the time on the contour with less than 50 % foreground clicks (`#Cont` $>= 50 \cap$ `%FG` < 50).
- **Rule 4**: A "surrounder" is a worker who clicked more than 50 % of the time on the contour with more than 50 % foreground clicks (`#Cont` $>= 50 \cap$ `%FG` $>= 50$).
- **Rule 5**: A "spammer" is a worker who spent less than 10 seconds per task with more than 20 % errors per task (`t` $<= 10 \cap$ `%err` $>= 20$).
- **Rule 6**: A "tired" is a worker who spent less than 10 seconds per task with less than 20 % errors per task (`t` $<= 10 \cap$ `%err` < 20).
- **Rule 7**: An "expert worker" is a worker who recorded an average Jaccard Index greater than or equal to 0.80 (`J` $>= 0.8$).

Note that, in order to be mutually exclusive, the workers defined in Rules 1 to 6 should record an average Jaccard Index lower than 0.80. If a worker does not fit any of these categories, he/she is labelled as "others".

We used the rules above to categorize users based on their statistics on the Gold Standard images. Table 2 summarizes our results:

Looking at Table 2, it can be seen that we mis-categorized three workers (out of 20), namely workers #10, #11, and #15. The main reason for these three errors was the "others" category, which currently designates users who drastically change their interaction pattern during the study. The results on Table 2 suggest that five Gold Standard images may not be sufficient to evaluate users and predict to which categories they belong with 100 % accuracy.

In any case, he main lesson of this categorization of user profiles is that, except for worker #19, who was just a spammer, the highest number of errors came from users who did not understand the job properly. This could have been avoided, or at least limited, with a

Table 2 User categorization obtained on the gold standard images versus ground truth user categorization

Category	Ground Truth	Categorization based on Gold Standard images
Painter	#1	#1
Mirror	#3	#3
Border Guard	#5, #8	#5, #8, #10, #11
Surrounder	#18	#18
Spammer	#19	#19
Tired	#7, #9, #17	#7, #9, #17
Expert Worker	#2, #4, #6, #12, #13, #14, #15, #16, #20	#2, #4, #6, #12, #13, #14, #16, #20
Others	#10, #11	#15

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.

4.3 Filtering low quality workers with gold standard tasks

Section 4.2 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. This section presents principled 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 10 displays the error rate per user on the gold standard dataset (in blue) and on the test dataset (in red).

There is a strong correlation between error rates on the gold standard and on the test set. A notable exception is worker #7, who made no mistakes on the gold standard set, but still showed an error rate greater than 30 % on the test set.

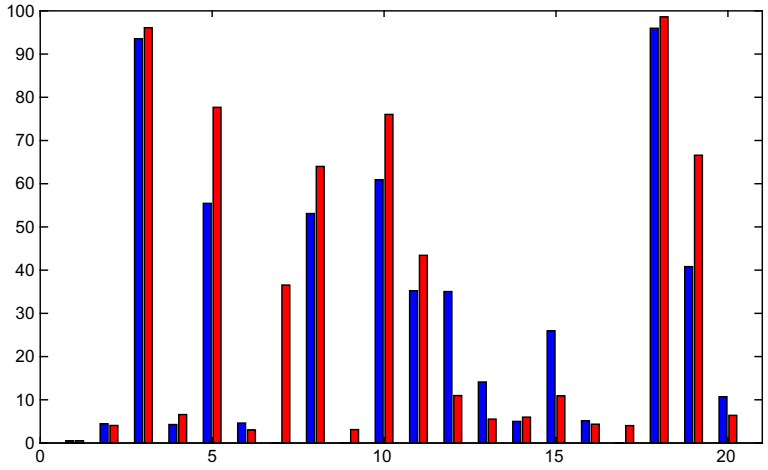


Fig. 10 Error rate on the gold standard image (*in blue*) and on the tasks (*in red*) for each worker (from [9])

