Collabmap: Crowdsourcing Maps for Emergency Planning

Sarvapali D. Ramchurn,[1] Trung Dong Huynh,[1] Matteo Venanzi,[1] Bing Shi[2]
[1]Electronics and Computer Science, University of Southampton, United Kingdom
{sdr,tdh,mv1g10}@ecs.soton.ac.uk
[2]School of Computer Science and Technology, Wuhan University of Technology
Wuhan, China
bingshi@whut.edu.cn

ABSTRACT

In this paper, we present a software tool to help emergency planners at Hampshire County Council in the UK to create maps for high-fidelity crowd simulations that require evacuation routes from buildings to roads. The main feature of the system is a crowdsourcing mechanism that breaks down the problem of creating evacuation routes into micro-tasks that a contributor to the platform can execute in less than a minute. As part of the mechanism we developed a concensus-based trust mechanism that filters out incorrect contributions and ensures that the individual tasks are complete and correct. To drive people to contribute to the platform, we experimented with different incentive mechanisms and applied these over different time scales, the aim being to evaluate what incentives work with different types of crowds, including anonymous contributors from Amazon Mechanical Turk. The results of the 'in the wild' deployment of the system show that the system is effective at engaging contributors to perform tasks correctly and that users respond to incentives in different ways. More specifically, we show that purely social motives are not good enough to attract a large number of contributors and that contributors are averse to the uncertainty in winning rewards. When taken altogether, our results suggest that a combination of incentives may be the best approach to harnessing the maximum number of resources to get socially valuable tasks (such for planning applications) performed on a large scale.

Categories and Subject Descriptors

C.5 [**World Wide Web**]: Crowdsourcing

1. INTRODUCTION

The creation of high fidelity scenarios for disaster simulation is a major challenge for a number of reasons. First, in the UK, the maps supplied by existing map providers (e.g., Ordnance Survey, TeleAtlas) tend to provide only road or building shapes and do not accurately model open spaces which people use to evacuate buildings, homes, or industrial facilities (e.g. the space around a stadium or a commercial centre both constitute evacuation routes of different shapes and sizes). Secondly, even if some of the data about evacuation routes is available, the real-world connection points between these spaces and roads and buildings is usually not well defined unless data from buildings' owners can be obtained (e.g. building entrances, borders, and fences). Finally, in order to augment current maps with accurate spatial data, it would require either a good set of training data (which is not available to our knowledge) for a computer vision algorithm to define evacuation routes using pictures (working on aerial maps) or a significant amount of manpower to directly survey a vast area.

Against this background, we developed a novel model of geospatial data creation, called CollabMap[1], that relies on *human computation*. CollabMap is a crowdsourcing tool to get users to perform micro-tasks that involve augmenting existing maps (e.g. Google Maps or Ordnance Survey) by *drawing evacuation routes*, using satellite imagery from Google Maps and panoramic views from Google StreetView. In a similar vein to [12, 4], we use human computation to complete tasks that are hard for a computer vision algorithm to perform or to generate training data that could be used by a computer vision algorithm to automatically define evacuation routes. Compared to other community-driven platforms such as OpenStreetMap and Google's MapMaker, Collabmap allows inexperienced and anonymous users to perform tasks without them needing the expertise to integrate the data into the system (as in OpenStreetMap) and does not rely on having experts verifying the tasks (as in MapMaker) in order to generate meaningful results.

To ensure that individual contributions are correct and complete, we build upon the Find-Fix-Verify (FFV) pattern [1] to develop a novel adaptive workflow that includes concensus-based trust metrics and allows the creation of new tasks where no ground-truth is known. Our trust metrics allow users to rate and correct each other's contributions while our workflow is adaptive in the sense that it allows the system designer to improve the performance of the crowd according to both the number and types of contributions into the system. As we show in our results, this approach was effective in preventing workers from getting bored and taking full advantage of users' motivation to contribute.

Given our implementation of the platform, we deployed our

[1] `www.collabmap.org`

system to help map the area around the Fawley Oil refinery next to the city of Southampton in the UK over three months. The area covered over 5,000 buildings (mainly residential) with a population of about 10,000. We experimented with different incentive mechanisms to incentivise both the local community around the refinery and other agencies, and (our) University staff and students to contribute to the platform. Thus, apart from using the moral or intrinsic incentive to participate in the exercise, we attempted to engage the crowd using different monetary incentives in turn including lottery-based rewards and competition-based rewards. As a benchmark, we also ran the system using Amazon's Mechanical Turk (AMT) to hire users to participate in the platform.

Our results show that the local community was not particularly responsive to monetary rewards and that competition-based rewards play a significant factor in attracting users, though not large numbers of them. Compared to results from the AMT deployment, the results from local and University users were significantly skewed. When taken together, these results allow us to claim that crowdsourcing deployments require a number of different incentive schemes to maximise the completion rate and quality of tasks, particularly when local knowledge is essential to guarantee a high level of quality.

The remainder of the paper is structured as follows. Section 2 surveys related work on crowdsourcing and incentive schemes in crowdsourcing. Section 3 describes the design principles behind Collabmap, while Section 4 elaborates on the workflow and interface design that build upon such principles. Then, Section 5 describes our deployments of Collabmap under different incentive schemes and compares the results of such deployments with that on AMT (Section 6). Finally, Section 7 concludes and discusses the key implications of our work and point to future work.

2. LITERATURE REVIEW

In what follows, we present related work and discuss how approaches within these fields relate to what we achieve in Collabmap. In particular, we focus on crowdsourced mapping platforms, workflows for crowdsourcing, trust and verification mechanisms in crowdsourcing, and most importantly, on the use of incentives of different types in crowdsourcing.

2.1 Crowdsourced Mapping

The most well known example of crowdsourcing of mapping tasks for disaster management is that of OpenStreetMap[2] where volunteers dedicate their free time to measure roads, footpaths, and in some cases, buildings, in order to build an accurate map. Other examples such as Crowdmap[3], Google's MapMaker[4], or Geo-Wiki,[5] allow users to view areas on an existing map and annotate them with extra information or identify mistakes in the maps.

