QROWD - Because Big Data Integration is Humanly Possible

Innovation action

D3.3 – Methods for task and time management

<table>
<thead>
<tr>
<th>Author/s</th>
<th>Eddy Maddalena and Luis-Daniel Ibáñez</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due date</td>
<td>30.11.2018</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
</tbody>
</table>
## Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>3</td>
</tr>
<tr>
<td>EXECUTIVE SUMMARY</td>
<td>4</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td><strong>VCE OPTIMISATIONS</strong></td>
<td>5</td>
</tr>
<tr>
<td>Taboo strategy for reduction of redundancy and cost</td>
<td>5</td>
</tr>
<tr>
<td>Computing the walkable path to better analyse coverage</td>
<td>9</td>
</tr>
<tr>
<td>Support for multiple item types</td>
<td>12</td>
</tr>
<tr>
<td>Enriching detected items with metadata</td>
<td>14</td>
</tr>
<tr>
<td>Additional improvements</td>
<td>16</td>
</tr>
<tr>
<td>Keeping contributors outside buildings</td>
<td>16</td>
</tr>
<tr>
<td>Teleport</td>
<td>17</td>
</tr>
<tr>
<td><strong>GAMIFIED REAL-TIME CROWDSOURCING</strong></td>
<td>18</td>
</tr>
<tr>
<td><strong>CONCLUSIONS</strong></td>
<td>20</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>22</td>
</tr>
</tbody>
</table>
ABSTRACT

This deliverable presents optimisations made on the crowdsourcing task prototypes presented in D3.2. In particular, we present an analysis of experiments performed with the Virtual City Explorer (VCE) that shows the benefits given by adopting the taboo strategy to discover static city items. Then we present new techniques to improve the measurement of the coverage of the virtual exploration, in order to allow the task requester to spot out areas of the city less visited than others. After that, we discuss new features that allow configuring the VCE to operate with different city items. Then, we discuss how we aim to collect additional information about the detected items by requiring contributors to label such items immediately after the submission takes place. The type of metadata required is not fixed but depends on the type of the detected object. Finally, we discuss experiments aimed to explore the usage of additional gamified incentives in microtask crowdsourcing. In particular, we focus on the labelling of streams of data that needed to be performed under time constraints.
EXECUTIVE SUMMARY

This deliverable reports optimisations made on the prototypes presented in D3.2. In particular, we discuss improvements in collection time and reduction of duplicate locations in the Virtual City Explorer (VCE) and experiments done with Qrowdsmith.

The audiences of this document are: (1) members of the consortium to be notified about last updates of Qrowd tools, and (2) any crowdsourcing researcher or practitioner who may benefit from the experience made in Qrowd to solve problems related to mobility, similar to those we face in Qrowd.

This deliverable proposes solutions for a variety of issues that can affect crowdsourcing tools similar to those developed in Qrowd. For example, the computation of the coverage in virtual environments exploration, the customisation of tasks by introducing new features, and the real-time crowdsourcing to operate with streams of data under strict time constraints.

This deliverable is the natural extension of D3.2 in with prototypes of crowdsourcing tools were presented and will be the base of D3.5, which will discuss the deployment of crowdsourcing services discussed so far.

The main outputs of this deliverable are: (1) a report of the implementation advancements of Qrowd crowdsourcing services; and (2) evaluations and tests of the optimisations made on the crowdsourcing tools discussed and developed up to this point (3) a list of ideas to be explored in the third year of Qrowd concerning aspects such as the labelling of the detected items and the measurement of the exploration coverage;
1. INTRODUCTION

This deliverable discusses methods and strategies we adopted to improve the prototypes presented in D3.2.

The first part of the document (Section 2) discusses optimisations of the VCE (Virtual City Explorer), the crowdsourcing tool based on Google Streetview\(^1\) we developed to locate specific static items inside cities. We present the Taboo strategy, that allows to limit the number of duplicate item detections collected to reduce possible waste of money. Then, we discuss a new strategy to measure the amount of area that task contributors cover when virtually exploring a target region. Our metric is based on the concept of virtually walkable path and allows us to both monitor accurately the virtual exploration performed by contributors, and easily spot out possibly sub areas of the city that were not sufficiently explored. Also, Section 2 presents two novel optimisations we included on the VCE: (1) a native support for multiple types of city items, in a way that a task requester can easily configure; and (2), enable the possibility of adding a form in the item location task to collect from workers additional annotations about the discovered items. The main benefits of (2), is that we do not need to launch of separate annotation task involving a different set of crowdworker, reducing cost and time.