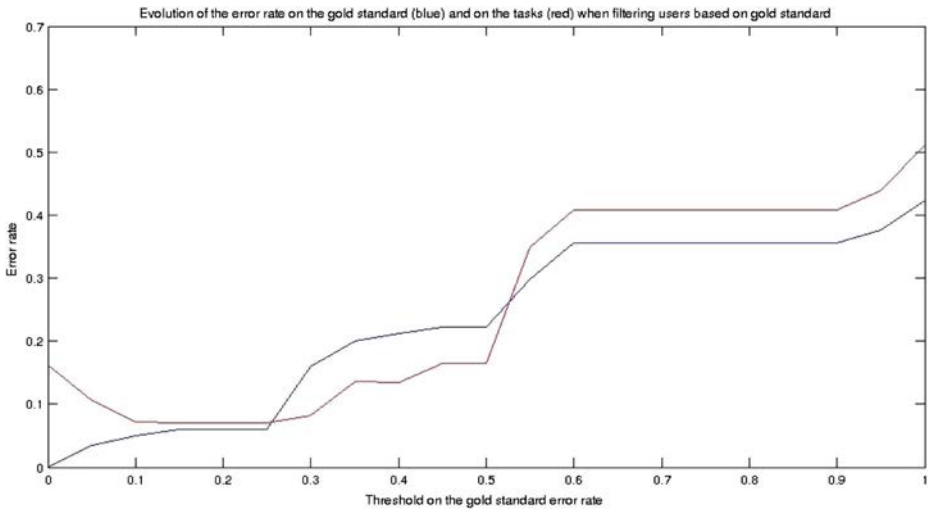


Fig. 11 Evolution of the overall error rate on gold standard images (*in blue*) and on the tasks (*in red*) when filtering users based on a threshold on the gold standard error rate

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 Fig. 11, 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 set of experiments we consider two thresholds based on the graph on Fig. 11. First we will keep only the users who made less than 50 % errors on the gold standard, which makes the total error rate around 20 % on the gold standard. We 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).

Table 3 presents the results on the three experiments: Click'n'Cut with experts, Click'n'Cut with workers, and finally Ask'nSeek. We use the Jaccard index (defined earlier) as a measure of the segmentation precision.

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'nSeek players), and they already

Table 3 Average Jaccard Index on the test dataset in the three experiments

	Click'n'Cut Experts	Click'n'Cut Workers	Ask'nSeek 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

know what is a good segmentation and what are the main difficulties to obtain it. In other words, their experience helps them 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'nSeek results are very low compared to Click'n'Cut experiments.

Figure 12 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'nSeek players. What is interesting from this graph is the evolution of the Jaccard index for the Ask'nSeek 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'nSeek and Click'n'Cut, which will be discussed later in this paper.

To conclude this section, we should add that we have explored more advanced methods for filtering data in another work [7] and showed that workers can reach a Jaccard Index of 0.86 when an appropriate filtering is applied.

4.4 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.