Collabmap is similar to MapMaker and Geo-Wiki in that it takes a *top-down* approach to mapping. By this we mean that the contributors in Collabmap are given existing maps or aerial pictures (or could be given satellite imagery like Geo-Wiki) in order to create a map. OpenStreetMap (OSM), on the other hand, takes a *bottom-up* approach to crowdsourcing in that the contributors have to physically visit areas needing mapping and collect GPX data points. Users can also edit maps using an browser-based editor. There are trade-offs in both approaches. Top-down mapping (including the edit-mode of OSM) is obviously more accessible to a larger pool of contributors sitting at their computers looking at maps and therefore *cheaper* to contributors but does require the task requester to have access to high quality aerial maps or satellite imagery (in the case of Collabmap, we utilised freely available images from GoogleMaps[6]). On the other hand, bottom-up mapping is costly for the contributors not only because it requires them to physically move to certain places, but also because it requires them to perform accurate measurements and learn to upload such data in the right format to the platform. This can be yet another hurdle for novice contributors.

In one special instance similar to Collabmap (OSM does not routinely use tracing from satellite imagery), OSM maps were traced using satellite imagery from GeoEye in order to permit the construction of one of the most accurate maps of Port-au-Prince after the Haiti earthquake.[7] In that case, it is not clear how many mappers were involved nor whether there was any verification process for all the routes and buildings drawn [15]. As Zook et al. point out, however, for the purpose of disaster mapping, it was not so important to have a highly accurate map and a good enough map normally would do. While the contributions to the Haiti earthquake mapping had a clear and urgent outcome (helping to save lives) and therefore took only about a week, it is not clear how such contributions could be incentivised from a larger crowd in daily emergency planning applications like ours. Moreover, a key challenge is to engage participants with other motives than the interest in mapping (e.g., such as a competition or a reward to map an area as in Collabmap) [4, 11]. This is a key distinction of our approach.

2.2 Workflows for Mapping

The crowdsourced mapping platforms mentioned in the previous section all require users to perform a set of key steps in a certain order to contribute roads, buildings, or any other measurements. These steps are very simple for Google's MapMaker and Geo-Wiki. In these workflow, the user locates an area of the map and identifies a feature (e.g., region, location, building) and annotates it using mouse-clicks and some text. For OpenStreetMap, specific measurements taken on the ground using special devices need to be uploaded to the system. In both cases, uploads are checked by expert volunteers (who have been pre-selected either through a long verification process or were trusted by the system designers). In the case of Collabmap, a similar approach is taken in that, instead of having experts check a road or building, Collabmap breaks down the mapping task into small, easy-to-perform, tasks and gets each of them checked by more than one other user (who may not be an expert).

[2] www.openstreetmap.org.

[3] http://crowdmap.com.

[4] www.google.com/mapmaker.

[5] www.geo-wiki.org.

[6] GoogleMap aerial imagery is not up-to-date but reasonably accurate. If an up-to-date map were required, this would come at a high cost and therefore render the top-down approach we take, very expensive.

[7] wiki.openstreetmap.org/wiki/WikiProject_Haiti/Imagery_and_data_sources.

Hence, Collabmap aims to achieve correctness via redundancy (i.e., multiple users vote on the correctness of tasks) similar to other platforms such as AMT. By so doing, Collabmap can effectively track the correctness of tasks performed by each individual user, and hence her trustworthiness. We elaborate on this point in the next section.

2.3 Trust and Verification Mechanism

Trust is a key issue when it comes to giving tasks to anonymous contributors whose incentives may be just to make the maximum profit by doing as many tasks as possible. While in some platforms [1, 9] automatic verification processes are put in place to make sure that workers do not get paid to do tasks poorly, in other platforms experts are used to correct and train newcomers (e.g., in Google MapMaker and OSM). As more information is acquired as contributors perform tasks, reputation metrics can be used to decide whether to let them do more tasks or to ascribe a level of credibility to their contributions (e.g., as in Yahoo Answers or AMT where workers can be blocked).

A key issue with mapping tasks from aerial imagery and street view is that many of the images can be outdated (as users reported in our system). This means that the ground truth cannot be obtained by simply looking at the pictures and figuring if the task was correctly performed. Moreover, different users may have different views as to what an evacuation route consists of. Typically, in such cases where it is either too expensive (to hire experts or to send a camera to map the area) to get the ground truth, one would rely on consensus metrics [9]. In Collabmap, we take an automatic verification approach as experts are hard to find for the tasks at hand. In particular, as we show later we take a majority voting approach aims to account for such uncertainties.

2.4 Incentives

A number of studies have been carried out to test how incentives affect the performance of the mapping crowd both in terms of quality and quantity. In particular, we note the work of [4], who describes the different types of mappers typically involved in crowdsourcing platforms. These are: (i) Map lovers — a small group who produce trustable and very valuable data, (ii) casual mappers — hikers, bikers, mountaineers for example who spend little effort mapping, (iii) experts — users in organisations that require mapping such as mountain rescue, fire brigades etc., (iv) media mappers — large groups motivated by competitions through media campaigns where the contributions are limited in time and extent and a big initial effort for the campaign is needed, (v) passive mappers — users with mobile phones or GPS positioning that may be unaware they are providing data to a system, (vi) open mappers — users that spend a significant amount of time and effort to create open datasets such as OSM, and finally (vii) Mechanical Turks — who perform tasks against monetary payments. For example, while the main contributors to OSM or MapMaker would particularly be contributors of type (i) and (vi), while we would typically fit Collabmap into categories (iv) and (vii).

Now, the issue of incentives to perform tasks correctly and completely is not typically addressed by the crowdsourcing platforms above as these platforms usually rely on experts to do the corrections. Instead, such issues are common in other crowdsourcing domains to AMT or in large-scale deployments like the Darpa Red Balloon Challenge [10] where users are more interested in the financial reward they offer, however minute it might seem to be. Community-sourcing is yet another recent approach that successfully engaged a local specialised community in order to get expert tasks done for small monetary rewards [3]. Gamification and monetary rewards have also been shown to be quite successful in the past in getting the crowd to participate in short lived and focused activities [2]. ESP games, for example, were very successful not only because of the playfulness of the games but also because players inherent competitiveness drove them to do tasks. In contrast to such successful incentive mechanisms, the MyHeartMap challenge[8], with the potential to generate really strong intrinsic motivations, such monetary rewards were less successful at engaging the crowd. Indeed, studies by [2, 7] point to the fact that incentives of different types, namely intrinsic (personal motivation as for map lovers), extrinsic (monetary), and social (reputation based), can have significant impacts on the performance of a crowd. In particular, under controlled settings, Frindley et al. [2] report that that social, intrinsic, and extrinsic incentives are all effectual, but *extrinsic incentives are the strongest in motivating individuals towards prosocial crowdsourcing behavior.*

