Towards a Unified Spatial Crowdsourcing Platform

Christopher Jonathan and Mohamed F. Mokbel

Department of Computer Science and Engineering
University of Minnesota, MN, USA
{cjonathan,mokbel}@cs.umn.edu

Abstract. This paper provides the vision of a unified *spatial crowdsourcing* platform that is designed to efficiently tackle different types of *spatial tasks* which have been gaining a lot of popularity in recent years. Several examples of *spatial tasks* are ride-sharing services, delivery services, translation tasks, and crowdsensing tasks. While existing crowdsourcing platforms, such as Amazon Mechanical Turk and Upwork, are widely used to solve lots of general tasks, e.g., image labeling; using these marketplaces to solve spatial tasks results in low quality results. This paper identifies a set of characteristics for a unified *spatial crowdsourcing* environment and provides the core components of the platform that are required to empower the capability in solving different types of *spatial tasks*.

1 Introduction

In recent years, crowdsourcing has been gaining a lot of popularity due to its capability in solving various computer-hard tasks, e.g., data labeling [1] and image sorting [2]. The popularity of crowdsourcing can be seen by the existence of several famous commercial crowdsourcing marketplaces, including Amazon Mechanical Turk and Upwork . While these marketplaces are widely used to solve lots of general tasks, many tasks were born to be spatially oriented, namely *spatial tasks*. In such tasks, the location of the worker plays an important role in solving the tasks efficiently. For example, ride-sharing and delivery services can only be done by workers who are located within the area of the tasks, image geotagging tasks will be done more accurately by workers who live near the location of the image, and rating a restaurant would be preferred to be done by local workers. Using the general crowdsourcing marketplaces to solve *spatial tasks* results in low quality results as they are not spatially-aware by randomly selecting workers to solve the tasks regardless of the tasks' location.

The popularity of *spatial tasks*, hindered by the limitations of existing general crowdsourcing marketplaces to support them, urges both industry and academia to provide several *spatial crowdsourcing* solutions where each solution is designed to tackle

This work is partially supported by the National Science Foundation, USA, under Grants IIS-1525953, CNS-1512877, IIS-1218168, and IIS-0952977.
https://www.mturk.com/
https://www.upwork.com/

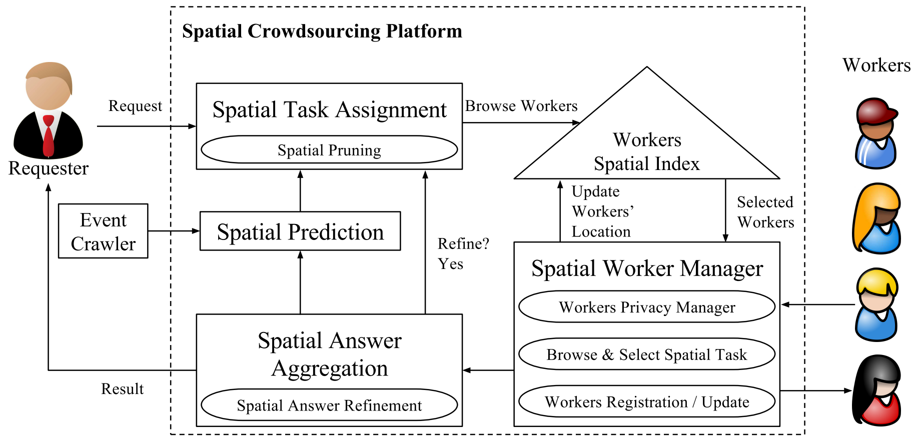

Fig. 1. *Spatial Crowdsourcing* Platform

a specific type of *spatial tasks*. This includes specific solutions for ride sharing [3, 4], crowd sensing [5], and asking workers to go to a certain location to do a task [6, 7]. For surveys of such tasks, see [8, 9]. Despite all such efforts on providing solutions for different kinds of *spatial tasks*, these approaches are not scalable from a system point of view. Currently, to create a new solution for a *spatial task*, developers need to rebuild the same components that have been built by other *spatial task's* solutions. Thus, identifying the core components that can be shared among different *spatial tasks* will allow us to have a single holistic framework that can facilitate and take the burden of recreating the same components for different *spatial tasks* away from the developers.

In this paper, we envision a unified *spatial crowdsourcing* platform that provides a one-stop solution for any *spatial tasks*. Our envisioned platform will be equivalent to Amazon Mechanical Turk, however, tailored for *spatial crowdsourcing*. In particular, a *spatial crowdsourcing* platform will provide the basic modules that are required to efficiently solve *spatial tasks*. Requesters of *spatial tasks* do not need to worry about the details of each internal decision of the *spatial crowdsourcing* process. For example, Uber can use the *spatial crowdsourcing* platform for ride-sharing while other users can use it to translate a document.

2 The Spatial Crowdsourcing Platform

Figure 1 gives the system architecture of our vision for a unified *spatial crowdsourcing* platform. There are two types of users that are using the platform, namely, a requester and a worker. The requester provides a *spatial task* along with the budget that she is willing to pay to the workers (i.e., the crowd) to the platform. Then, the platform reports back the result to the requester. The worker will provide her current location when she

https://www.uber.com/

is available to do some tasks and the platform will provide the tasks that she is qualified for which depends on her current location.