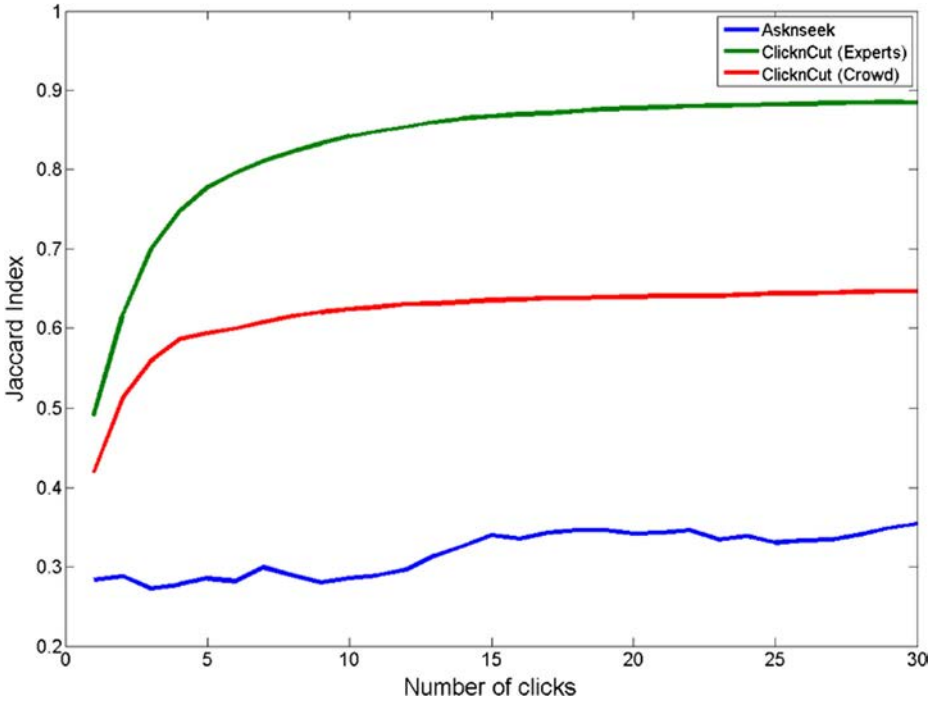


Fig. 12 Jaccard Index vs. the number of clicks used for segmentation

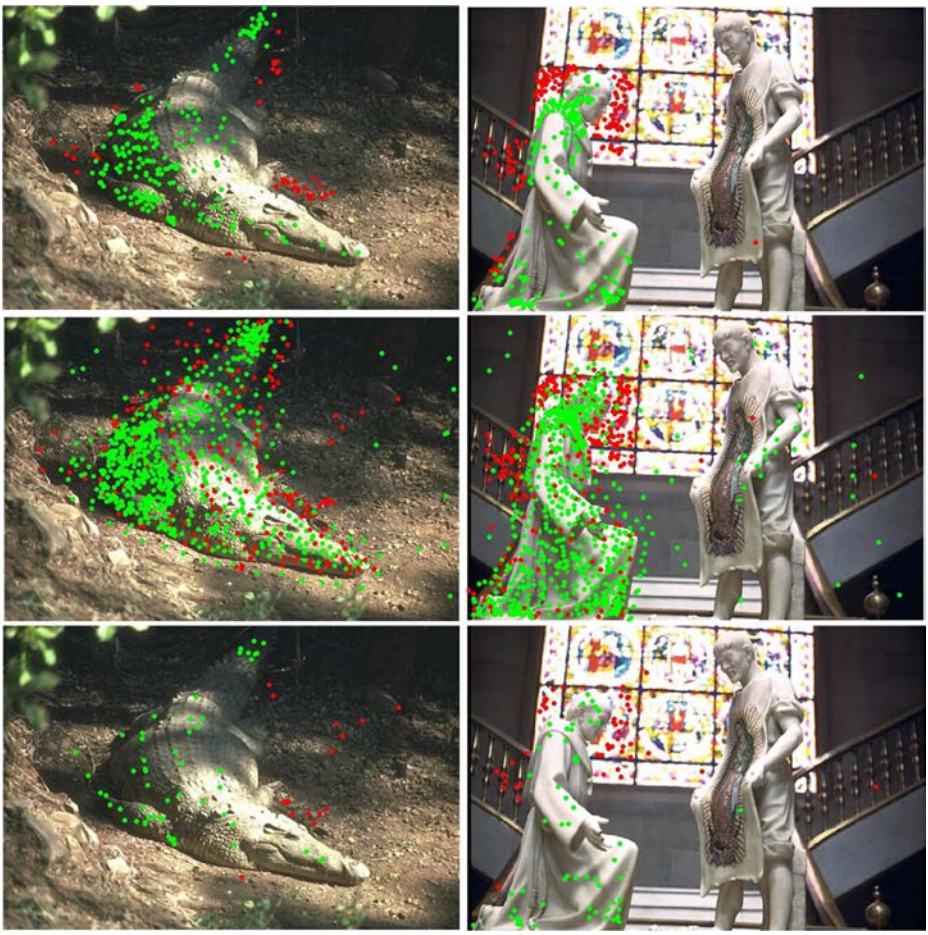


Fig. 13 Spatial distribution of the clicks (foreground in *green*, background in *red*) from the experts (*top*), from the workers (*middle*), and from the workers categorized as experts according to Section 4.2 (*bottom*)

First, the data is noisier with workers than with experts. This can be visualized in Fig. 13: green and red points are densely concentrated 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 Fig. 13 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 expert workers perform worse than the average expert. Eight "expert workers" reach a Jaccard index between 0.80 and 0.86 which is below the average (0.89, see Table 3) of the real experts. Only two workers can really compare to the real experts: they reached a 0.89 and 0.90 jaccard index.