Moreover, in some deployments, it has been shown that feedback provided by the platform and other contributors [14, 6] gives better motivation and guidance to contributors to do tasks and to do them correctly as well. In a similar vein, in Collabmap, we experimented with all the three types of rewards and show that they are each beneficial in bringing in different types of contributors. Stretching the deployment of Collabmap over time, along with different types of feedback given to the crowd, allowed us to identify the most effective methods to get people to do work for both monetary and non-monetary rewards. Our results also corroborate the expectations of Mason and Watts (2009) in that the quality of work that gets done by contributors with intrinsic motivation is higher than those with purely extrinsic or social motivations [7]. Indeed, our results generalise theirs to some extent beyond the context of AMT.

3. DESIGN PRINCIPLES

In this section we explain the key principles upon which we designed Collabmap. In more detail, we can characterise the work in Collabmap in terms of the architectural (i.e., workflow and task design), reward engineering (incentives and engagement) and the quality assurance (trust and verification mechanism) elements it consists of. Our design assumes that it is possible for both local inhabitants and remote users on the web to work together to create an accurate map even though remote users may not be familiar with the local environment being mapped — therefore relying on the local inhabitants to correct their mistakes if any.[9] This was motivated by the successful deployment of OpenStreetMap in the Haiti Earthquake using a similar process and from the work done on Google's MapMaker by anonymous contributors.

3.1 Workflow and Task Design

[8] `www.med.upenn.edu/myheartmap/`.

[9] Obviously having more local inhabitants to perform such a task using GPS loggers would be the best option but this is not always feasible.

The workflow adopted in Collabmap builds upon previous divide-and-conquer approaches that have underpinned many crowdsourcing and citizen science deployments over the last few years such as [13] or [1]. Thus our workflow divides the task of drawing evacuation routes from a building into a number of micro-tasks, each requiring seconds to complete. This includes verifying whether the buildings or routes drawn are correct and complete. This approach, while inspired mainly by FFV, significantly differs in that it incorporates notions of trustworthiness, whereby redundant verification or route checking tasks are used to make sure that every 'find', 'fix', and 'verify' task is viewed, corrected, or completed by every worker (see Section 4). Moreover, we present a novel adaptive workflow that can be quickly reformatted (e.g., require fewer verifications, or impose more or fewer restrictions on access to tasks) in order to allow the workflow to adapt to the performance of the crowd. In more detail, as we show in evaluation section, for example, restricting users to perform certain tasks in an attempt to prevent them from validating their own work, may be unwieldy; and changing the workflow to let them do more work but get others to verify their work may be a more productive alternative. Our approach alleviates the issue of boredom in the case contributors get to see too many of the same type of tasks.

In designing tasks, we followed some of the guidelines from [8] by providing clear and accurate instructions, broken down into a number of concrete steps to avoid misinterpretations and boredom. As part of this, feedback was considered a key element of the design. Thus, while performing individual tasks, feedback needs to be given to participants on the validity of their actions and other participants' inputs into the system. More importantly, feedback needs to be given about how their contribution is helping to improve the aggregate performance of the crowd. This allows them to see the value of their work within the whole process.

3.2 Community Incentives and Engagement

Collabmap was designed with the premise that the local community would be incentivised to help their local emergency planners to be better prepared for disasters. Such intrinsic motivations were taken as being the key driver for contributions to the platform (our expectations were later revealed to be less positive). However, we also anticipated that participation from a local community (with no particular interest in Web technologies) would not be significant. To address this, we explored other incentive schemes that could be used to incentivise crowds, including the use of gamification and use of micro-payments per task.

Building upon this reasoning, in Collabmap, we applied different rewards (in order to generate intrinsic, extrinsic, and social motivations) over different time-scales to tease out their effectiveness at engaging different communities. This differs from other approaches that have specifically targeted crowds of a certain type with the aim of identifying the relationship between incentives and task performance. Rather, we are more interested in understanding how incentives can be shaped to access different communities and what this means for the quantity and quality of their contributions.

4. CROWDSOURCING WORKFLOW

CollabMap crowdsources the task of identifying building evacuation routes to a large number of contributors, by offering them freely available data, such as satellite imagery (e.g. Google Maps), and panoramic views (e.g. Google Streetview) to carry out this task. By so doing, even users not familiar with an area can potentially contribute evacuation routes (though local inhabitants are expected to provide more accurate data and tasks could be targeted at them if their locations are known). The scope of a task is to identify a single building, and each task follows a workflow based on a divide-and-conquer approach with verification processes in-built [1].

4.1 Tasks

We divide the task of identifying evacuation routes for a single building into smaller activities, called *micro-tasks*, carried out by different contributors. We have designed five types of micro-task:

A. Building Identification The outline of a building is drawn by clicking around the shape of a building on the map. It serves as the basis for the other micro-tasks.

B. Building Verification The building outline is assessed, with a vote of either valid (+1) or invalid (−1).

C. Route Identification An evacuation route is drawn by clicking as many times as is needed along an observed path, to connect an exit of the building to a nearby road (which is connected to the building through a footpath or a walkable/driveable space[10]).

D. Route Verification The evacuation routes are assessed for invalid routes. Those are marked as invalid receive a −1 vote, while the rest get a +1 vote.

E. Completion Verification The set of evacuation routes is assessed for exhaustiveness, with a vote of either complete (+1) or incomplete (−1).

The CollabMap workflow (Figure 1) has two main phases:

Building phase The outline of a building that has no evacuation route needs to be drawn (**A**). The outline is then checked by other contributors, who vote up or vote down the building outline without seeing others' votes (**B**). If the total score of the building, defined as the sum of all the votes, reaches +3 then the Building phase ends and the Evacuation route phase begins. If the score reaches −2, the building outline is rejected and marked as invalid.

