Crowd-Mapping Urban Objects from Street-Level Imagery

Sihang Qiu
Delft University of Technology
Delft, Netherlands
s.qiu-1@tudelft.nl

Achilleas Psyllidis
Delft University of Technology
Delft, Netherlands
a.psyllidis@tudelft.nl

Alessandro Bozzon
Delft University of Technology
Delft, Netherlands
a.bozzon@tudelft.nl

Geert-Jan Houben
Delft University of Technology
Delft, Netherlands
g.j.p.m.houben@tudelft.nl

ABSTRACT

Knowledge about the organization of the main physical elements (e.g. streets) and objects (e.g. trees) that structure cities is important in the maintenance of city infrastructure and the planning of future urban interventions. In this paper, a novel approach to crowd-mapping urban objects is proposed. Our method capitalizes on strategies for generating crowdsourced object annotations from street-level imagery, in combination with object density and geo-location estimation techniques to enable the enumeration and geo-tagging of urban objects. To address both the coverage and precision of the mapped objects within budget constraints, we design a scheduling strategy for micro-task prioritization, aggregation, and assignment to crowd workers. We experimentally demonstrate the feasibility of our approach through a use case pertaining to the mapping of street trees in New York City and Amsterdam. We show that anonymous crowds can achieve high recall (up to 80%) and precision (up to 68%), with geo-location precision of approximately 3m. We also show that similar performance could be achieved at city scale, possibly with stringent budget constraints.

CCS CONCEPTS

• **Information systems → Geographic information systems**; **Crowdsourcing**; **Web interfaces**; • **Human-centered computing → Collaborative and social computing systems and tools**.

KEYWORDS

Crowd-Mapping, Street-Level Imagery, Task Scheduling, Crowdsourcing, Urban Objects

1 INTRODUCTION

Streets, squares, plots, and buildings constitute the most important elements that shape the physical form of cities [27]. Integral to these elements are city objects such as trees, lamp posts, and other artifacts of urban infrastructure (e.g. benches, fire hydrants etc.). The combination and configuration of physical elements and their constituent objects contribute to the "character" of a city [17] and lead to different formations of the urban fabric [6, 15]. An up-to-date knowledge of the amount and distribution of physical elements and objects across the urban fabric is of vital importance in planning future urban interventions.

At present, the process of constructing such body of knowledge is based on two main approaches. The first, and more traditional one, relies on commissioned authoritative work, by means of on-site documentation conducted by municipal workers [2, 4]. This method usually entails high costs, in addition to being laborious and time consuming. The second approach involves the voluntary contribution of geographic information, which is enabled by collaborative mapping platforms, e.g. OpenStreetMap [5, 12]. On one hand, its participatory and voluntary nature makes it a less costly approach to mapping the urban environment. On the other, it makes it prone to limited participation and, subsequently, to variable spatial coverage and precision [24]. Moreover, it assumes a degree of familiarity of the contributors with the places mapped [5]. Although it has been shown that actual knowledge of a place contributes to the amount and quality of the provided content [13, 24], it could also be seen as a limiting factor, in terms of the number of individuals who can actively create and assemble data about a place.

One way to make contributors familiar with an area of interest, without requiring knowledge at the local level, is by means of street-level imagery. The recent availability of this type of image data through online platforms, such as Google Street View and Mapillary, could enable the documentation of physical elements of the urban environment from a wider pool of potential contributors. Street-level imagery provides panoramic digital representations of – mostly – open public spaces along the street network. Contrary to the widely-used satellite imagery, it simulates the process of a person walking along the streets, thereby providing a detailed three-dimensional overview of visible spatial elements (e.g. building frontages, trees, lamp posts, trash bins, and other objects of the built environment). Given its high penetration rate, currently covering all major cities in the world, it could open up new avenues in perceiving, documenting, and understanding urban systems.

The coupling of street-level imagery with state-of-the-art computer vision and machine learning techniques enables the automatic discovery of highly recognizable recurring objects (e.g. traffic lights, street lamps) [18, 35]. However, with object types characterized by high visual variability (e.g. trees), automatic approaches suffer from severe identification and geo-location accuracy issues.

Recent studies [8–11, 31] have shown that the combination of street-level imagery with micro-task crowdsourcing can successfully engage a larger number of contributors – possibly unfamiliar with the targeted urban area – through object annotation tasks. While demonstrating the feasibility of a micro-task crowdsourcing approach, these works do not fully address the important quality and scalability issues that generally characterize large scale object

annotation campaigns. Two issues are of importance: 1) the precision of object detection and geo-location within an area of interest, and 2) the ability to cover large areas within budget constraints.

Original Contributions. In this paper, we introduce a novel approach to crowd-mapping physical objects of the urban environment. Our method enables the annotation and geo-tagging of urban objects along the street network, through the combination of (a) cost- and worker quality-aware strategies for micro-task scheduling and assignment, and (b) inference methods for object density and geo-location estimation. We develop a dedicated Web user interface that facilitates urban object annotation by crowd workers, using Google Street View imagery as a proxy for real-world places. Annotation, geo-location estimation, and aggregation techniques allow us to map the objects in space, and to estimate their location. A specifically designed scheduling strategy estimates the density of objects in street segments in an iterative fashion, and uses it to prioritize mapping tasks with the aim to maximize the coverage of the mapped objects within budget constraints. Task assignment strategies allocate workers in accordance to their measured annotation quality, so as to improve the geo-location estimation quality.

To evaluate our method, we conducted two experiments. Specifically, we use areas of New York City and Amsterdam as case studies, and focus on the estimation of the number and location of trees along their street networks. Results show that through the proposed Crowd Mapping interface and geo-location estimation techniques, it is possible to engage anonymous crowd workers and achieve high recall (up to 80%) and precision (up to 68%), with geo-location precision of approximately 3m. Using a discrete event simulation technique, we show that it is possible to obtain high object density estimation precision (up to 80%) and promising geo-location precision (up to 85%) at city scale, also with limited budget availability.

The remainder of the paper is organized as follows: Section 2 discusses related work. In Section 3, we present the logical organization of the proposed crowd-mapping method. Section 4 introduces the Web interface for urban object annotation, and describes in detail the methods used to geo-locate and aggregate the labeled annotations. Section 5 describes the developed scheduling, task aggregation, and task assignment strategies. In Section 6, we present the configuration and metrics of the conducted experiments regarding the evaluation of our proposed method. In Section 7, we present and discuss the results of the experiments. Finally, Section 8 summarizes the conclusions and discusses future lines of research.

2 RELATED WORK

We address previous work on crowdsourcing with street-level imagery, and then discuss the relevant work about using street-level imagery to mapping urban objects. We acknowledge the existence of abundant literature related to task assignment and scheduling strategies (e.g. [14, 34]) for crowdsensing. However, as this paper specifically focuses on crowd-mapping with street-level imagery, we consider the analysis of such literature outside of our scope.

2.1 Crowdsourcing using Street-level Imagery

Previous work has addressed the problem of integrating crowdsourcing and street-level imagery, to enable task execution by workers through Web-based platforms. In an early feasibility study [9] manually created a database consisting of 100 images from Google Street View. Researches from the same group afterwards studied the combination of crowdsourcing and Google Street View to identify street-level accessibility problems [10] and developed an online curb ramp data collection system *Tohme* [11]. The system enables workers to remotely label curb ramps through Google Street View with the assistance of computer vision and machine learning techniques. The combination of crowdsourcing techniques and street-level imagery has further applications in risk assessment studies and, particularly, the vulnerability of structures against earthquakes [28]. Relevant techniques have also been applied in medical- and health-related research [3, 33]. Street-level imagery, informed by data from parents of school children, has been used as an efficient alternative to on-site observations for the assessment of the environmental characteristics of cycling-to-school routes [33].