Table 4 Analysis of the experts' interaction patterns

Pattern	Frequency	Average number of clicks in the sequence	Average Jaccard index
FB	18.4 %	5.7 ; 3.9	0.8867
F	18.2 %	8.9	0.8949
FBF	17.8 %	6.7 ; 2.7 ; 8.0	0.8980
FBFB	12.1 %	4.4 ; 3.0 ; 3.6 ; 2.7	0.8907
FBFBF	10.4 %	4.5 ; 2.6 ; 4.2 ; 2.6 ; 5.4	0.8873
FBFBFB	6.6 %	3.9 ; 2.8 ; 3.0 ; 2.4 ; 3.2 ; 3.6	0.8728
FBFBFBF	6.3 %	4.2 ; 2.8 ; 3.2 ; 2.5 ; 5.1 ; 2.2 ; 5.7	0.8693
Others	10.2 %	N/A	N/A

Figure 13 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 produced by the segmentation algorithm fail 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 Fig. 13, makes these high density areas disappear.

In order to understand why expert workers do not perform as well as true experts, we looked at the interaction patterns of these two groups. The results are presented in Tables 4 and 5.

We regroup the interaction by considering the alternating pattern between foreground clicks and background clicks. For example if a user clicked twice on the foreground, and then once on the background, we categorize the interaction as following the FB pattern in the table (F stands for foreground, B for background).

For each pattern, we display the frequency of appearance of this pattern and the average Jaccard Index such pattern led to. We also compute the average number of clicks in the sequence. For example on the second line of Table 4, when experts follow the FB pattern, they start in average with 5.7 foreground clicks and end with 3.9 background clicks in average.

From these tables, we can first say that both experts and expert workers tend to always start with foreground clicks. But an important difference between these two groups of users is that the most frequent pattern among expert workers is to only click on the foreground, which leads them to a 0.8541 Jaccard Index in average. If we compare these numbers to the ones from true experts, we can say that expert workers do not have the same sense of a complete segmentation than experts do. For those cases when experts only click on

Table 5 Analysis of the expert workers' interaction patterns

Pattern	Frequency	Average number of clicks in the sequence	Average Jaccard index
F	28.8 %	8.4	0.8541
FB	22.3 %	5.7 ; 3.3	0.8654
FBF	12.5 %	6.9 ; 2.2 ; 6.9	0.8605
FBFB	8.4 %	6.5 ; 2.6 ; 3.9 ; 2.7	0.8822
FBFBF	5.4 %	6.8 ; 2.8 ; 4.4 ; 2.0 ; 6.1	0.8648
Others	22.6 %	N/A	N/A

Fig. 14 Spatial distribution of the clicks (foreground in *green*, background in *red*) from the Ask'nSeek players

the foreground, they tend to click more (8.9 clicks against 8.4 in average) and obtain a significantly better Jaccard Index (0.8949 against 0.8541).

In addition, experts also tend to have longer interaction patterns than expert workers. We only displayed in both tables the patterns that appear more than 5 % of the time, and the sequences FBFBFB and FBFBFBF respect this constraint among experts, but not among expert workers. This means that expert workers are less inclined to have long interaction sequences, and sometimes stop interacting too soon to achieve a very good Jaccard Index.

To sum up, expert workers do not perform as well as experts mostly because they stop interacting too soon. This could be due to two principal reasons: first they are less conscious of what a good segmentation is, and second they are motivated to complete the tasks as quickly as possible and as a consequence tend to quickly move on to the next task once the result is acceptable.

4.5 The gamification loss

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

A first very simple reason for Ask'nSeek's performance is the number of clicks gathered through the game, as discussed in Section 4.1. 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.