Evacuation route phase This is the main activity carried out by CollabMap contributors. The first is permitted only to draw a route (**C**). Subsequent contributors are asked to verify routes (**D**) and are asked whether the set of routes is complete (**E**); if it is not, they are invited to draw new routes (**C**). New routes may be drawn if there are multiple walkways or open spaces between the building and the road or where there are multiple entrances through fences or walls. All likely entry points from from a building to a road can therefore be considered (and may lead to some ambiguities as we noted in our deployments).

[10] Users are not asked to draw a route over any part of the road network but only over the space between the building and the road.

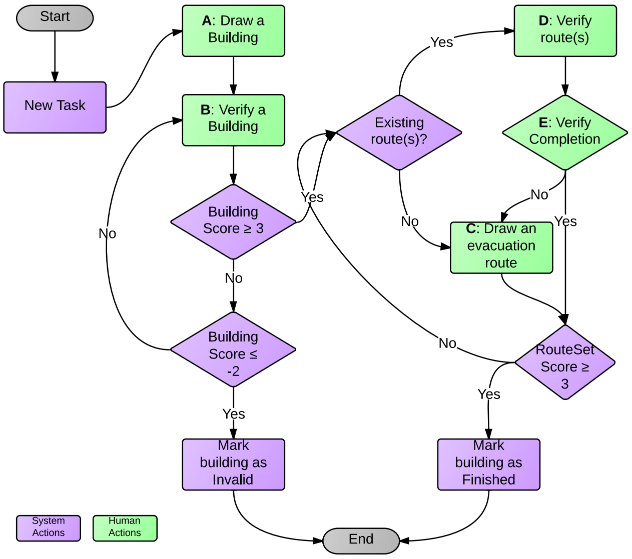


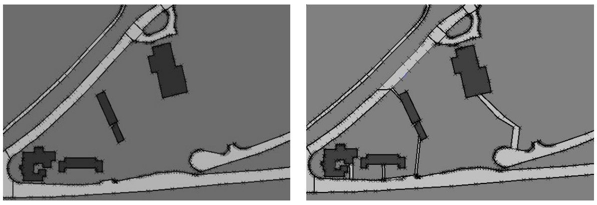
Figure 1: The CollabMap workflow for identifying evacuation routes of a building.

In both phases, in order to avoid biases (or obvious exploitations of the workflow), a contributor is not allowed to verify his or her own work.

By only progressing tasks that obtain a small majority vote, we aim to reduce the uncertainty inherent in these tasks, particularly because in many cases the definition of an evacuation route or even a building may be different for different users (sometimes we found people understood a terraced building to be a composition of several terraced houses rather than a single building). However, our intuition was that, by getting people to verify similar tasks done by other participants (and seeing what was being drawn/accepted by others), they would converge to an agreement about what constituted an evacuation route and what did not. This was indeed found in our deployment.

5. DEPLOYMENTS AND EVALUATION

To test whether Collabmap could achieve its objective of creating a high resolution map of an area, we deployed Collabmap in two different settings: as an open web application and as a separate AMT session. In the first case, we started the 3-month trial in December 2011 (we let the system run thereafter without any promise of rewards as well) and it generated more than 38,000 micro-tasks from the crowd. The mapping exercise in high definition routes as shown in Figure 2, and, as can be seen from this example, remote users were successful at performing mapping tasks even though sometimes they were corrected by the few local residents who participated and knew the area well. The AMT deployment was run in only 6 hours and generated more than 8,500 micro-tasks. In what follows, we elaborate on the web-app deployment, which we will term, the 'Local' setting for the rest of this section and describe how the incentives applied at various stages of the deployment allowed us to engage with different communities of users. We then compare these results with the AMT deployment and discuss the advantages and disadvantages of the two approaches.



(a) Original Map. (b) Map with routes.

Figure 2: Example of the results of Collabmap where remote users along with local residents drew routes for a given area. Note that a simple 'nearest road to building' approach to automatically drawing routes would not be particularly effective and shows how this task is non-trivial for an algorithm to do.

5.1 Local Deployment

In this setting, we advertised to a number of communities at different times during the deployment. Through our link to Hampshire County Council, we were able to access a number of organisations across the region around the Fawley refinery. This included local government agencies, companies with facilities located at the refinery, and local libraries and community centres. This was mainly done via email and phone calls directly to such agencies. We also targeted local newspapers and blogs which were forthcoming in helping us advertise the deployment and invited users to contribute. The total cost in advertising reached about £300. Following lack of involvement from the local community (details follow in the Results section), we also targeted local students and staff at our University through mailing lists, social media, and news articles on the University's web pages (from the 6th of February onwards). Overall, through these initiatives, we received over 2,200 hits on the main Collabmap page (i.e., excluding task execution pages) over the duration of the deployment, with 793 unique visitors.

5.1.1 Data Collection

To collect data, participants were requested to fill in a registration form (following usual ethics approval) which allowed us to collect participants' usernames (rather than real names to keep them anonymised), email, age, and location (both town and country). Thus, the deployment resulted in 118 participants from 8 different countries, where 90% were resident in the UK. As expected, a large number of these participants were from Southampton (65) but, to our surprise, only 6 participants were from the area around Fawley (not counted as part of Southampton). Moreover, out these 6 participants only 2 made it into the top 35 contributors who contributed more than 100 tasks to the platform, the rest being University students and staff. While performing the tasks, participants were also allowed to feedback on individual tasks and we collected over 120 comments from the crowd (most of them from the top contributors).

5.1.2 Incentive Schemes

The low turn-out from the local community was a key driver to alter the incentive scheme during the deployment at various stages, in order to maximise participation in the system. In turn, we tried the following incentive schemes over different timescales:

1. **Lottery-based reward** — December 2011 to February 13th, 2012 — we set up a lottery mechanism by which participants to the platform would be allocated tickets based on the number of tasks performed. For every 10 micro-tasks completed, a participant would receive 1 ticket to be drawn at random in order to win one of two prizes of £100 and £200. Furthermore, we ensured only *valid* tasks counted; that is, only tasks that had been voted up by other participants. This reward mechanism was accompanied by a leader board that showed the top 10 contributors to the platform *along with the total number of tasks* performed by each participant. If a given participant was out of the top 10, she was shown her rank on the leader board along with her contributions.