These studies show the feasibility of a crowdsourcing approach to urban scenes and object labeling through street-level imagery. However, issues of accurate object detection and geo-tagging, scalability, and cost received limited attention. To the best of our knowledge, our work is the first to fill this gap.

2.2 Mapping Urban Objects using Street-level Imagery

Street-level imagery is increasingly used in studies concerned with the mapping of urban objects. *Mapillary* [1] is a dataset consisting of 400M street-level images covering 6M kilometers worldwide, in which images are processed with computer vision, and identified objects are connected to open-source and commercial mapping platforms (such as OpenStreetMap, and Here) [26]. Street-level imagery has already been an important data source for mapping urban trees. *Treepedia* is a project estimating the vegetation index of cities. Researchers proposed a method that can automatically detect urban greenery with computer vision by assessing street-level imagery from Google Street View [20, 21]. The urban canopy cover was mapped in Singapore using the hemispherical photographs from Google Street View [29]. Similar works are also conducted in Chinese cities with alternative street-level imagery sources, such as Tencent Street View [23]. It is also used in the detection and mapping of other types of physical objects. Balali et al. used images from Google Street View to detect, classify, and map traffic signs of two interstate highways in the United States [1].

Street-level imagery has also been used in the classification of POI types [25]. Previous work used deep learning techniques to geo-locate and match a street-level image to an aerial one [22].

Work on the automatic analysis of street-level imagery still suffers from geo-localization issues. Moreover, the lack of training datasets results in limited applicability to object types that present high visual variability (e.g. trees). Our work is complementary to these effort, and can simplify the collection of street-level imagery training data for computer vision.

3 METHOD

The Crowd Mapping method proposed in this paper is organized in three sequential steps, elaborated in Algorithm 1: (1) initialisation, (2) Crowd Mapping, and (3) geo-location prediction. The algorithm

[1] https://www.mapillary.com/

takes as *input* a set of n streets $ST = \{st_1, st_2, \cdots, st_n\}$ pertaining to an urban area of interest. Streets are partitioned into segments, where $S(st) = \{s_1, s_2, \cdots, s_l\}$ is the set of segments of a given street $st \in ST$, and $U_S = \{S_1, S_2, \cdots, S_n\}$ is the set of all street segments. The set $E = \{e_1, e_2, \cdots, e_m\}$ defines all the physical objects of a given type that are present in the targeted area; in the scope of our work, each object $e = (lat, lng)$ is described by the World Geodetic System (WGS) coordinates of its lat and lng. We define x_s the *density*[2] of urban objects contained in a street segment $s \in U_S$; $X(st) = \{x_1, x_2, \cdots, x_l\}$ is the corresponding set of segment densities in a street st; and $U_X = \{X_1, X_2, \cdots, X_n\}$ is the universe of all sets densities.

Algorithm 1: The proposed Crowd Mapping method

Input: all the streets ST, workers W, and $budget$.
Output: estimated physical elements of a given type $\hat{E}$

```
// Step 1: Initialization
U_S ← Segmentation(ST);
T ← InitTaskGeneration(U_S);
// Step 2: Crowd Mapping with Street Level Imagery
while budget > TaskMaximumCost() ∨ T = ∅ do
    w ← GetNewWorker(W);
    t ← TaskAssignment(T, w);
    A ← A ∪ AnnotationUI(w, t);
    budget ← budget − TaskCost();
    T ← T − t;
    Ê, U_X̂ ← Aggregation(A);
    H_PS ← SegmentScheduling(U_S, U_X̂);
    T ← T ∪ TaskGeneration(H_PS);
end
// Step 3: Geolocation Prediction
Ê ← Ê ∪ GeoLocationPrediction(Ê, H_PS, ST);
```

Given ST, a set $W = \{w_1, w_2, \cdots, w_i\}$ of available crowd workers, and a *budget*, the goal of our method is to calculate: **(1)** the set $\hat{E} = \{\hat{e}_1, \hat{e}_2, \cdots \hat{e}_m\}$ of estimated physical objects; and **(2)** the set $U_{\hat{X}} = \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n\}$ of estimated segments density. Indeed, $|\hat{E} \cup E| \neq |E|$, as the crowd mapping activity could miss existing objects, or produce non existing ones, due to crowd mapping issues. Conversely, all $\hat{X}_n$ are estimated.

The first step, `Initialisation`, segments each street within ST according to a defined *segmentation* policy (e.g. streets are split in segments of at least 50 meters), to create the set $U_S = \{S_1, S_2, \cdots, S_n\}$ of street segments. Here each $s \in S_n$ is a unit of work. The initial set of crowd tasks $T = \{t_1, t_2, \cdots, t_j\}$ is then created, where each task t contains one or more segments from U_S that are selected and aggregated according to an *initial task generation* strategy (e.g. include one random segment from each street).

The `Crowd Mapping` step, executed until either *budget* is available or $T = \emptyset$, involves the creation of the object *annotations* set $A = \{a_1, a_2, \cdots, a_j\}$ through a Web interface. An annotation $a \in A$ is a *bounding box* drawn on a street-level image by a worker w. Annotations are aggregated by means of an *aggregation function* that progressively populates the set $\hat{E}$, and the set $U_{\hat{X}}$ for the considered segments. The details of the annotation user interface and aggregation strategies are described in Section 4. As segments get annotated, new tasks are created by a *Segment Scheduling* strategy, which establishes the priority of all pending segments into a priority queue H_{PS} that is then used to generate new tasks. A *Task Assignment* strategy distribute tasks to available workers. Section 5 describes the segment scheduling and Task Assignment strategies.

The final `Geo-location Prediction` step enriches the set $\hat{E}$ by predicting the location of urban objects in segments that are not annotated by crowd workers. This could be done by using the density information acquired from the second step, including measured or estimated segment densities, and the location of objects in nearby segments. The adopted geo-location prediction method heavily depends on the targeted urban object, as the regularity of spatial distribution varies significantly among different objects (e.g. street trees and lamp posts are usually more regularly distributed than trash bins). We describe a prediction method in Section 6, where we elaborate on the use case addressed in the experiments.

4 CROWD MAPPING OF URBAN OBJECTS WITH STREET-LEVEL IMAGERY

This section describes the Crowd Mapping step (i.e. Step 2). Section 4.1 describes the Web user interface supporting the *annotation* of urban objects using Street-Level Imagery. Section 4.2 and Section 4.3 respectively elaborate on the methods used to geo-locate and aggregate annotations for urban object mapping.

4.1 User Interface

Figure 1 presents the Web user interface for object annotation. The top-left panel renders the Street-Level Imagery (SLI) exploration widget; the worker can virtually visit an urban area of interest by navigating between *bubbles* [16], i.e. 360° panoramas that provide a photorealistic impression of an area as seen from a given viewpoint. The user can pan and zoom inside the bubble, and move towards contiguous bubbles. The **ANNOTATE** button activates the object annotation interface, described below. The bottom-left panel contains a map of the area under analysis, with the segments pertaining to the current annotation tasks masked. Visual clues provide the workers with an indication of the explored bubbles (red *footprint* dots), and of the location of the provided annotations (green *annotation* dots). The bottom panel contains a task description, a task progress bar, and commands to teleport to the previous or next segments to analyze. The right panel lists all the objects annotated in the current task, and allows the workers to teleport back in the same bubble where the annotation has been made. Finally, the **SUBMIT** button finalizes the task. The interface supports different SLI and map services, including Google Street View.

The annotation interface is central to the mapping task, as it allows for the estimation of an object's geo-location. We assume an object's coordinates to indicate the centroid of its bottom surface. As we show later, this is a precondition for geo-location estimation. Workers are asked to draw the bounding box as described in Figure 2, according to the following steps: (1) Position the mouse pointer at the center point of the object's bottom surface. If this point is not obvious or occluded by other objects, workers should estimate

[2]With a slight abuse of terminology, we refer to *density* as to the normalized number of objects in a given street segment.