The second reason of Ask'nSeek's poor performances is the spatial distribution of the seekers' clicks. Figure 14 shows the foreground and background clicks gathered through Ask'nSeek 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'nSeek 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 Fig. 14). When we look at Fig. 13, 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'nSeek is to place the target region on an object, and preferably on a salient part of the object (e.g., human's face).

5 Future work

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

A common aspect to improve is a more accurate user categorization in order to improve any collected feedback. As discussed in Section 4.2, 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 4.5 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.

6 Conclusion

In this paper, we have presented and studied the crowdsourcing and gamification losses incurred when attempting to solve a computer vision problem, such as object segmentation, with human users in the loop. This work has been carried on by using the Click'n'Cut interactive segmentation tool and the Ask'nSeek 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'nSeek. Not surprisingly, the experts who use Click'n'Cut produced the best results. The crowd of workers originally produced 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, as a consequence of the nature of the task (playing a game rather than segmenting an object through a series of clicks).

We have given special attention to the analysis of the loss induced by having a crowd of paid workers perform an object segmentation task, when the quality of the segmentation results produced by such crowd is compared to the one obtained when we invited computer vision experts to perform the same tasks. 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 informative nevertheless.

Conversely, Ask'nSeek clicks are not very informative because they are very redundant. Due to the nature of the game, players are biased towards positioning their clicks on objects, which is not necessarily the best strategy for our segmentation algorithm. In the future we plan to modify the gameplay in order to encourage a higher diversity of clicks' positions.

To conclude, we would like to emphasize once more the workers' categorization depicted in Fig. 9. Among the different categories of workers we have identified and labeled, currently the only traces that are useful to our algorithm are the ones provided by the "expert" workers. We do acknowledge, though, that the traces from the "painter", the "surrounder" and the "border guard" carry a lot of potentially useful information for the segmentation task. Devising a smart way of using this information could produce an approach that overcomes the current best performances in interactive segmentation. This suggests an interesting research avenue that, to our knowledge, may have not been explored enough: introducing several different types of user interaction to bring complementary information that would work together to achieve improved object segmentation results.

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

Appendix A: Images used in our experiments

We used the five images taken from the PASCAL VOC dataset (see Fig. 15) as Gold Standard in our experiments. The textual descriptions that were provided to the users during

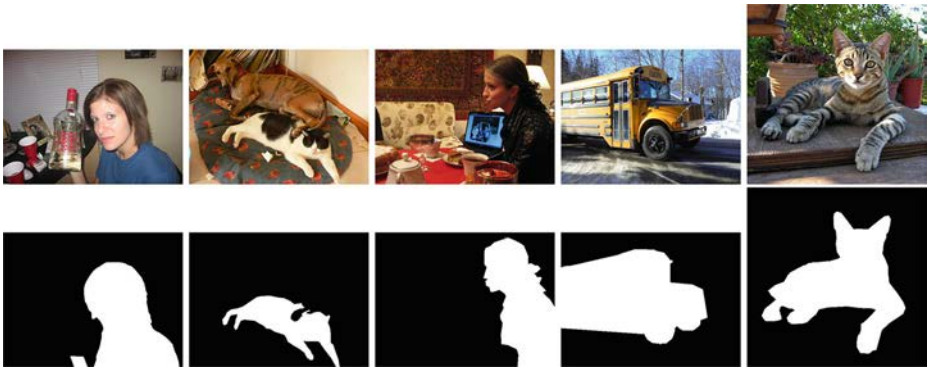


Fig. 15 Original images (*top*) and ground truth segmentation masks (*bottom*) of the 5 images we used in our exeperiments as gold standard

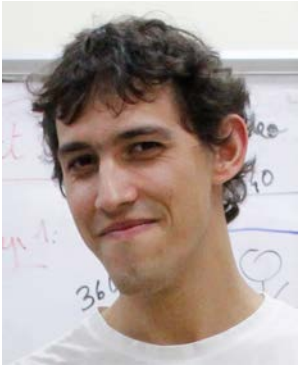
our experiments were :