2. **Lottery+Competition-based reward** — February 13th to February 24th, 2012 — we advised participants that the reward scheme was changing on the 13th of February given the the low contributions received that far into the deployment. Thus, we increased the lottery-based reward to the following scheme to get more participants into the system while keeping the same ticket allocation system. In particulars, 8 lottery-based prizes were to be given out: one £300, one £200, and six £50. We also included a top prize of £100 in the system (which proved to be a key driver). Moreover, we changed the leader board to reflect the competitive nature of the interaction with the system by removing the number of total tasks completed from the leader board. This was done to prevent participants from losing motivation if they saw the number of tasks completed by the top contributors, meaning thinking they would never be able to win the top prize. However, *they were able to see how many tasks were left for them to beat the competitor just above them.* Skewing the presentation of contributions in this way is common practice in gaming systems and is meant to keep a participant engaged in a two-player game with clearer rewards (i.e., beating the one above) than in multi-player settings where one may be too far behind to hope to win anything (the player's rank only gives a clue but not complete information).

As can be seen from the above setup, we increased the rewards from a total of £300 to £900. However, the probability of winning with increased rewards was also smaller as the players also expected an increase in the number of competitors. The guaranteed reward of £100 for the top contributor was a key driver to get large numbers of tasks done as we show in the next section.

5.1.3 Results

Here we analyse the results of the local deployment and discuss the features of the work produced during the Local deployment. In turn, we present results with regards to participation, work done, and quality of work done.

Incentives to Participate
Our incentive schemes clearly had different impacts on different communities. As can be seen from the results provided on Figure 3 the advertising drive targetted the local community did not work well (but for one contributor). Over a period of a month, only about 20 participants signed up, some of whom were researchers interested in the project, and only 3 from the local community. As confirmed by Figure 4, the number of tasks performed by this pool of participants was negligible.

When further advertising events hit the University, the number of participants can be seen to nearly triple around the 6th of February. At this point, clearly the students were interested in the lottery-based reward and joined in masses. However, the number of tasks performed was still relatively low. This clearly points to the weakness in the incentive scheme in not being high enough or competitive enough for the students to engage with the system. On February 13th, 2012, with the announcement of the Lottery+Competition incentive scheme, not only did the number of participants rise further (by 300%, see Figure 3) but the number of tasks significantly jumped as can be seen on Figure 4.

Turning to the details of Figure 4, it can ben noted that different participants performed different numbers of tasks at various times. Clearly, two contributors performed many of the tasks (the top two contributors contributed nearly 9000 tasks each) and there is a clear distinction between them. Indeed, one contributor, who we will call participant T, who lives in the area around Fawley, joined the system very early on and did large numbers of tasks on a daily basis from January 25th, 2012 onwards; while the other top participant, who we will call S, a student who lives in Southampton, joined the system as soon as the platform was advertised to the University (i.e., on February 6th), did moderate number of tasks for two days and then stopped.

While T did varying amounts of work on a daily basis, S clearly contributed significant amounts as soon as the new Lottery+Competition scheme was put in place. Several others joined (at least 4 other participants, mainly students) at the same time (they had not joined on February 6th) but were obviously not as successful at S (who did nearly 4500 tasks in one day, which, at a rate of 20 seconds per task would take about 25 hours). These results tell us that the students were clearly more motivated by the competition than the lottery or the social benefits of the task, particularly when we study the lack of participation between the 6th and the 13th of February.

Behaviour of the Top Participants
Following interviews with S and T, we found that, as expected, S and T had different motives. In particular, it was found that S, keen on winning the competition, *crowdsourced* his tasks to friends that were either based in Southampton or abroad (in a different time zone), which would explain the number of tasks completed in a day. The strategy employed involved opening multiple browser windows and performing as many tasks in parallel as possible (e.g., doing a task on one page while another page loads another task). Sharing of login information was not prevented by our rules and, clearly, this participant exploited this. However, this backfired at some point.

When asking friends to perform tasks, S did not specify to them how well the tasks must be performed and, as a result, many tasks ended up being performed really poorly (buildings being drawn as triangles — as allowed by our system as a minimum requirement for a building, but voted down by

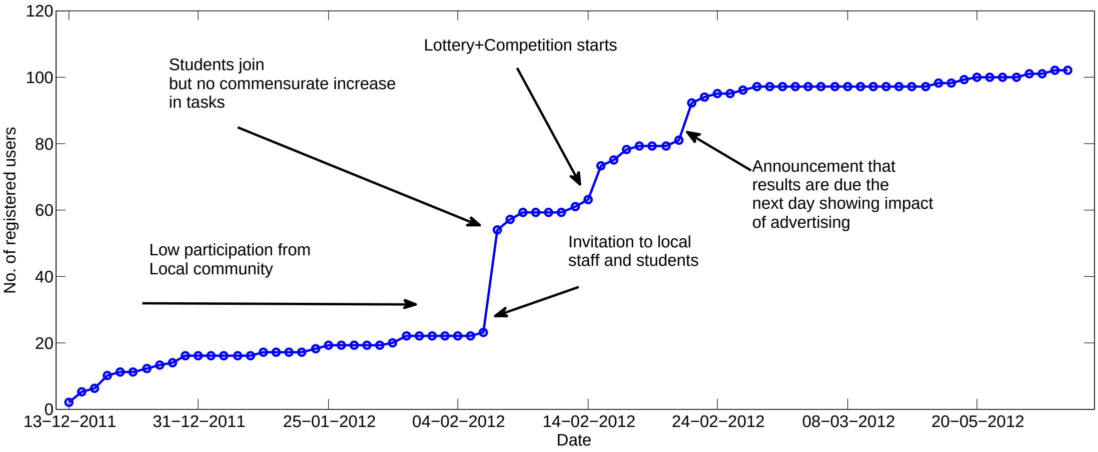


Figure 3: Number of registered participants over the duration of the deployment.