Figure 1: Urban Object Annotation Web interface. In the example, workers are asked to annotate *street trees*.

Figure 2: Object Annotation Procedure.

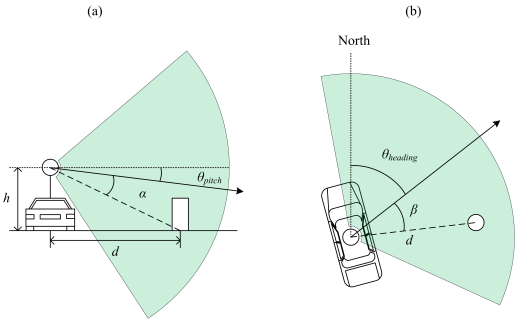

Figure 3: Estimation of the distance between the observation viewpoint and the object's center (a) and geo-location (b).

the position of the point. (2) Click and drag the mouse upwards to draw a symmetrical box that bounds the object.

4.2 Linking Individual Crowd Annotations to Geo-Locations

Street-Level Imagery is typically acquired by vehicles (e.g. cars and bicycles) equipped with 360° image acquisition devices. The geo-location of a bubble (i.e. the virtual observation viewpoint) corresponds to the geo-location of the vehicle at the image acquisition time, as acquired by a GPS device. As shown in Figure 3(a), the distance d between the bubble geo-location and the geo-location of the annotated object can be calculated using $h/\tan(90° - \theta_{pitch} - \alpha)$, where h is the height of the camera installed on the acquisition vehicle. The pitch angle θ_{pitch} (i.e. the up or down angle of the camera) is one of the essential parameters of a street-level panorama image. The angle α and β (in Figure 3(b)) are the angles between the central ray of camera and the ray casting to the bottom of the object in vertical and horizontal plane respectively, which could be estimated from the relative position of the bounding box on the Street View panel.

Figure 3(b) shows how the geo-location of an object could be estimated, using the calculated distance, and the $\theta_{heading}$ could be acquired from parameters of the panorama image. Then, assuming the current latitude and longitude of the observation viewpoint are (lat_{cam}, lng_{cam}), the relative Cartesian coordinate (rx_a, ry_a) (if the coordinate origin is the location of camera) of an annotation a can be calculated as:

$$\begin{cases} rx_a = C_a^x h, \\ ry_a = C_a^y h, \end{cases} \quad \begin{cases} C_a^x = \frac{\cos(90° - \theta_{heading} - \beta)}{tan(90° - \theta_{pitch} - \alpha)}, \\ C_a^y = \frac{\sin(90° - \theta_{heading} - \beta)}{tan(90° - \theta_{pitch} - \alpha)}. \end{cases} \tag{1}$$

where C_a^x and C_a^y are coefficients of the parameters (such as heading, pitch, etc.) of the annotation a, and the arc length of each degree of latitude is approximately $L = 111300$ meters. The WGS coordinate of the annotation a can then be calculated as:

$$\begin{cases} lat_a = lat_{cam} + ry_a/L, \\ lng_a = lng_{cam} + rx_a/[L\cos(lat_{cam})]. \end{cases} \tag{2}$$

Camera Height Correction. The height h of the camera is the only parameter that could not be measured nor estimated from

the user interface. In the case of Google Street View, online documentation suggests that the height of the camera is around 8.2 feet[3]. We however account for different acquisition configurations, and for differences in altitude between the street and the object's bottom surface. The camera height might also be inconsistent due to different vehicles and different acquisition time. Therefore, we estimate an optimal camera height to minimize the measurement error. High accuracy annotations can be acquired for the set of objects $SE \in E$, and the relative Cartesian coordinate (rx_e, ry_e) of the exact location of the urban object $e \in E$ can be expressed as:

$$\begin{cases} rx_e = L(lng_e - lng_{cam})\cos(lat_{cam}), \\ ry_e = L(lat_e - lat_{cam}). \end{cases} \tag{3}$$

Having the Cartesian coordinates of all the real objects in the sample set SE and the corresponding high-quality annotations, the camera height correction minimizes the distances between these pairs (e, a_e). Specifically, the following cost function indicating the mean square error should be minimized:

$$J(h) = \sum_{e \in SE} \{[rx_e - rx_{a_e}(h)]^2 + [ry_e - ry_{a_e}(h)]^2\}. \tag{4}$$

After differentiation, the corrected camera height is:

$$\hat{h} = \frac{\sum_{e \in T}(rx_e C^x_{a_e} + ry_e C^y_{a_e})}{\sum_{e \in T}({C^x_{a_e}}^2 + {C^y_{a_e}}^2)}. \tag{5}$$

4.3 Aggregation of Crowd Annotations for the Location Estimation of Urban Objects

To improve the quality of the estimation, the same object is annotated multiple times, possibly by different crowd workers. While each single annotation might be imperfect – as well as its geo-location estimation – it is reasonable to assume that different labels for the same object will be distributed around its centroid, and that the location of the object should correspond to the highest density of annotations. Traditional density-based clustering algorithms such as DBSCAN are sensitive to a density threshold parameter. We use the clustering algorithm proposed in [30] and extend it to account for the quality $quality_w$ of the worker that performed the aggregation. In Algorithm 2, the density ρ of objects is first calculated (Lines **1–3**). The function $dis(a, a')$ calculates the distance between the estimated geo-locations of a and a', and it is used to estimate the minimum distance δ_a between the annotation a and any other annotation with higher density (Lines **4–6**). Finally, the location of all objects $\hat{E}$ is calculated by selecting annotations whose δ_a is larger than the minimum threshold δ_{min} (Lines **8–14**).

5 SEGMENT SCHEDULING AND TASK GENERATION

To maximize the coverage and quality of estimation within budget constraints, street segments must be scheduled, aggregated in tasks, and assigned to crowd workers.

[3] A Glimpse of Google's Fleet of Camera-Equipped Street View Cars. https://petapixel.com/2012/10/15/a-glimpse-of-googles-fleet-of-camera-equipped-street-view-cars/

Algorithm 2: Geo-location aggregation algorithm

Input: all annotations A, the minimum distance threshold between trees δ_{min}, radius $r = \delta_{min}/2$ for calculating density
Output: estimated geo-locations $\hat{E}$

1 **for** *each annotation* $a \in A$ **do**
2 $\quad \rho_a \leftarrow \sum_{a' \in A: dis(a,a') < r} quality_{w_{a'}}$;
3 **end**
4 **for** *each annotation* $a \in A$ **do**
5 $\quad \delta_a \leftarrow \min_{a' \in A: \rho_a > \rho_{a'}} \{dis(a, a')\}$;
6 **end**
7 $\hat{E} = \emptyset$;
8 **for** *each annotation* $a \in A$ **do**
9 $\quad$ **if** $\delta_a > \delta_{min}$ **then**
10 $\quad\quad lat \leftarrow \sum_{a' \in A: dis(a,a') < r} quality_{w_{a'}} lat_{a'}/\rho_{a'}$;
11 $\quad\quad lng \leftarrow \sum_{a' \in A: dis(a,a') < r} quality_{w_{a'}} lng_{a'}/\rho_{a'}$;
12 $\quad\quad \hat{E} \leftarrow \hat{E} \cup (lat, lng)$;
13 $\quad$ **end**
14 **end**

5.1 Segment Scheduling

The *Segment Scheduling* (Algorithm 1, **Line 10**) operation is responsible for the prioritization of segments to be allocated to crowd masking tasks. Scheduling is logically organized around a priority queue (heap) data structure, which orders segments to be analyzed according to their priority. Segments with the highest priority are at the head of the heap and are, therefore, selected first. Prioritization is generally related to the likelihood of the successful prediction of object densities $U_{\hat{X}}$, while simultaneously optimizing budget consumption. Such likelihood is assumed to be unknown in advance and must, therefore, be calculated dynamically from the annotations produced by crowd workers on processed segments.