- 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. Adamek T (2006) Using contour information and segmentation for object registration, modeling and retrieval. Ph.D. dissertation, Dublin City University
2. Arbelaez P, Cohen L (2008) Constrained image segmentation from hierarchical boundaries. In: CVPR
3. Arbeláez P, Pont-Tuset J, Barron JT, Marques F, Malik J (2014) Multiscale combinatorial grouping. In: CVPR
4. Batra D, Kowdle A, Parikh D, Luo J, Chen T (2010) icoseg: Interactive co-segmentation with intelligent scribble guidance. In: Proceedings of CVPR'10, pp 3169–3176
5. Bell S, Upchurch P, Snavely N, Bala K (2013) Opensurfaces: A richly annotated catalog of surface appearance. ACM TOG 32(4)
6. Boykov Y, Jolly M-P (2001) Interactive graph cuts for optimal boundary map; region segmentation of objects in n-d images. In: ICCV
7. Cabezas F, Carlier A, Salvador A, Giró-i Nieto X, Charvillat V (2015) Quality control in crowdsourced object segmentation. arXiv:1505.00145
8. Carlier A, Marques O, Charvillat V (2012) Ask'nseek: A new game for object detection and labeling. In: Computer Vision–ECCV 2012. Workshops and Demonstrations. Springer, pp 249–258
9. Carlier A, Charvillat V, Salvador A, Giro-i Nieto X, Marques O (2014) 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, pp 53–56. [Online]. Available: doi:10.1145/2660114.2660125
10. Carreira J, Sminchisescu C (2010) Constrained parametric min-cuts for automatic object segmentation. In: CVPR
11. Chen L-C, Fidler S, Yuille AL, Urtasun R (2014) Beat the mturkers: Automatic image labeling from weak 3d supervision. In: CVPR
12. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
13. Fathi A, Balcan MF, Ren X, Rehg JM (2011) Combining self training and active learning for video segmentation. In: Hoey J, McKenna S, Trucco E (eds) Proceedings of the British Machine Vision Conference (BMVC 2011), vol 29, pp 78–1
14. Giró-i Nieto X, Martos M, Mohedano E, Pont-Tuset J (2014) From global image annotation to interactive object segmentation. MTAP 70(1)
15. Jain SD, Grauman K (2013) Predicting sufficient annotation strength for interactive foreground segmentation. In: Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, pp 1313–1320
16. Lee HS, Kim J, Park SJ, Kim J (2014) Interactive segmentation as supervised classification with superpixels. In: WCVPR 2014-W. on Computer Vision and Human Computation
17. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. CoRR
18. Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum H-Y (2011) Learning to detect a salient object. PAMI 33(2)
19. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440

20. Lux M, Müller A, Guggenberger M (2012) Finding image regions with human computation and games with a purpose. In: AIIDE
21. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV
22. McGuinness K, O'Connor N (2013) Improved graph cut segmentation by learning a contrast model on the fly. In: ICIP
23. McGuinness K, O'connor NE (2010) A comparative evaluation of interactive segmentation algorithms. Pattern Recogn 43(2):434–444
24. Noma A, Graciano ABV, Cesar Jr RM, Consularo LA, Bloch I (2012) Interactive image segmentation by matching attributed relational graphs. Pattern Recogn 45(3)
25. Oleson D, Sorokin A, Laughlin GP, Hester V, Le J, Biewald L (2011) Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. Human Computation 11:11
26. Pinheiro PO, Collobert R (2015) From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1713–1721
27. Rother C, Kolmogorov V, Blake A (2004) "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3)
28. Rupprecht C, Peter L, Navab N (2015) Image segmentation in twenty questions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3314–3322
29. Russakovsky O, Bearman AL, Ferrari V, Li F-F (2015) What's the point: Semantic segmentation with point supervision. arXiv:1506.02106
30. Russell BC, Torralba A, Murphy KP, Freeman TW (2008) Labelme: A database and web-based tool for image annotation. IJCV 77(1-3)
31. Salembier P, Garrido L (2000) Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Trans Image Process 9(4)
32. Salvador A, Carlier A, Giro-i Nieto X, Marques O, Charvillat V (2013) Crowdsourced object segmentation with a game. In: ACM CrowdMM
33. Steggink J, Snoek C (2011) Adding semantics to image-region annotations with the name-it-game. Multimedia Systems:17
34. Sun Y, CHen Y, Wang W, Tang X (2014) Deep learning face representation by joint identification-verification. In: Proceedings of Neural Information Processing Systems Conference (NIPS)
35. von Ahn L, Dabbish L (2004) Labeling images with a computer game. In: ACM CHI
36. von Ahn L, Liu R, Blum M (2006) Peekaboom: a game for locating objects in images. In: ACM CHI
37. Wang J, Bhat P, Colburn RA, Agrawala M, Cohen MF (2005) Interactive video cutout. ACM Trans. Graph. 24(3)