Similar to general crowdsourcing marketplaces, a *spatial crowdsourcing* platform would have three main components, namely, *worker manager*, *task assignment*, and *answer aggregation*. However, all three modules need to be modified to be spatially-aware. In addition, the *spatial crowdsourcing* platform introduces two new components, namely a *spatial index* and a *spatial prediction* module. The details of each component is described briefly below:

Spatial Worker Manager. In general crowdsourcing marketplaces, this module is responsible on maintaining workers' information, broadcasting available tasks to qualified workers, allowing workers to browse available tasks, and reporting workers' answers. In *spatial crowdsourcing* platform, this module will also store the current location information of every worker. It will also be equipped with a *worker privacy manager* submodule to maintain the worker's location privacy.

Workers Spatial Index. Most *spatial tasks* need to be assigned to workers within spatial proximity of the task. Thus, *spatial crowdsourcing* platform incorporates a *spatial index* to manage its workers' locations for fast retrieval. The index can be any types of spatial index, e.g., a quadtree, an R-tree, or a grid index.

Spatial Task Assignment. In general crowdsourcing marketplaces, this module receives a request from the requester and runs a task assignment algorithm to find qualified workers, e.g., workers within a certain age or gender. In *spatial crowdsourcing* platform, this module is modified to be spatially-aware by providing the capability of assigning workers within spatial proximity of the task by exploiting the spatial index. Furthermore, this module is also equipped with a *spatial pruning* submodule that tries to prune out workers that do not need to be checked by the *spatial task assignment* module for a given task, even when the workers are within spatial proximity of the task. Consider an example of a ride-sharing task where the requester provides both pick-up and drop-off locations to the platform. The *spatial pruning* submodule may prune out workers who are currently not traveling towards the drop-off location of the task even when they are currently located near the pick-up location of the task.

Spatial Answer Aggregation. In general crowdsourcing marketplaces, this module receives the results from workers and tries to deduce the final answer and returns it to the requester. In *spatial crowdsourcing* platform, this module is modified to be spatially-aware by taking into consideration the worker's location in evaluating each worker's answer. The closer the distance of a worker to the task, the higher the confidence of the worker in providing the answer is. This module is also equipped with a *spatial answer refinement* submodule to decide whether another crowdsourcing cycle is needed to further spatially refine the result, e.g., to assign workers within a finer spatial granularity in the next crowdsourcing cycle, thus, results in a better quality result.

Spatial Prediction. This module receives a stream of information, e.g., traffic, weather, and event information, in order to help the *spatial task assignment* module to assign more *spatial tasks*. For example, when this module receives an information about a football game that is currently happening at a certain location, it will tell the *spatial task assignment* module to send drivers to the location of the game before the game is finished as there might be many ride-sharing tasks available later.

3 Case Studies

This section discusses three *spatial tasks*: ride-sharing, taking picture of a landmark, and image geotagging, as case studies to show the functionality of the platform.

Case Study 1: Ride-Sharing. Requester provides the pick-up location and the drop-off location to the platform and she will receive the driver identification. The *spatial task assignment* module uses the spatial index to find the closest k workers to the pick-up location and the *spatial pruning* submodule prunes out workers who are currently not traveling in the direction of the drop-off location. The *spatial worker manager* module receives the information about the driver who accepts the task. Then, the *spatial answer aggregation* module returns this information to the requester.

Case Study 2: Taking Picture of a Landmark. Requester provides a landmark with its location to the platform and she will receive a picture of the landmark. The *spatial task assignment* uses the spatial index to find the closest worker to the landmark. The *spatial worker manager* module receives the image from the worker who accepts the task. Then, the *spatial answer aggregation* module returns the image to the requester.

Case Study 3: Image Geotagging. Requester provides an image to the *spatial crowdsourcing* platform and it will return the location of the image. The *spatial task assignment* module uniformly selects workers from around the world. The *spatial worker manager* module receives locations of the image from the workers who accept the task. Then, the *spatial answer aggregation* module assigns a different weight to each worker's answer depending on the distance of her answer to her location. Then, it aggregates the results to find the location of the image and return it to the requester. The *spatial answer refinement* submodule may run another crowdsourcing cycle, however, this time with a more refined location based on the current result, if it is unable to find the location of the image.

References

1. Haas, D., Wang, J., Wu, E., Franklin, M.J.: Clamshell: Speeding up crowds for low-latency data labeling. PVLDB (2015)
2. Marcus, A., Wu, E., Karger, D., Madden, S., Miller, R.: Human-powered sorts and joins. PVLDB (2011)
3. Asghari, M., Deng, D., Shahabi, C., Demiryurek, U., Li, Y.: Price-aware real-time ride-sharing at scale: an auction-based approach. In: SIGSPATIAL. (2016)
4. Cici, B., Markopoulou, A., Laoutaris, N.: Designing an on-line ride-sharing system. In: SIGSPATIAL. (2015)
5. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine (2011)
6. Kazemi, L., Shahabi, C.: Geocrowd: enabling query answering with spatial crowdsourcing. In: SIGSPATIAL. (2012)
7. Chen, Z., Fu, R., Zhao, Z., Liu, Z., Xia, L., Chen, L., Cheng, P., Cao, C.C., Tong, Y., Zhang, C.J.: gmission: A general spatial crowdsourcing platform. PVLDB (2014)
8. Chen, L., Shahabi, C.: Spatial crowdsourcing: Challenges and opportunities. IEEE Data Eng. Bull. **39**(4) (2016) 14–25
9. Zhao, Y., Han, Q.: Spatial crowdsourcing: current state and future directions. IEEE Communications Magazine **54**(7) (2016) 102–107