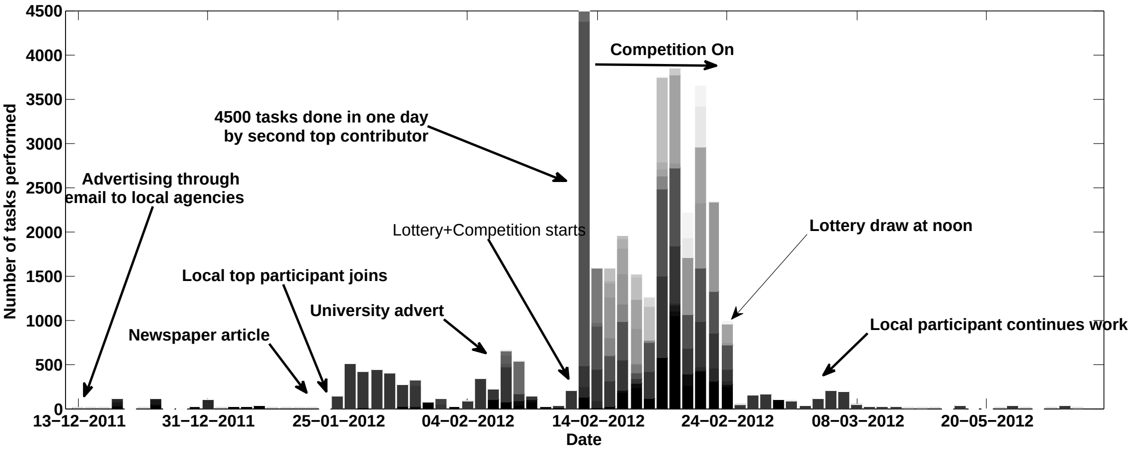


Figure 4: A stacked bar chart of the number of tasks performed over time (per day) by all participants. The different shades of grey represent individual bar charts of tasks performed by different participants.

other contributors when incorrect). As the tasks performed by S degraded, it became apparent that such a strategy was not working and S had to stop the 'sub-crowdsourcing' of tasks. Such behaviours are reminiscent of AMT where a number of 'companies' hire teams of 'turkers' in order to perform tasks under a single username in order to maximise rewards.

T, in contrast, was only partially motivated by the monetary gains and clearly expressed an interest in the project to build the simulation map. This is evidenced by T's performance of tasks going beyond the announcement of the results of the lottery/competition winners as shown on Figure 4. Also, as can be seen beyond the announcement of the Lottery+Competition (February 13th), T did not adopt alternative means to increase task performance and was also constrained by work commitments (students clearly had an advantage in this case).

The other students who contributed a large chunk of the tasks were not as strategically aggressive nor as altruistic as S and T, which points the fact that these are extremal behaviours rather than the average case. However, these results show that, in the design of crowdsourcing platforms, it is important to take into account both types of extremes in building the workflow and incentives in order to maximise task completion. In our deployment, we reached out to a small population and, if such a deployment were to be carried out on a large scale, we would expect such extreme behaviours to be more widespread.[11]

Task Types

The nature of tasks and the workflow within which they fit in Collabmap imply that contributors can only perform some

[11] A similar behaviour is observed for example on Google's MapMaker, for example, where they report: "One of our most prolific US users is a woman with over 100,000 edits. The interesting thing is that the vast majority of her edits are in Senegal, a place she's never been before. How and why does she do it? Well, she has an academic interest in the topic of Senegal, so she looks at the satellite images and creates roads where they show up in the images, then lets local users fill in the data, which they often do quickly." — at `http://thenextweb.com/google/2011/05/28/the-story-behind-googles-map-maker-editing-app/`

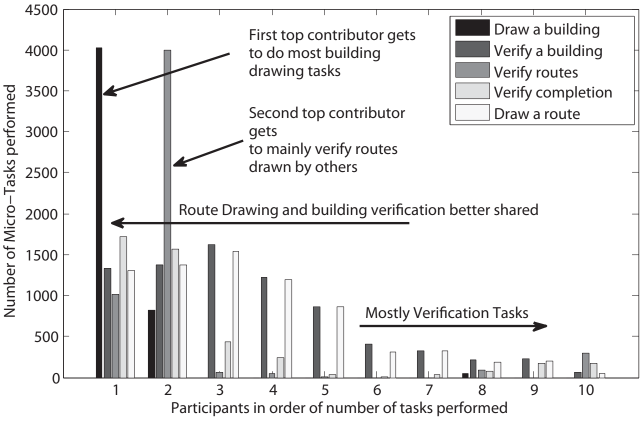


Figure 5: Distribution of task types across the top 10 participants for the Local deployment.

tasks if other contributors have acted in the system before. The fact that the initial task involves drawing buildings, it is clear that the arrival of tasks in the system will skew the distribution of tasks across the participants. Figure 5 shows that the top contributors were the ones doing the most building drawing tasks, while the other contributors were playing catch-up by verifying and correcting tasks done by their leaders. This 'race' condition generated reactions from the crowd which were characterised as follows:

1. Lack of tasks in the system and boredom — reported by participant T initially, as he was given too many buildings to draw and found the tasks getting increasingly boring. This issue was exacerbated by the fact that participants could not initially draw routes for buildings they had drawn but this constraint was relaxed in our workflow in order to allow more tasks to be completed by motivated contributors.

2. Unfairness in task distribution — those who came into the system later in the deployment were given verification tasks for many buildings drawn by earlier contributors. This is because the workflow prioritised such tasks in order for contributors to move to route drawing tasks (which was the main aim of the exercise). As verification tasks are easier (take seconds as opposed to up to a minute for a drawing task), the top contributors complained of getting too many drawing tasks to do.

The above issues were clearly a result of a unique mix of events that are peculiar to some crowdsourcing platforms where participant arrival rate is not constant and this hampers the efforts of the most eager contributors. In particular, as in our case the top contributors performed tasks reasonably well, with hindsight, it was unnecessary to impose consensus-based metrics to restrict their actions in the system as per our workflow. However, our workflow allowed us to detect anomalies in the population (such as the poor sub-crowdsourced tasks discussed above) and, hence, weed out poor performance. These constraints are therefore a necessary evil in a system with keen and trustworthy contributors. Our interaction with the crowd showed that it is important to communicate with them the reason for the behaviour of the system in order to prevent them from leaving (we did this through exchange of emails based on received in-task feedbacks).

6. AMT DEPLOYMENT

As introduced earlier, Collabmap implements the feature of crowdsourcing its micro-tasks also through AMT. We next discuss the system setup and the results of the deployment as we varied the incentives in the system.