We define $AS = \{as_1, as_2, \cdots, as_{l_a}\}, AS \in S$ as the set of segments that have been already annotated by crowd workers; $PS = \{ps_1, ps_2, \cdots, ps_{l_p}\}, PS \in S$ is the set of segments yet to be annotated. Clearly, $AS \cap PS = \emptyset$, $AS \cup PS = S$. Considering spatial dependence, deriving from Tobler's first law of geography [32], we assume that it is more likely for adjacent street segments to feature a similar amount and configuration of urban objects. It is therefore possible, with some degree of uncertainty, to walk a street in a given direction and build a density model for that street. In other words, we can assume that the probability of the object density of a street segment depends on the object density of its previous segment. We can easily calculate the density of the segment $as \in AS$ using the estimated objects $\hat{E}$. The question is how to predict the missing density of the segment $ps \in PS$.

Our solution, called **Maximum Likelihood** (ML) of density prediction is inspired by the *Greedy-based Minimization Entropy* algorithm [14]. We model the object density of street segments as a Markov model. $\hat{X} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_l\}$ is the set of object densities of all segments in S. The probability of object density $\hat{x}$ of a segment s is a function of the object density $\hat{x}'$ of the previous segment s', which is denoted as $p(\hat{x}', \hat{x})$. When the densities $\hat{x}_\sigma$ and $\hat{x}_\tau$ of two segments (s_σ and s_τ) from the same street are calculated, it is therefore possible to estimate the density of the other segments located between them. Thus, the problem is to find the most likely density

sequence $\hat{X}_{\sigma\to\tau} = \{\hat{x}_{\sigma+1}, \hat{x}_{\sigma+2}, \cdots, \hat{x}_{\tau-1}\}$. The estimated density is $\hat{x} = k/length, \hat{x} \in \hat{X}_{\sigma\to\tau}$, where $k \in K = \{0, 1, 2, \cdots, k_{max}\}$ is the possible amount of objects and $length$ is the length of the street segment $s \in S$. k_{max} is the maximum possible amount of object, which is used for limiting the estimation range.

The optimal density sequence is efficiently calculated with dynamic programming. Let $D_{s,k}$ record the maximum likelihood of the sequence (the product of all the transition probabilities) from the crowd-annotated density $\hat{x}_\sigma = k_\sigma/length$ to a possible density $\hat{x}_s = k/length, \sigma < s < \tau$. The final goal is to maximize D_{τ,k_t} and estimate the most likely sequence $\hat{X}_{\sigma\to\tau} = \{\hat{x}_{\sigma+1}, \hat{x}_{\sigma+2}, \cdots, \hat{x}_{\tau-1}\}$. $D_{s,k}$ can be calculated using the following equations (here s' means the previous segment of s):

$$\begin{cases} D_{\sigma,k_\sigma} = 1, \\ D_{s,k} = \max_{k' \in K}\{p(k',k) D_{s',k'}\} \end{cases} \tag{6}$$

Algorithm 3 describes how the most likely sequence is finally calculated. During the process of predicting the density of pending segments, the likelihood π of optimal prediction of the street can be calculated by the following equation:

$$\pi = \prod_{s \in AS : s' \in PS} D_{s', \hat{x}_s' length} \tag{7}$$

Algorithm 3: Dynamic programming. Find the most likely state sequence

Input: Initial density $x_\sigma = k_\sigma/length$ and $x_\tau = k_\tau/length$, the set of possible number of objects $K = \{0, 1, 2, \cdots, k_{max}\}$, transition probability function p

Output: the most likely density sequence $\hat{X}_{\sigma\to\tau} = \{\hat{x}_{\sigma+1}, \hat{x}_{\sigma+2}, \cdots, \hat{x}_{\tau-1}\}$

```
D_{σ,k_σ} ← 1 ;
for each s ∈ {σ + 1, σ + 2, ··· , τ} do
    for each k ∈ K do
        // s' means the previous segment of s.
        D_{s,k} ← max_{k'∈K} {p(k', k) D_{s',k'}} ;
        // Variable pre records the sequence.
        pre_{s,k} ← arg max_{k'∈K} {p(k', k) D_{s',k'}} ;
    end
end
// Get the most like sequence using pre.
k = pre_{τ,k_τ} ;
for each s ∈ {τ − 1, τ − 2, ··· , σ + 1} do
    x̂_s ← k/length ;
    k ← pre_{s,k} ;
end
```

For each street, the scheduling strategy calculates the priority of segments by first calculating the product of likelihood of the optimal prediction of all streets, which is denoted as $\prod_\pi$. Then, we use the predicted density $\hat{x}$ of the pending segment as "true" analyzed data, and take it together with the density values of analyzed segments AS to re-calculate the product of likelihood of the optimal prediction of all streets, which is denoted as $\prod'_\pi$. The final segment priority is then determined as $priority = \prod'_\pi - \prod_\pi$.

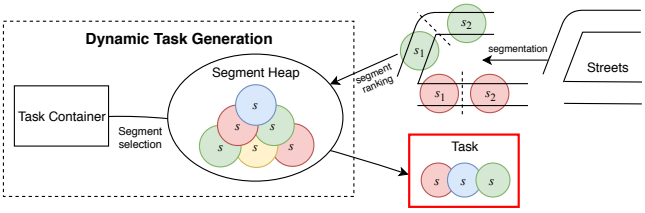

Figure 4: Heap-based Task Generation.

5.2 Task Generation

Task generation is performed 1) at task initialization time (Algorithm 1, **Line 2**), when no prior information about the distribution of urban objects is available; and 2) at run time (Algorithm 1, **Line 11**), after segment scheduling. We employ a *street-based initialization strategy* where, for each street, the first and the last segments of streets are first selected. At run-time, tasks are selected by drawing street segments from the heap, as shown in Figure 4.

As tasks are typically published in batches having the same monetary compensation for workers, it is important to balance the total amount of time needed to complete a task. Segments might have variations in length; their *exploration* might, therefore, require different amounts of time. Similarly, segments contain a varying amount of objects to be annotated. We, therefore, introduce the notion of segment *workload*, which is defined as follows:

$$workload = length + w \cdot \hat{x} \cdot length, \tag{8}$$

where w is used to balance the effort needed for *exploration* (that is proportional to the segment length), and the effort needed for *annotation*, which is proportional to both the segment length and the estimated amount of objects in it. In a set of preliminary experiments conducted with crowd workers of the FigureEight[4] platform, we found that the average time needed for object annotation (8.05 seconds) is 2.5 times longer than the time needed for exploration (3.20 seconds per meter). We therefore set $w = 2.5$. At task initialization, when no prior information is available about the density of objects, the *workload* is calculated only using segment length information.

Segments are, therefore, included in tasks in a greedy fashion, until the sum of their workload reaches a predefined minimum value that is proportional to the targeted total task execution time and compensation.

5.3 Worker Quality-Aware Task Assignment

A task includes several street segments. To acquire annotations for the same street segments and objects, tasks are assigned to multiple workers. The quality of incoming workers is assessed through an initial evaluation task.

Worker Quality Evaluation. After reading the task instructions, the worker is offered an assessment task to perform on segments and objects with known geo-location. The quality of the worker is assessed according to three properties: the annotation recall (R), precision (P), and root mean square error ($RSME$):

[4] https://www.figure-eight.com

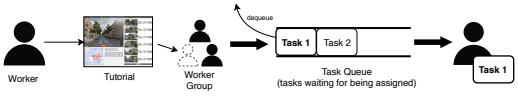

Figure 5: Single-queue Assignment Strategy.