In Section 3 we discuss Qrowdsmith: the crowdsourcing platform that make use of gamification to engage contributors in performing real-time tasks related to mobility. We describe a number of experiments we carried out in the context of paid crowdsourcing in which we investigated the effect of gamification in the number of tasks that workers undertake. In particular, we observed that increasing the reward spread to workers led to more tasks completed, the effect being clearer with better workers than the ones that deliver less quality work. Also, we studied task abandonments, and in particular we focused on strategies to predict when contributors are about to leave the task, in order to avert that to happen. Thus, when we know a contributor is close to leaving the task, we offer them additional incentives to convince them to continue performing the task.

\(^1\) [https://www.google.com/streetview/](https://www.google.com/streetview/)
2. VCE OPTIMISATIONS

In D3.2 we introduced the Virtual City Explorer (VCE), a standalone tool that allows crowdsourcing contributors (either paid or volunteers) to explore 3D visual imagery of cities (the Virtual City) context to detect static POIs required to the “Completing mobility infrastructure information” use case (BC2-UC#3). In D3.2, we demonstrated that contributors could effectively locate items, however we also identified two areas of improvement for making the most out of contributors' effort: reduce the number of multiple detections of a small subset of items (we measured in some cases more than 20, when typically only 3 to 5 are needed for considering a detection as "true"); second, the detection of which areas of the region have been explored, in order to enable a degree of guidance towards less explored sections of the area.

1.1 Taboo strategy for reduction of redundancy and cost

In D3.2 we introduced the Taboo Strategy, an approach that allow to up limit the number of times a specific item can be detected (we fixed such a threshold equal to a value of three in our studies). Once an item has reached its taboo limit, new workers cannot submit it anymore. Providing the VCE with the Taboo strategy we aim at two goals: (1) a more balanced distribution of the number detections received by different city items to favour searching of new items; and, (2) a more exhaustive exploration of the contributor into the area of interest. In this section, we discuss the results of some study carried out using the VCE in the cities of Trento in Italy and Washington in the US, focusing on the comparison of results with and without the Taboo strategy.

We carried out a study to test the effect of that limitation of the redundancy of detections has on the number of bike racks detected. The experiment was performed with the VCE in Trento and Washington first without the Taboo strategy (called Basic version) and then repeated with the taboo experiment enabled. The number of times an item need to be detected to become taboo was set to three. In our setting, each runs involved 60 crowdworkers recruited on FigureEight\(^2\). Each contributor was paid 0.2USD to detected five bike racks, thus the cost of each run was of 12 USD plus 2.4USD (20 per cent) for the FigureEight fee.

Figures 1 and 2, show the number of bike racks that received at least three detections when adopting the taboo strategy and when not. We call these bike racks "Confirmed". In both charts, the gap between the red and green lines represents the additional redundancy that non-taboo strategy has with respect to when the Taboo strategy is activated. Without taboo, some bike racks are detected up to 20 times (in Trento), and up to 17 (in Washington). With the Taboo Strategy enabled (green line), the number of bike racks discovered increased from 27 to 34 in Trento, and from 37 to 40 in Washington. Note, that when adopting the taboo strategy, some items might be detected more than the required number of times if two workers submit the same item into a short interval of time; this is due to our task design that stores the submitted item only at the end of the task execution.

\(^2\) https://www.figure-eight.com/
When contributors search for items in a constrained area with the taboo strategy active, new item submissions that make previously submitted items becoming taboo are expected. These new submissions are going to progressively reduce the availability of items still detectable, and, as a direct consequence, there will be a point in which contributors will be unable to complete their task because of the lack of items still identifiable. It is difficult to predict when this is going to happen since the number of items on the area is unknown a priori. Thus, to be fair with workers that arrive at a later stage to the task, we designed an 'Escape' strategy, where, when a worker has spent a fixed amount of time and has explored a fixed number of meters without submitting any new item, he/she is get rewarded as if the task was completed, accepting his/her previous contributions. In our experiment, we set the escaping threshold to 1800m for the distance walked, and to three minutes for the time spent exploring.
Figures 3 and 4 show the evolution of the bike rack detections made as new workers arrive and complete the task:

- The blue and orange lines show the overall amount of detections performed after N workers submitted their discovered items. In both the Basic and the Taboo version of the task, workers were required to submit five bike racks. Nevertheless, in the Taboo version the Escape strategy was active, so workers could leave the task in advance, even with less than five submissions, if the escaping conditions were satisfied. For this reason, in the chart, the height of the orange line cannot overtake that of the blue one.