6.1 System Setup

An AMT extension was developed to allow Collabmap to combine inputs from AMT with those from users coming from local crowds. By so doing, we aimed to ensure sufficient participants engaged with the system to complete the mapping of the area. In what follows, however, we focus only on the AMT deployment. In more detail, the extension allowed us to post a certain number of tasks on the AMT crowdsourcing market, and to specify the reward to be paid for each task. Then, for each task, an AMT human intelligent task (HIT) is created and is immediately available to the community of the AMT workers. It should be noted that no worker requirement or qualification was specified for a HIT (typically used to make the HIT available only to a subset of targeted workers) since we aimed to reach a high rate of HIT acceptance amongst the largest pool of available workers. Indeed, setting these requirements is still an area of research as the trade-off between rate of task completion and quality of work is not well understood. Moreover, we rejected work that was obviously wrong (i.e., no buildings with fewer than 3 corners and roads not connecting to a building) and accepted the rest of the submitted work.

To allow AMT workers to access our server, HITs are created using the "external question" AMT template which displays the Collabmap task execution page in a frame in the worker's web browser. Furthermore, in order to provide an interested worker with the basics of a Collabmap task, the HIT can be first previewed with an example of the drawing task and the verification task. Once the worker accepts the HIT, a new Collabmap user is automatically created in our database, or an existing user is logged in for a returning worker, and one of the available micro-tasks is assigned to such a user. This allows the system to keep track of AMT workers and prevent them from verifying their own work.

6.2 Results

In our deployment, we analysed the response of the workers and the frequency of task executions with an incremental reward strategy as follows. In the first hour, 100 Collabmap HITs were posted, for the reward of $0.01 for each HIT, and only 8 tasks were accepted, i.e. 0.13 tasks/min. In the second hour, a batch of 100 HITs was posted with the increased reward of $0.02 per HIT. These were completed in less than 40 minutes, i.e . 2.5 tasks/min. Then, we set the price to $0.03 for a new batch of 1500 HITs, and these were also all completed within 90 minutes, thus, the acceptance rate grew to 16.3 tasks/min. Finally, as we noticed that an increasing number of eager workers were keen in taking Collabmap, then we published a larger set of more than 7000 HITs, paying $0.02 each. Thesewere completed in less than three hours at the average rate of 44 tasks/min.

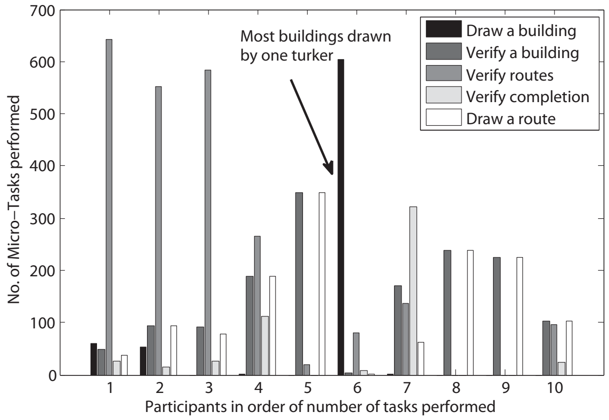


Figure 6: Distribution of task types across the top 10 participants for AMT.

Summing up, in the 4 trials, we were able to recruit 150 AMT workers in less than six hours, paying a total cost of \$187. The workers completed 8,979 micro-tasks. However, the AMT deployment generated data of significantly lesser quality compared to the those from the local deployment.

6.3 Local versus AMT deployments

The task completion rates in the Local and AMT deployments were significantly different as noted from the previous results. During both the deployments, we recorded data about the duration of tasks d, the rate at which tasks were completed γ, and the ratio of drawing tasks to verification tasks τ — to analyse how significant were the participants' disagreements about the correctness/completion of tasks. Furthermore, we separated the data for the local deployment into two parts to fit the two incentive schemes we used. We also captured the results across both of the schemes. The data is reported in Table 1.

Our results show that AMT workers, who completed tasks rapidly, did not perform to a high quality standard. On aggregate AMT workers completed tasks orders of magnitude faster (γ at 44 per minute) than those in the local deployment (γ at 1.7 per minute at best for the Lottery+Competition condition). Moreover, the duration of tasks is significantly lower in AMT (d at 9.4 seconds compared to 23.5 for the Lottery condition) which shows that they are out there to exploit the reward scheme. However, not all tasks were poorly performed though they were worse than in the Local deployment. For example, a study of the votes cast on buildings reveals that the ratio of poorly drawn buildings (as voted down by one or more participants) to the total number of buildings, in the Local deployment is 8%, while in AMT, this number rises to 35%, that is more than four-fold. This is not so unexpected given that significant filters were not applied to AMT workers. However, performing a cost-benefit analysis[12] reveals that the payout for 38,000 micro-tasks would be around £500 for AMT (at \$0.02 per task), while in the Local deployment, the total payout was £700. However, nearly £175 would be lost in the AMT deployment if 35% of the tasks were performed poorly. In turn, in the Local deployment, we would expect the cost per task to drop further if we allowed the deployment to run for an extra day or so. Building more constraints into AMT will obviously increase costs and therefore render it more expensive, though faster than our Local Deployment.

	Combined	Lottery	Lottery+Comp	AMT
d	14.4 ± 0.27	23.5 ± 0.6	11.64 ± 0.23	9.4 ± 0.38
τ	0.35	1.5	0.22	0.24
γ	0.33	0.07	1.7	44

Table 1: Table showing durations (d) in seconds and drawing v/s verification ratios (τ), and task performance rate (γ per minute). 95% confidence intervals around the means are also given for task duration.

In the AMT deployment, we noticed many routes being drawn in similar ways to the kind of routes we obtained from the local deployment. People clearly had different views on what an evacuation route was and where it should lead to as in the local deployment. As can be seen from Table 1, the ratio of tasks to verifications is very similar between AMT and the Lottery+Competition condition (τ is 0.22 and 0.24), whereas the Lottery condition had few participants contributing a lot and was restricted by the workflow from doing verification tasks (hence $\tau = 1.5$). The local deployment significantly gained from the work performed by participant T who, based on his local knowledge, drew routes that were not clearly visible on the aerial view. Finding and exploiting such local knowledge within the timescale of an AMT deployment would be a significant challenge, if not impossible.