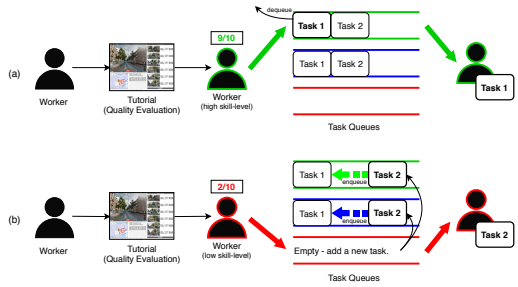

Figure 6: Multi-queue Task Assignment Strategy.

$$R = \frac{\text{\# of annotated objects}}{\text{\# of objects in assessment}} \qquad P = \frac{\text{\# of correct annotations}}{\text{\# of annotations of the worker}}$$

$$RMSE = \sqrt{\frac{\sum (\text{distance from groundtruth annotations})^2}{\text{\# of correct annotations}}}$$

We consider an annotation to be correct if its distance from the targeted objects is lower than a given *threshold*. The worker *quality score* is evaluated by synthesizing these three properties as:

$$\text{quality score} = (1 - RMSE/threshold) + recall + precision \quad (9)$$

Task Assignment Strategies. We consider two strategies.

Single-queue Assignment Strategy. This task assignment strategy is commonly used in microtask crowdsourcing. Tasks are pushed in an execution queue, and then served to workers according to their order of arrival (Figure 5). Tasks are de-queued when the number of required workers is reached.

Multi-queue Assignment Strategy. The final annotation quality can be significantly affected by the performance of their workers. Previous work [36] suggests that the overall quality can be increased by assigning workers with different quality levels, while keeping the average worker quality of each task as high as possible. Inspired by the finding in skill-and-stress-aware assignment [19], the Multi-queue Assignment Strategy divides workers into three groups (*low*, *medium* and *high*) by ranking them according to their quality score, and equally splitting the rank in three groups.

The multiple-queue strategy builds three tasks queue corresponding to three quality levels. After evaluating the quality level of the worker, the task at the head of queue of the same quality level will be de-queued and assigned to the worker (Figure. 6 (a)). If there is no task in the queue of the same quality level as worker's, a new task will be generated and assigned to the worker. Meanwhile, this task will be queued into the other two queues, waiting for being assigned to workers with different quality levels (Figure. 6 (b)).

6 EXPERIMENTAL SETUP

We conducted a number of experiments to investigate: (RQ 1) to what extent can the designed interface, geo-location annotation, and aggregation algorithms lead to correct and complete maps of urban objects when operated by crowd workers from micro-task crowdsourcing platform; and (RQ 2) to what extent the proposed segment scheduling and task assignment strategies can provide correct and complete maps within budget constraints. In this section, we provide details on the case study, the adopted datasets, experimental configuration, and evaluation metrics.

6.1 Case Study: Mapping Street Trees

Urban green infrastructure (e.g. street trees, parks etc.) is an integral set of physical objects and elements comprising the urban fabric [15] [6]. Trees have obvious ecological benefits in terms of air quality and heat regulation, and further contribute to defining the "character" of an urban area. It is therefore important to regularly record information about their location, properties (e.g. tree type), and status for maintenance and planning purposes. American cities such as New York[5] and San Francisco[6] have tree census projects that are run in collaboration, between the municipalities and volunteer organizations. The municipality of Amsterdam, began to record the information about trees since 1875.[7] Although of significant value for cities and communities, tree mapping is still a tedious and expensive task to perform.

6.2 Cities and Datasets

The experiments focus on two regions of two world cities [7], respectively New York City's Manhattan borough, and the area of Amsterdam that is defined by the ring of the A10 highway. We retrieve information about the street networks from OpenStreetMap[8], excluding pedestrian ways, bicycle ways, and other similar types of streets for which associated street-level imagery might not be available. Ground truth data about the location of trees have been collected from the *New York City Street Tree Map* project[9] and the official trees dataset from the municipality of Amsterdam.[10]

We removed from the ground truth datasets trees that are at a distance larger than 30 meters from the street's mid-line, to heuristically exclude trees that could belong to parks or other public spaces (e.g. squares). Table 1 provides basic statistics on the number of streets and the total number of urban trees in the considered datasets. While these datasets are extensive, they are difficult to maintain and can contain errors. The Google Street View panel has been configured to display the latest available version of SLI. We acknowledge the likely presence of temporal mismatches between SLI acquisition time and the dataset update. We address the issue when manually refining the set of street segments used in RQ1 (Section 7.1). However, we ignore such inconsistencies in the

[5] https://tree-map.nycgovparks.org
[6] https://urbanforestmap.org
[7] www.amsterdam.nl/bomen
[8] https://www.openstreetmap.org/. We selected streets whose tag is *motorway*, *trunk*, *primary*, *secondary*, *tertiary*, *residential* or *unclassified*.
[9] https://data.cityofnewyork.us/Environment/2015-Street-Tree-Census-Tree-Data/pi5s-9p35. Last update 2015.
[10] https://maps.amsterdam.nl/open_geodata/. Last update 2017.

experiments addressing RQ2, considering the emerging differences negligible in the context of the experimental goal.

Table 1: Dataset Statistics on the Two Areas of Interest.

# Urban elements and objects	**Manhattan**	**Amsterdam**
# Streets	4081	9084
# Trees	57499	88512
# Segments	18676	23974
Segment Length (m)	44.60 ± 6.45	39.46 ± 10.81
Segment Tree density (#/100m)	3.15 ± 7.34	5.77 ± 11.75

6.3 Configurations

Our crowd-mapping method has a number of parameters that can be configured.

Street Segmentation. We set the maximum length of each segment to 50 meters, to allow for short micro-tasks and reduce annotation fatigue. All segments belonging to the same street have equal length, which is calculated as $length_{st}/(\lceil length_{st}/50 \rceil + 1)$. The resulting number of segments is reported in Table 1.

Camera Height Correction. For each city, we manually select 10 segments with abundant vegetation, and accurately annotate the trees they contain. The locations of these trees are used to estimate the corrected camera heights, which are set to 1.979 and 1.730 meters in Manhattan and Amsterdam, respectively.

Geo-location Prediction. The last step of the crowd-mapping method must be customized for the urban object at hand. For the targeted use case, we designed a simple interpolation method that predicts the location of trees in non-annotated segments from the estimated location of objects in contiguous street segments. To achieve this, we calculate (1) the distance between the centroid of the tree (on both sides of the street) and the mid-line of the street, and (2) the distance between the centroids of adjacent trees in the annotated segments . Based on the corresponding measured distances of the last trees in the annotated segments, we perform the interpolation to estimate the number and location of trees in the non-annotated parts.

Annotation Task Configuration. The urban object annotation Web interface has been deployed on the Figure-Eight platform. For each segment, we collected three judgments. To maximize diversity in the pool of recruited workers while providing sufficient incentives for participation, we allowed workers to perform a maximum of three annotation tasks per city. As a quality control mechanism, workers were allowed to submit the task (and receive compensation) only if the whole area covered by the task segments was explored, and if the exploration of each segment lasted at least 60 seconds. The minimum *workload* for each task has been set to 500 (see Section 5.2), with an average hourly compensation greater than the current US minimal wage. In Figure Eight preliminary experiments, each task contains one segment.

For each city, 50 segments were selected (for RQ1) as follows. We retrieved the vegetation index for each segment in the street networks of New York City and Amsterdam from Google satellite imagery using the vegetation index equation from previous work [21]. We then performed importance sampling on the segments, where the frequencies of samples were sorted according to the vegetation index (importance weight) on the segments. Given that the vegetation index is measured from satellite imagery, it cannot fully reflect the amount of trees at the street level. Therefore, we excluded some street segment that were selected by the sampling method, but were not compatible with our experimental setup. These street segments are usually at the outskirts of the areas in question and contain a large amount of trees that are not included in the ground truth data set. Eight segments from Manhattan and nine segments from Amsterdam were, respectively, excluded, resulting in 42 remaining sampled segments for Manhattan (i.e. 2.2‰of the total number of street segments, 5.4‰of the total number of urban trees) and 41 for Amsterdam (i.e. 1.7‰of the total number of street segments, 3.8‰of the total number of urban trees).