- The red and green lines show the number of confirmed bike racks submitted.
D3.3 – Methods for task and time management

in both Taboo and Basic tasks. In Basic executions the Escape strategy is not needed, thus, when the task begins (left side of the chart), the Basic detections are most likely to contribute to increase the number of confirmed bike racks. However, then the task execution goes on, more visible and easily discoverable items tend to be re-identified again and again by subsequent contributors. On the contrary, when the Taboo strategy is active, already known bike racks cannot be detected. This encourages additional exploration as well as the discovery of those bike rack that might be missed because of their hidden location or lack of visibility.

1.2 Computing the walkable path to better analyse coverage

One important aspect to take in consideration when looking for static items into a constrained area of a city is the coverage: has all the area been explored? To guarantee an exhaustive exploration it is necessary to verify that each street or place on the area of interest is visited by a reasonable number of explorers.

In real physical world scenarios, coverage is measured by tracking the position of workers with GPS devices, and visualising it as a heatmap. In the virtual world, a similar tracking can be implemented. In the particular case of the VCE, based on the Google Street View service. Nevertheless, to better explain how we measure coverage in our experiments conducted with VCE, some preliminary remarks might be useful. The VCE is a standalone tool based on the Google Street View service. Such service makes use of 3D imaging techniques to map the real world. The mapping phase is carried out by 3D cameras, mounted on vehicles, or carried by people, which take spherical images from different public locations from discrete points in spaces. Adjacent locations are typically few meters away from each other, and Google Street View users move between them through ‘virtual’ steps. These maps can be seen as graphs, where the nodes are the locations where photos are taken, and the edges are the virtual links that connect adjacent nodes. Given a graph of locations and virtual links computed in a limited area or a town, the sum of the lengths of all its edges gives the “Street View walkable distance” of the area.
In Figure 5 is possible to see the Street View graph for the Limited Traffic Zone (ZTL) of Trento in Italy. The area of the surface is 0.348km². The Street View graph is composed of 921 nodes (locations), connected by 961 edges (virtual links), that summed up give the Street View walkable distance of 8.8Km.

With the Street View walkable path, coverage can be computed in several ways, for example by counting the number of nodes visited by each contributor, or measuring the "walked" distance. In particular, we might be interested in checking if there are particular areas of the maps where nodes are insufficiently visited, in order to direct additional exploration in that specific area.
Figure 6: Heatmap showing the contributors exploration in Trento (ZRL) area. The black dots represent the locations that have not been visited (113 out of 921).

Figure 6 highlights how in our experiment carried out in Trento using the Taboo strategy some of the points (depicted in black in the heatmaps) were not visited at all, making impossible to confirm the presence or not of an item. With the "Cold" points identified, a new task focused on that specific area can be carried out.

The information about the number of visits received by each node of the graph gives an idea of how much that location was visited. Nevertheless, the fact that a location was exhaustively visited does not ensure that a required item placed nearby such location would have been for sure detected since it also depends by other factors, such as its visibility or the contributor attention when visiting the area.
1.3 Support for multiple item types

So far, we have discussed the VCE focusing exclusively on bike racks detection. The results of our first experiments have shown that, not only the use of VCE by crowdworkers is effective, but also that it can lead to excellent results in term of costs and time saving and in the number of elements identified, in particular when adopting the taboo strategy. Nevertheless, as discussed in D3.1, we aim to use the VCE not only to identify bike racks but also to detect other static items related to mobility.

A task requester can specify the required item type in the task setup phase. Each task type requires a template which defines the task components. The current version of the VCE offers two templates, one for bike racks and yellow parking spots detection (that as we will see further on are spatial parking areas in Italy). The VCE is designed to make easy the creation of a template for new item types. To design a task template for a new task type, it is sufficient to provide the following components:

1. **Configuration file**: A simple file containing information about the required item specified in a prefixed format.
2. **Task instructions**: An HTML file and relative images that is shown at the beginning of the task to explain to the contributor the work that needs to be done. The file also contains two sections that are shown only when: (1) taboo strategy is active, and (2) the exploring area is limited and the contributor cannot go out. The requester can customise this part, or just modify one of the already available templates.
3. **Metadata panel and logic**: An HTML page having javascript embedded containing the view and the controller of the metadata panel (it is going to be detailed in Section 1.4).
4. **Two markers**: Images to be rendered on the detected item location. Two markers are required, one for the taboo items and one for the items deleted in the ongoing task execution.