Also note the distribution of tasks in Figure 6, which contrasts sharply against the results we obtained in the Local deployment (Figure 5). The reason for this is that many more participants were active in AMT at the same time. While the tasks seem fairly distributed across participants, participant number 6 has a significantly higher number of building drawing tasks. This was due to the participant being the first to enter the system and therefore creating tasks for others.[13]

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented Collabmap, a crowdsourcing platform to help collect data to construct a high-resolution map for disaster simulation. The Collabmap workflow divides the task of creating evacuation routes into micro-tasks that can be performed independently by large numbers of participants. The application was deployed over a period of 3 months with two different incentive schemes, Lottery and Lottery+Competition. Our deployments revealed how hard it was to harness a local community's intrinsic motivations and led us to resort to more extrinsic/social motivations using increased lottery payments and a competition. In what follows we discuss key implications of our work for web science and Human-Computer Interaction (HCI) research, with a particular focus on the use of crowd. We finish with an outline of future work.

7.1 Implications for Web Science Research

Our deployment of Collabmap raises the issue of how crowdsourcing platforms should be evaluated. Our 'in the wild'

[12] We ignore the advertising costs for the local deployment.

[13] The top ten AMT workers completed the following numbers of tasks in total each: 812, 805, 780, 757, 718, 697, 692, 476, 448, 325.

deployment with the local community and University populations (more knowledgeable of web technologies) showed that different arrival rates of participants in the system could significantly affect the participants' interaction with the system; part of which was imposed by the workflow, and part of which is affected by the other participants in the system. Essentially, this close linkage between individual interactions and *system dynamics* point to the need for new tools such as control theory and multi-agent simulation, to quantify the impact of different crowd behaviours (acting as the input) into the system. This cannot be done, however, without a clear understanding of the impact of incentives on user engagement in the specific context of application.

Incentive Schemes Prior to our work, several papers in the HCI and Web communities have looked at incentives in crowdsourcing (see Section 2) and clearly point to the use of extrinsic motivations as the means for maximising task completion and completion rates. It is also well known that trust issues are rampant on platforms such as AMT, though others such as Crowdflower, have mechanisms in place to improve performance. However, the interplay between incentives is rather poorly studied to date and our deployment is but one example of work contrasting incentive types over different time scales and comparing them side-by-side. Our results show that a mix of incentives may be best to get tasks done quickly but also to a high degree of quality. More importantly, targeting populations with different capabilities (e.g., web technology-aware v/s local knowledge aware) and using the best of both could potentially be useful to improve a number of planning-oriented crowdsourcing applications (e.g., emergency planning, redevelopment of run-down areas given pictures on StreetView[14], or remote sensing of land conditions as in GeoWiki). This is in line with insights from behavioural game theory that show how monetary and social motivations can engender different levels of effort [5].

Community-sourcing Harnessing the expertise of a local community was the key goal of Collabmap and, to a limited extent, this was successful in our initial Local deployment. Our work, in this sense resonates with 'Community-sourcing' ideas of Heimerl et al. [3] where a physical device with very specific extrinsic incentives was used. However, going beyond localised interactions such as theirs (which was very successful), in Collabmap we tried to access local communities of experts spread over a region (and who knew their own region well), appealing to their intrinsic motivations and along with some monetary rewards (though with some uncertainty). The cheapest way to access these populations was through the media but this was only partially successful. Hence, we believe that for such large-scale deployments, it is crucial to research better incentive schemes or artefacts that are cheap to deploy and improve user engagement without undermining the quality of tasks.

7.2 Future Work

Future work in Collabmap will look at some of the above research challenges, with a particular focus on combining local knowledge and the speed of AMT. Moreover, we aim to improve the workflow to avoid unfairness in task distribution across workers and also reduce the creation of poor building drawing tasks by AMT workers. To do so, we envisage defining 'gold' tasks [9] that, for such a domain, can be quite challenging to create (cheaply) given the different views people have of what a building or evacuation route is.

Acknowledgments

This work was supported by the EPSRC-Funded ORCHID Project EP/I011587/1 and an EPSRC-funded Knowledge Transfer Secondment EP/C548051/1. We thank Ruben Stranders for initial work on design and implementation and Ian Hoult at Hampshire County Council for his support.

8. REFERENCES

[1] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of UIST 2010*, pages 313–322, 2010.

[2] M. G. Findley, M. C. Gleave, R. N. Morello, and D. L. Nielson. Extrinsic, intrinsic, and social incentives for crowdsourcing development information in uganda: A field experiment. *Working Paper*, 2012.

[3] K. Heimerl, B. Gawalt, K. Chen, T. Parikh, and B. Hartmann. Communitysourcing: engaging local crowds to perform expert work via physical kiosks. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1539–1548, New York, NY, USA, 2012. ACM.

[4] C. Heipke. Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):550–557, 2010.

[5] J. Heyman and D. Ariely. Effort for payment a tale of two markets. *Psychological Science*, 15(11):787–793, 2004.

[6] W. S. Lasecki, R. Wesley, A. Kulkarni, and J. P. Bigham. Speaking with the crowd. In *Proceedings of the Symposium on User Interface Software and Technology (UIST 2012)*, 2012.

[7] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 77–85, New York, NY, USA, 2009. ACM.

[8] J. Nielsen. Participation inequality: Encouraging more users to contribute, 2006. `http://www.useit.com/alertbox/participation_inequality.html`.

[9] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1403–1412, New York, NY, USA, 2011. ACM.

[10] J. C. Tang, M. Cebrian, N. A. Giacobe, H.-W. Kim, T. Kim, and D. B. Wickert. Reflecting on the darpa red balloon challenge. *Commun. ACM*, 54(4):78–85, Apr. 2011.

[11] S. Van Wart, K. J. Tsai, and T. Parikh. Local ground: a paper-based toolkit for documenting local geo-spatial knowledge. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, pages 11:1–11:10, New York, NY, USA, 2010. ACM.

[12] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006.

[13] H. Zhang, E. Horvitz, and R. C. Miller. Crowdsourcing general computation. In *CHI Workshop on Human Computation*, 2011.

[14] H. Zhang, E. Law, R. Miller, K. Gajos, D. Parkes, and E. Horvitz. Human computation tasks with global constraints. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 217–226, New York, NY, USA, 2012. ACM.

[15] M. Zook, M. Graham, T. Shelton, and S. Gorman. Volunteered geographic information and crowdsourcing disaster relief: A case study of the haitian earthquake. *World Medical and Health Policy*, 2(2), 2010.

[14] `www.ratesouthampton.com`