6.4 Evaluation Metrics

We evaluate the performance of crowd workers, annotation aggregation, density prediction, and geo-location prediction against the ground truth in the selected datasets. We consider an annotation *correct* if its distance from the corresponding tree in the ground truth dataset is equal to or lower than 5 meters.

In terms of density estimation, we measure *Density Prediction* (*DP*) and *Density Mean Square Error* (*DE*) as follows:

$$\mathrm{DP} = \frac{|\{s|\text{Density Error} = 0,\ s \in S\}|}{|U_S|} \quad \mathrm{DE} = \hat{x}_s - \frac{|E_s|}{length_s} \tag{10}$$

where E_s is the set of physical objects (i.e. street trees in this case) of the segment s. In terms of object geo-location, we measure *Recall* (R), *Precision* (P), and *Root Mean Square Error* as follows:

$$\mathrm{R} = \frac{\#\text{ annotated objects}}{|E|} \quad \mathrm{P} = \frac{\#\text{ of correct annotations}}{\#\text{ of all aggregated annotations}}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\text{correct annotations}} (\text{annotation error})^2}{\#\text{ of correct annotations}}}$$

7 RESULTS AND DISCUSSION

7.1 RQ1: Crowd Mapping with Street-Level Imagery

A total of 71 and 63 unique workers (Level 3) from Figure-Eight selected the annotation tasks for Manhattan and Amsterdam, respectively. On average, $N = 44$ workers were *Somewhat Satisfied* by the task (3.9/5); they were *neutral* with regard to task difficulty (2.9/5); they found the instructions and the interface *Somewhat Clear* (3.8/5), the quality control *Somewhat Fair* (3.9/5), and the compensation *Somewhat better* than similar tasks (3.8/5).

We manually processed the results. From the Manhattan experiment, we excluded 26 task executions where workers provided no annotations, or the annotation was of bad quality. In Amsterdam, 9 task executions were excluded for the same reasons. With the remaining judgments, we split workers into three quality classes (High, Medium, and Low) of equal size.

Table 2 and Figure 7 show the performance of crowd workers. The precision, recall, and RSME of high-quality individual workers are remarkably high. As expected, worker annotation recall and

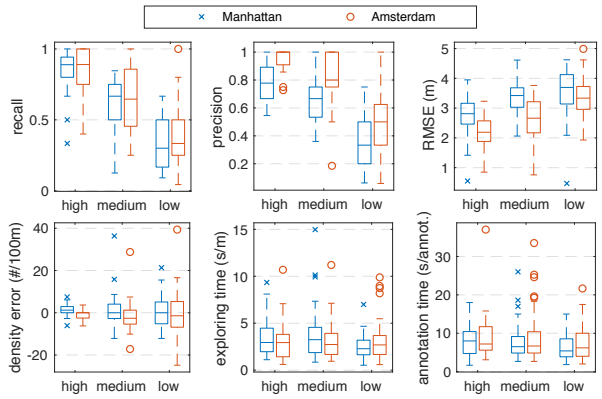

Figure 7: Performance distribution of FigureEight workers.

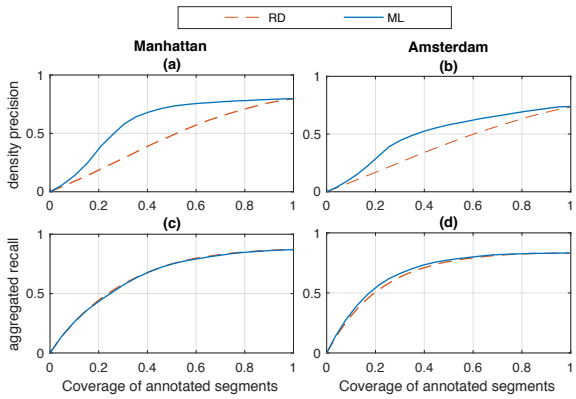

Figure 8: Density precision and Aggregated Annotations Recall for increasing segment coverage with SQ task assignment strategy.

precision decrease with the worker quality level (high to low) in both Manhattan and Amsterdam, while RMSE increases. The resulting density error is good in general, with larger standard deviation at lower quality levels. Interestingly, workers in Manhattan have (on average) a positive density error, whereas the Amsterdam ones have a negative error. A manual inspection of the worker annotations shows no significant difference between workers of similar quality. We, therefore, account the issue to the mismatch between the tree datasets (the latest updated Manhattan dataset is from 2015) and Google Street View imagery. The aggregation method generally mitigates the impact of low-quality annotations, although the precision is negatively affected. Workers of lower quality tend to speed through tasks, but only the difference in exploration time between the low-quality level and the other two levels in Manhattan is of statistical significance (Wilcoxon rank-sum test, $p < 0.1$). Furthermore, the differences in recall, precision, and RMSE across quality levels are also of statistical significance. Other metrics are of no statistical significance across all quality levels ($p > 0.1$).

7.2 RQ2: Segment Scheduling and Task Assignment

To test the performance of the segment scheduling and task assignment strategies, we simulated the behavior of workers by drawing their annotation performance (in terms of precision, recall, and RSME) from the distribution of the workers that participated in the previous experiment. We fed these distributions to a Discrete Event Simulation (DEVS) system that mimics the exploration and annotation behavior of workers of different quality.

When a simulated worker annotates an object, the *Precision* similation parameter decides whether the annotation is correct or not; the error (distance) between simulated annotations and the object following normal distribution $N(0, RMSE)$, while the direction of annotation follows a uniform distribution $U(0, \pi)$. The *Recall* parameter controls the number of correctly annotated objects in a segment. For each experimental configuration, the simulation has been run five times.

Segment scheduling strategy. We first test the performance of the Maximum Likelihood (ML) scheduling strategy (Section 5.1). We employ the single-queue (SQ) task assignment strategy, and compare ML against a random segment scheduling baseline, where segments from U_S are picked randomly. Figure 8 (a) and (b) illustrates how the density precision increases, with regard to increased annotated segments – i.e. allocated *budget*: 100% of segment coverage can be achieved with infinite funds. Note that density precision measures density estimation with *zero* estimation error. ML can achieve greater density precision at lower coverage levels. The maximum performance gain of ML against RD is respectively obtained in at 35% segment coverage, with 35% increase in Manhattan and 22.23% increase in Amsterdam. Table 3 ($\mu \pm \sigma$ of five simulation executions) shows that the average density prediction error also decreases.

Figure 8 (c) and (d), and Table 3 show that ML can provide small improvements in terms of predicted tree locations. This is an expected result, as the Geo-location step implemented in this experiment does not make use of the predicted segment density.

Task assignment strategy. We compare the performance of the multiple-queue (MQ) task assignment strategies against the single-queue (SQ) baseline. In both cases, we employ the ML scheduling strategy, and we simulate an unlimited *budget*. As the MQ task assignment is designed to mitigate the impact of low-quality workers, we progressively skew the distribution of workers towards the low-quality end: in MQ2, half of the workers are of low quality, a third of medium quality, and a sixth of high quality. In MQ3 only low quality workers are simulated. Results ($\mu \pm \sigma$ of five simulation executions) are summarized in Table 4. While recall remains similar to SQ, the precision of the aggregated annotations substantially increases, and the RSME also lowers. Density estimation error and precision also improve. Table 4 reports the performance variations brought by lower quality workers. As the proportion of low-quality workers increases, recall, precision, and RSME progressively worsen. When half of the worker population is of low quality, the recall (-0.03 and -0.01) and RMSE (+0.05 and +0.03) variations in Manhattan and Amsterdam respectively are still acceptable, but the precision is still significantly higher than for the SQ strategy. MQ3 brings more

Table 2: $\mu \pm \sigma$ (average and standard deviation) of recall, precision, RMSE (meter), density error (# of objects per 100 meters), normalized exploration time (second/meter), and normalized annotation time (second/annotation) of Figure Eight workers.