As anticipated, after the template for bike racks we focused on the one for special parking spots. We chose to test the VCE in detecting these items since being them painted on the floor and not standing up as bike racks, they could result more challenging to be discovered by crowdworkers. In fact, problems of perspective, fading of the printed lines that reduce the visibility and occlusions, potentially caused by vehicles on the parking spot, might complicate the search considerably.
Instructions

Your task is to use the tool below to find three yellow parking spots in a city.

A yellow parking is an area of the floor delimited by yellow lines, such those in these next images:

Please, notice that there are three types of disable parking spots:

For disabled people, distinguishable by an icon representing a wheelchair.

For freight load areas, distinguishable by an icon depicting a folding sack truck.

Other not-specified types of spot.

Figure 7: Task Instruction - Required item description
Figures 7 and 8 show the sections of the task instructions where the required items are described and where some good and bad example of yellow parking spots are discussed. To help crowd worker memorising the required parking slots during the task execution, we called them "yellow parking spot", because of the colour used in Italy to paint such parking spots edges. Conventional parking spots edges are painted white (free parking) or blue (paid parking).

1.4 Enriching detected items with metadata

So far we discussed the VCE as a crowdsourcing tool that generates maps of static Pols in the city. However, a municipality may want to collect additional information about the Pols, for example, in the case of bike racks, one may be interested in the type of bike rack and its capacity. First, recall that the VCE registers the snapshots taken by contributors used to compute the location. This information can reveal particularly worthy for disambiguation activities such as the “Bike Rack Use-case For Entity Disambiguation” discussed in D7.2, Section 3.

The VCE can be used to collect this metadata in two ways. These images can be used as input for a separate crowdsourcing task where a different set of contributors can annotate the images. A second alternative is to ask to the same contributors that locate the items for additional metadata. Thus, we provided the VCE with an optional functionality, that can be enabled during the task setup phase, that adds a new panel that pops out after the submission of a new item. This panel requires the contributors...
D3.3 – Methods for task and time management

to provide additional information to the item just submitted. The required information is item-type dependent, since, if for bike racks information about capability of material can be relevant, for a parking spot this kind of information would not be very meaningful. Figure 9 shows the metadata panel used in tasks for bike rack detection, and Figure 10 shows the one for tasks requiring yellow parking spot detection.

Figure 9: VCE metadata panel for bike rack detection tasks.

Figure 10: VCE metadata panel for yellow parking spot detection tasks.
D3.3 – Methods for task and time management

Figure 11 shows the task setup page. In particular, the left-bottom side of the image depicts: (1) the panel through which the task requester can specify the type of the required item, and (2) a radio-button to enable or disable the metadata request after an item submission.

1.5 Additional improvements

With the experience gained from the analysis of the data collected, we identify some areas of the VCE that could be improved. We aim at providing this improvements for the end of the project.

1.1.1 Keeping contributors outside buildings

From the analysis of the log that shows the path walkers from the contributors during their explorations, it turned out that some of them ended up inside buildings. Indoor exploration is something unwanted not only because we are not interested in finding items located inside buildings, but also because, in some conditions and with the current Street View interface, it may be complicated to leave the building once ended up inside. For these reasons we added a check in the contributor position that triggers when they attempt to access a building, preventing it and showing them a message which explain the reasons of the applied limitation. Figure 12 shows the message shown to a contributor who tries to access a building.
Another issue emerged from the analysis of the first experiments logs, and confirmed by the textual feedback collected from contributors is the difficulty in moving out from specific locations. Indeed, some semi-isolated areas are difficult to reach, and these are as much challenging to abandon since typically are in narrow streets or tunnels with a single access. When a contributor gets stuck in such locations, it can compromise the quality of their experience, thus negatively affect the overall quality of the task results too. To prevent this, we implemented a new functionality called "Teleport". A new button labelled "Back to start" that can be pressed in any moment by a contributor to be relocated in the same position they were at the beginning of the task. When teleporting shots of not yet submitted items are deleted, as they will be useless for the new location. Figure 13 shows the panel by which we require to a contributor to the confirmation of an imminent teleporting after clicking the "Back to start" button.
3. GAMIFIED REAL-TIME CROWDSOURCING

In D3.2 we presented Qrowdsmith, a standalone crowdsourcing platform that allows designing solo or cooperative tasks to be performed by paid or voluntary contributors recruited via external channels. Two important features offered by Qrowdsmith are gamification and real-time task performing. The first one is a strategy that makes use of social incentive strategies such as leaderboards, levels, and badges to improve user engagement. Traditional crowdsourcing platforms adopt monetary payment as main motivators for their contributors (in this case called crowd-workers), thus gamification is an interesting opinion to be studied along traditional monetary payment to improve contributor engagement. This idea is supported by several studies show how contributor-engagement and the quality-of-results obtained can be improved by combining economic and not economic incentives, as explained in D3.1,