Axel Carlier is an Assistant Professor at Université de Toulouse (INPT). He received the Eng. degree in Computer Science and Applied Mathematics from ENSEEIHT, Toulouse France and the M.Sc. in Computer Science from Universite Paul Sabatier, both in 2011. He worked for a year as a Research Assistant at the National University of Singapore. He received his PhD at Université de Toulouse (INPT) in 2014. His research interests are visual processing and human computation.

Amaia Salvador is a PhD Candidate at Universitat Politècnica de Catalunya (UPC) under the advisement of Professor Xavier Giró and Professor Ferran Marqués. She obtained her B.S. in Audiovisual Systems Engineering from UPC in 2013, after completing her thesis in interactive object segmentation at the ENSEEIHT Engineering School in Toulouse. She also holds a M.S. in Computer Vision from Universitat Autònoma de Barcelona. During the Summer of 2014, she joined the Insight Centre for Data Analytics in the Dublin City University, where she worked on her master thesis on visual instance retrieval. Her current research focuses in computer vision, multimedia retrieval and image segmentation. In the Spring of 2015, she visited the National Institute of Informatics in Tokyo.

Ferran Cabezas was born in Barcelona, Spain in 1992. He studied Audiovisual Systems Engineering in Universitat Politècnica de Catalunya (Barcelona, Spain) from which he graduated in 2015. He was awarded with an Erasmus fellowship to develop the final degree project at ENSEEIHT (Toulouse, France). His research interests include computer vision, image and video segmentation, machine learning, data mining, and human-computer interaction.

Xavier Giro-i-Nieto is an Associate Professor at the Universitat Politecnica de Catalunya (UPC). He graduated in Electrical Engineering studies at UPC in 2000, after completing his master thesis on image compression at the Vrije Universiteit in Brussels (VUB). In 2001 he worked in the digital television group of Sony Brussels, before joining the Image Processing Group at the UPC in 2002. In 2003, he started teaching courses in Electrical Engineering on the EET and ETSETB schools. He obtained his PhD on image retrieval in 2012 from the Signal Theory and Communications Department of the same university. Between 2008 and 2014, he has been a part-time visiting scholar of the Digital Video and MultiMedia laboratory at Columbia University in New York City.

Vincent Charvillat received the Eng. degree in Computer Science and Applied Mathematics from ENSEEIHT, Toulouse France and the M.Sc. in Computer Science from the National Polytechnic Institute of Toulouse, both in 1994. He received the Ph.D. degree in Computer Science from the National Polytechnic Institute of Toulouse in 1997. He joined the Computer Science and Applied Mathematics department of ENSEEIHT in 1998 as an Assistant Professor. He obtained the habilitation degree in Computer Science in 2008 and is currently a full Professor at the University of Toulouse, IRIT research lab, ENSEEIHT Eng. School. Vincent CHARVILLAT is the head of VORTEX research team at ENSEEIHT (Visual Objects: from Reality To EXpression). His main research interests are visual processing and multimedia applications.

Oge Marques is Professor of Engineering and Computer Science at Florida Atlantic University (FAU). He received his Ph.D. in Computer Engineering from FAU in 2001. His current research interests include the use of serious games and crowdsourcing to advance human and computer vision. He is a senior member of both the ACM and the IEEE.