Parameters	Manhattan				Amsterdam			
	High	Medium	Low	Aggregation	High	Medium	Low	Aggregation
Recall	0.85 ± 0.15	0.61 ± 0.19	0.33 ± 0.18	0.75 ± 0.33	0.85 ± 0.17	0.64 ± 0.22	0.38 ± 0.21	0.80 ± 0.23
Precision	0.79 ± 0.14	0.66 ± 0.16	0.35 ± 0.18	0.54 ± 0.20	0.96 ± 0.07	0.81 ± 0.17	0.48 ± 0.25	0.68 ± 0.26
RMSE	2.75 ± 0.62	3.41 ± 0.60	3.35 ± 0.80	3.19 ± 0.73	2.26 ± 0.50	2.59 ± 0.74	3.36 ± 0.70	2.74 ± 0.73
Density error	1.28 ± 2.65	1.56 ± 8.32	0.20 ± 8.20	2.16 ± 4.19	−0.97 ± 2.19	−1.71 ± 7.13	−0.53 ± 10.08	−0.39 ± 5.37
Exploration Time	3.55 ± 1.99	3.88 ± 3.05	2.54 ± 1.27	/	2.97 ± 2.03	3.10 ± 2.07	3.21 ± 2.33	/
Annotation Time	8.06 ± 3.66	7.98 ± 5.00	6.48 ± 3.53	/	8.69 ± 5.71	9.25 ± 7.06	7.82 ± 4.70	/

Table 3: $\mu \pm \sigma$ density error (# objects / 100m), annotation precision and RMSE (meter) of predicted annotations, with SQ task assignment, unlimited $budget$

City	*Strategy*	Density Prediction	Predicted Annotations	
		Error	Precision	RMSE
Manhattan	*RD*	1.22 ± 0.05	0.04 ± 0.00	3.51 ± 0.07
	ML	1.11 ± 0.02	0.04 ± 0.00	3.51 ± 0.02
Amsterdam	*RD*	0.64 ± 0.03	0.05 ± 0.00	3.50 ± 0.01
	ML	0.55 ± 0.03	0.05 ± 0.01	3.51 ± 0.01

extreme performance decay, especially for recall. In terms of density estimation, precision remains generally similar; density error substantially increases only in the MQ3 configuration.

Table 4: Estimation performance with multiple-queues (MQ) scheduling strategy, and unlimited $budget$.

City	*Strategy*	Aggregated Annotations			Object Density	
		Recall	Precision	RMSE	Error	Precision
Manhattan	*SQ*	0.87 ± 0.00	0.44 ± 0.00	3.04 ± 0.01	1.71 ± 0.02	0.80 ± 0.00
	MQ	0.87 ± 0.00	0.67 ± 0.01	2.89 ± 0.01	0.13 ± 0.01	0.84 ± 0.00
	MQ2	0.84 ± 0.00	0.62 ± 0.00	3.09 ± 0.01	0.11 ± 0.03	0.83 ± 0.00
	MQ3	0.60 ± 0.01	0.46 ± 0.00	3.46 ± 0.00	−0.54 ± 0.05	0.82 ± 0.00
Amsterdam	*SQ*	0.83 ± 0.00	0.55 ± 0.00	2.36 ± 0.01	1.10 ± 0.04	0.73 ± 0.00
	MQ	0.83 ± 0.00	0.87 ± 0.00	2.18 ± 0.01	−0.72 ± 0.02	0.80 ± 0.00
	MQ2	0.82 ± 0.00	0.79 ± 0.00	2.39 ± 0.01	−0.69 ± 0.01	0.78 ± 0.00
	MQ3	0.71 ± 0.01	0.58 ± 0.01	3.32 ± 0.00	−1.37 ± 0.09	0.74 ± 0.00

7.3 Discussion

RQ1. Results show that the designed user interface and overall annotation experience have been positively received by workers. The task has been perceived as moderately difficult, yet the overall judgment has been positive. The annotation, geo-location estimation, and aggregation methods also proved reasonably accurate: despite the approximations introduced by the different technologies (e.g. the α and β angles estimated through the position of the bounding box, Figure 3), our interface achieved geo-location accuracy comparable to GPS-enabled smartphones[11]. A main simplification in the proposed approach concerns the estimation of the camera height h (Section 4.2). The difference of the height of the camera with regard to the terrain level of the targeted object could vary greatly, thus leading to inaccurate annotation geo-location estimation. Workers of high quality achieved remarkable precision, recall, and RSME performance. The impact of low quality workers on the aggregated annotations suggest the introduction of better in-task quality control mechanisms. Considering the average exploration and annotation time, the annotation of Manhattan and Amsterdam would take a total of **4.11** and **5.85** man months, and would cost 33K$ and 46K$ respectively.

RQ2. The Maximum Likelihood (ML) scheduling strategy allows for good density prediction and errors, also at low segment coverage levels. While the performance increase as regards the random scheduling strategy is not overwhelming, it is important to notice that the method is not informed by any prior knowledge about the distribution of physical objects in the city. We believe that the integration of third-party data sources in the density prediction could result in better coverage/recall ratio. On the other hand, we observed that the prediction performance of the geo-location estimation step is relatively low. This is neither surprising nor discomforting, as the developed method is rather simple. The multi-queue task assignment strategy brought clear benefits in terms of aggregated annotation precision, and proved reasonably robust to workers with lower annotation quality.

[11] https://www.gps.gov/systems/gps/performance/accuracy/

8 CONCLUSION

In this paper, we have presented an approach to crowd-mapping urban objects from street-level imagery. The proposed method is novel with regard to the combination of an object annotation Web interface for micro-task crowdsourcing with object density and geo-location inference methods. Task scheduling and assignment strategies provide increased performance in terms of cost/quality ratio. Through a use case pertaining to the mapping of urban greenery along streets, we have demonstrated the feasibility of the approach, showing very good annotation, geo-location estimation, and aggregation figures, and promising object density prediction performance. This work is only the first step towards a better understanding of how crowd-mapping with street-level imagery can provide accurate and economic object annotation data. In future work, we plan to improve the density and object geo-location estimation methods by incorporating third-party sources (e.g. greenery indexes) and machine learning techniques for object identification. We believe that active learning techniques and hybrid workflows could lead to more accurate and cheaper annotations. Finally, we aim to extend our experiments to other cities and classes of objects.

REFERENCES

[1] Vahid Balali, Armin Ashouri Rad, and Mani Golparvar-Fard. 2015. Detection, classification, and mapping of U.S. traffic signs using google street view images for roadway inventory management. *Visualization in Engineering* 3, 1 (dec 2015), 15. https://doi.org/10.1186/s40327-015-0027-1

[2] Stefano Bocconi, Alessandro Bozzon, Achilleas Psyllidis, Christiaan Titos Bolivar, and Geert-Jan Houben. 2015. Social Glass: A Platform for Urban Analytics and Decision-making Through Heterogeneous Social Data. In *Proceedings of the 24th*

International Conference on World Wide Web (WWW '15 Companion). ACM, New York, NY, USA, 175–178. https://doi.org/10.1145/2740908.2742826

[3] Chris Clews, Roza Brajkovich-Payne, Emily Dwight, Ayob Ahmad Fauzul, Madeleine Burton, Olivia Carleton, Julie Cook, Charlotte Deroles, Ruby Faulkner, Mary Furniss, Anahera Herewini, Daymen Huband, Nerissa Jones, Cho Wool Kim, Alice Li, Jacky Lu, James Stanley, Nick Wilson, and George Thomson. 2016. Alcohol in urban streetscapes: a comparison of the use of Google Street View and on-street observation. *BMC Public Health* 16, 1 (dec 2016), 442. https://doi.org/10.1186/s12889-016-3115-9

[4] Tim Cresswell. 2014. *Place: an introduction.* John Wiley & Sons.