Also, in some circumstances, requesters could need some work to be done quickly, for example when processing information coming as a real-time stream of data. When it is so, the time factor becomes critical since such tasks may require to be carried out in quasi-real-time. Unfortunately, common crowdsourcing platforms do not support this type of time needs, since are designed to offer tasks where the time needed for their resolution is not a priority, and latencies of hours and days are generally accepted. Real-time is also essential when direct communication among workers is required. Typically, it is so in collaborative or competitive tasks, where contributors need to interact and make decisions that can influence other contributors experiences, and even rewards.

In order to explore these areas, we carried out some experiments aimed at getting a better understanding of how monetary incentives can be joint with other incentive
techniques to improve crowd performance and enhance contributors engagement. Specifically, we focused on the deliverable-time: the time between a new task became available in a platform and the moment when it gets performed.

To do so, we designed a crowdsourcing live contest, in which participants were shown with ten twitter posts, and required to identify three types of entities into them: people, locations, and organizations. Workers need to deliver many annotations as possible in under specific time constraints. The quality of the annotations was computed by computing the overlap with the gold standard and the inter-annotator agreement (Bhowmick et al., 2008). The task was gamified, workers could see their results on a leaderboard that was also used to establish which of the workers would have been paid. Only top positions workers were paid. Additionally, to be paid crowdworkers had to (1) complete work for a fixed amount of time, and (2) reach or passed the task-threshold, consisting into the completion of a minimum amount of tasks.

We recruited contributors on FigureEight, choosing from the top 15 English speaking countries. The experiment used a dataset of 7600 aggregated tweets. We use stream parallelism to split the task is sub-tasks and assign them to different groups of worker (Quinn et al., 2011). Multiple workers accessed the platform at once to perform the task simultaneously.

To keep workers the platform longer, we implemented in our systems that detect when workers are about to abandon the task to trigger some action to convince them to keep engaged on the task. This intervention consists of making an offer, called “tipping point” to the workers we believe is about to abandon the task.

The main finding of our work can be summarised as follow: First, we asked if contests a good solution to collect data when time constraints are given. Our research shows that contests are potentially a suitable task model to be adopted when needed to collect data in presence of strict time constraints. Second, we investigated how the worker's reward and the amount of effort required affects the task quality. The study demonstrated that best workers tend to complete more tasks when increasing the reward spread. Also, an increment of the minimum effort required to workers even if supported by an increment of the task reward leads to a reduction of the number of completed tasks. Third, we wondered if it was possible to predict the task abandonment and which are the factors that influence it more. Our experiment shows that workers tend to spend more time in the task when they are offered with a higher reward and having less demanding task threshold. Behaviours of previous workers can be used to build predictive models able to prevent from workers task abandonment.
4. CONCLUSIONS

In this deliverable, we discussed the optimisations of prototypes presented in D3.2. Concerning the Virtual City Explorer, we reported the results of experiments that show the benefits derived from the adoption of the Taboo Strategy. Our optimisations achieve:

- A reduction in costs since the taboo strategy allows the location of more PoIs given a fixed amount of money.
- A more exhaustive exploration that leads to the finding of items that without the taboo strategy could have not been discovered.

We presented a new feature that allows using VCE with multiple item types. We make it possible with the use of task templates to simplify task creation and execution. We simplify the phase of template design, that can now be done in an agile way just specifying a few of elements. At the moment the VCE offer two ready-to-use templates, for bike racks and for special parking spots.

The VCE was optimised to allow requesting metadata of submitted items. Contributors can be asked to fill a form after the submission of an item. This potentially reduces the number of tasks that need to be run to get a full description of the items sought after.

The second part of the deliverable discusses crowdsourcing experiments performed to study the effect of non-monetary incentives in a near-real-time setting. The study was a crowdsourcing contest in which workers were required to provide annotations of tweets under significative time constraints. The study shows that our approach is effective. Also, we investigate how changes in reward spread and amount of work required could affect the workers' effort and quality. It turned out that by increasing the reward spread to workers, lead to a larger number of tasks completed by best workers. On the other hand, given a fixed payment, the number of tasks completed decreases when the minimum amount of effort required to complete the task is bigger. Finally, we study task amendments, focusing on how to predict them, in order to trigger countermeasures to convince them to continue the task. Our study demonstrate that predictive models built on past workers behaviours can be realised.
5. REFERENCES

**Bhowmick et al., 2008**

**Quinn et al., 2011**