[5] Michael F Goodchild. 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal* 69, 4 (2007), 211–221.

[6] Cullen Gordon. 1961. The concise townscape.

[7] Peter Geoffrey Hall. 1984. *The world cities.* Weidenfeld & Nicolson.

[8] Kotaro Hara, Jon E. Froehlich, Shiri Azenkot, Megan Campbell, Cynthia L. Bennett, Vicki Le, Sean Pannella, Robert Moore, Kelly Minckler, and Rochelle H. Ng. 2015. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View: An Extended Analysis. *ACM Transactions on Accessible Computing* 6, 2 (mar 2015), 1–23. https://doi.org/10.1145/2717513

[9] Kotaro Hara, Victoria Le, and Jon Froehlich. 2012. A feasibility study of crowdsourcing and google street view to determine sidewalk accessibility. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12*. ACM Press, New York, New York, USA, 273. https://doi.org/10.1145/2384916.2384989

[10] Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, New York, New York, USA, 631. https://doi.org/10.1145/2470654.2470744

[11] Kotaro Hara, Jin Sun, Robert Moore, David Jacobs, and Jon Froehlich. 2014. Tohme: Detecting Curb Ramps in Google Street View using Crowdsourcing, Computer Vision, and Machine Learning. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* Figure 1 (2014), 189–204. https://doi.org/10.1145/2642918.2647403

[12] Darren Hardy, James Frew, and Michael F Goodchild. 2012. Volunteered geographic information production as a spatial process. *International Journal of Geographical Information Science* 26, 7 (2012), 1191–1212.

[13] Brent J Hecht and Darren Gergle. 2010. On the localness of user-generated content. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 229–232.

[14] Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. 2015. Inferring Air Quality for Station Location Recommendation Based on Urban Big Data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*. ACM Press, New York, New York, USA, 437–446. https://doi.org/10.1145/2783258.2783344

[15] Allan B Jacobs. 1993. Great streets. (1993).

[16] Johannes Kopf, Billy Chen, Richard Szeliski, and Michael Cohen. 2010. Street Slide: Browsing Street Level Imagery. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)* 29, 4 (2010), 96:1 – 96:8.

[17] Karl Kropf. 1996. Urban tissue and the character of towns. *Urban Design International* 1, 3 (1996), 247–263.

[18] Vladimir Krylov, Eamonn Kenny, and Rozenn Dahyot. 2018. Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing* 10, 5 (2018), 661.

[19] Katsumi Kumai, Masaki Matsubara, Yuhki Shiraishi, Daisuke Wakatsuki, Jianwei Zhang, Takeaki Shionome, Hiroyuki Kitagawa, and Atsuyuki Morishima. 2018. Skill-and-Stress-Aware Assignment of Crowd-Worker Groups to Task Streams. *Sixth AAAI Conference on Human Computation and Crowdsourcing* (jun 2018). https://aaai.org/ocs/index.php/HCOMP/HCOMP18/paper/view/17921

[20] Xiaojiang Li and Carlo Ratti. 2018. Mapping the spatial distribution of shade provision of street trees in Boston using Google Street View panoramas. *Urban Forestry and Urban Greening* 31 (2018), 109–119. https://doi.org/10.1016/j.ufug.2018.02.013

[21] Xiaojiang Li, Chuanrong Zhang, Weidong Li, Robert Ricard, Qingyan Meng, and Weixing Zhang. 2015. Assessing street-level urban greenery using Google Street View and a modified green view index. *Urban Forestry & Urban Greening* 14 (2015), 675–685. https://doi.org/10.1016/j.ufug.2015.06.006

[22] Tsung Yi Lin, Yin Cui, Serge Belongie, and James Hays. 2015. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 07-12-June-2015. IEEE, 5007–5015. https://doi.org/10.1109/CVPR.2015.7299135

[23] Ying Long and Liu Liu. 2017. How green are the streets? An analysis for central areas of Chinese cities using Tencent Street View. *PLOS ONE* 12, 2 (feb 2017), e0171110. https://doi.org/10.1371/journal.pone.0171110

[24] Afra Mashhadi, Giovanni Quattrone, Licia Capra, and Peter Mooney. 2012. On the accuracy of urban crowd-sourcing for maintaining large-scale geospatial databases. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*. ACM, 15.

[25] Yair Movshovitz-Attias, Qian Yu, Martin C. Stumpe, Vinay Shet, Sacha Arnoud, and Liron Yatziv. 2015. Ontological supervision for fine grained classification of Street View storefronts. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1693–1702. https://doi.org/10.1109/CVPR.2015.7298778

[26] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2017-Octob. 5000–5009. https://doi.org/10.1109/ICCV.2017.534 arXiv:1711.11556

[27] Vítor Oliveira. 2016. *Urban morphology: an introduction to the study of the physical form of cities.* Springer.

[28] S. K. Ploeger, M. Sawada, A. Elsabbagh, M. Saatcioglu, M. Nastev, and E. Rosetti. 2016. Urban RAT: New Tool for Virtual and Site-Specific Mobile Rapid Data Collection for Seismic Risk Assessment. *Journal of Computing in Civil Engineering* 30, 2 (mar 2016), 04015006. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000472

[29] Daniel R. Richards and Peter J. Edwards. 2017. Quantifying street tree regulating ecosystem services using Google Street View. *Ecological Indicators* 77 (jun 2017), 31–40. https://doi.org/10.1016/J.ECOLIND.2017.01.028

[30] Alex Rodriguez and Alessandro Laio. 2014. Machine learning. Clustering by fast search and find of density peaks. *Science (New York, N.Y.)* 344, 6191 (jun 2014), 1492–6. https://doi.org/10.1126/science.1242072

[31] Manaswi Saha, Kotaro Hara, Soheil Behnezhad, Anthony Li, Michael Saugstad, Hanuma Maddali, Sage Chen, and Jon E. Froehlich. 2017. A Pilot Deployment of an Online Tool for Large-Scale Virtual Auditing of Urban Accessibility. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. ACM, New York, NY, USA, 305–306. https://doi.org/10.1145/3132525.3134775

[32] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[33] Griet Vanwolleghem, Delfien Van Dyck, Fabian Ducheyne, Ilse De Bourdeaudhuij, and Greet Cardon. 2014. Assessing the environmental characteristics of cycling routes to school: a study on the reliability and validity of a Google Street View-based audit. *International Journal of Health Geographics* 13, 1 (jun 2014), 19. https://doi.org/10.1186/1476-072X-13-19

[34] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M'Hamed. 2016. Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Communications Magazine* 54, 7 (2016), 161–167. https://doi.org/10.1109/MCOM.2016.7509395

[35] Jan Dirk Wegner, Steve Branson, David Hall, Konrad Schindler, and Pietro Perona. 2016. Cataloging Public Objects Using Aerial and Street-Level Images - Urban Trees.. In *CVPR*. IEEE Computer Society, 6014–6023. http://dblp.uni-trier.de/db/conf/cvpr/cvpr2016.html#WegnerBHSP16

[36] Jianwei Zhang, Yuhki Shiraishi, Katsumi Kumai, and Atsuyuki Morishima. 2016. Real-time Captioning of Sign Language by Groups of Deaf and Hard-of-hearing People. In *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services (iiWAS '16)*. ACM, 54–63. https://doi.org/10.1145/3011141.3011143