Co-Tracking: Target Tracking via Collaborative Sensing of Stationary Cameras and Mobile Phones

ZHIXIONG YU, LEI HAN, QI AN, HUIHUI CHEN, HOUCUN YIN, AND ZHIWEN YU

1College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350002, China
2Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou University, Fuzhou 350002, China
3School of Computer Science, Northwestern Polytechnical University, Xi’an 710072, China
4School of Electronic and Information Engineering, Foshan University, Foshan 266100, China

Corresponding author: Huihui Chen (hhchen@fosu.edu.cn)

ABSTRACT Tracking moving objects in a city, such as suspicious vehicles or persons, is important for public safety management. Traditionally, target tracking is assisted by the pre-deployed stationary surveillance cameras, which are with insufficient coverage. In this work, we propose a different approach called Co-Tracking, a real-time target tracking system that leverages both citizens’ mobile phones and stationary surveillance cameras to track moving objects collaboratively. Two key techniques are focused. Firstly, in order to accurately assign tracking tasks, we propose the Middle Query Location Prediction (MQLP) algorithm for predicting the target’s location. Secondly, in order to efficiently utilizes these human/machine resources, we propose a heuristic algorithm, namely S-Maximum, to optimize the task allocation, including maximizing the number of completed tracking tasks and minimizing the number of mobile phones. Experimental results show that the proposed Co-Tracking system can effectively track moving objects with low incentive costs.

INDEX TERMS Mobile crowdsensing, location prediction, target tracking, collaborative sensing.

I. INTRODUCTION

Target tracking is very important in public safety management. Once a hit-and-run or child abduction happens, the police usually searches the suspect from records of the pre-deployed video surveillance system [1]–[3] and reports from casual witnesses as in AMBER Alert. Since the coverage area of video surveillance system is limited, and the coverage by casual witnesses are not stable, a natural idea is building a more available tracking system that contains not only the stationary cameras, but also the mobile phones carried by ordinary citizens [7]. We call this idea as “collaborative sensing”. It is realizable nowadays because plenty of smartphones move around in the city, and each smartphone is with enough capabilities of sensing, computing, and communication.

However, to track a target via collaborative sensing of stationary cameras and mobile phones, some challenges need to be addressed. First, we need decide where to search the target. Usually we know at least one clue about when and where the target has been in the past, but after a period of time, currently it is missing. Thus a location prediction algorithm is needed to overcome this challenge. Second, within the predicted location area, we need schedule stationary cameras and mobile phones to execute tracking tasks, i.e., to check whether the target appears at the corresponding intersections near their positions. The scheduling algorithm should meet certain requirements such as to complete more tracking tasks with lower tracking cost. We reasonably assume that requisition of a stationary camera is much cheaper than recruitment of a mobile phone (actually we ignore the cost of stationary cameras in this paper). We also assume that stationary cameras or mobile phones have a certain probability to refuse or fail to complete the tracking tasks.

To address the challenges, we develop the Co-Tracking system that illustrates the idea of collaborative sensing, and make further contributions as following:
1. We propose a location prediction algorithm named Middle Query Location Prediction (MQLP) to utilize intermediate query and large-scale crowdsourced objects’ trajectories to predict the location of a moving object after a specific time interval.
2. We propose an algorithm to optimally select stationary cameras and mobile phones, namely S-Maximum, in which a quantitative model is designed to calculate the trust level that a stationary camera or a mobile phone will complete the assigned tracking task.
3. A data set with trajectories of vehicles, check-ins of citizens, positions of stationary cameras and road networks are used to evaluate Co-Tracking.

The following part of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the Co-Tracking framework. Section 4 presents algorithms MQLP and S-Maximum. Then experimental evaluation is provided in Section 5. Finally we conclude the paper in Section 6.

II. RELATED WORK

A. LOCATION PREDICTION

Location prediction and destination prediction are extensively studied in existing work. Based on the temporal and spatial regularity of persons’ daily routes, Chen et al. [4] developed an extended CRPM (Continuous Route Pattern Mining) algorithm to extract movement pattern and built a pattern tree to predict a person’s destination. External information can be added to the historical trajectories to improve the prediction accuracy. For example, Yu et al. [5] proposed the SMLP algorithm, which utilizes the road conditions and driving habits of users to predict their locations based on a Markov model. Xue et al. [6] proposed an algorithm named SubSyn, which decomposes user’s historical trajectories into sub-trajectories, and utilizes sub-trajectories to synthesize new trajectories, thereby expands the scale of the training set and improves the accuracy of destination prediction. Wang et al. [8] proposed the MGDPre method, which only investigates the characteristics of testing trajectory itself without matching with historical trajectories, and it is suitable for sparse datasets. Lian et al. [9], [10] put forward the CEPR algorithm to figure out the most possible locations to visit and offer recommendations based on the collaborative filtering and users’ historical behaviors. In addition, there are many studies related to measuring travel destination estimation, such as real-time destination estimation [32], [33], online trajectory compression [34], and personalized destination estimation [35]. In addition, Yang [36], [37] et al. researches location prediction problems based on Location Based Social Networks (LBSNs), and has achieved excellent experimental results.

Few studies focus on the time-specific location prediction. We propose the MQLP algorithm to address this problem. It uses the Markov probability model to predict the location of the target after a specific time interval, and adds the updated information from the intermediate query in the prediction process to improve the prediction accuracy.

B. TARGET TRACKING

Target detection/tracking in an image or a video is a hot topic in the field of computer vision, e.g., [13]. Here we try to track objects in the real world. Existing work mainly focused on how to deploy camera networks. Wang et al. [1]–[3] utilized the pre-deployed video surveillance system to detect and track moving targets with cameras as few as possible. They proposed different camera coverage models for different tracking scenarios. Zou et al. [12] tracked pedestrians by distributed cameras based on face recognition technology. Aaron et al. [16], [17] described geometric, topological and transitional coverage models for camera networks. Wei et al. [18] proposed a path coverage model in the background of searching for suspects with surveillance camera networks, and achieved the corresponding optimization goal. Pham et al. [19] studied the collaborative coverage and dynamic scheduling of visual sensor networks, focusing on the coverage probability of specific moving targets.

Besides the stationary surveillance cameras, another information source is the mobile phones carried by ordinary citizens. Guo et al. [14], [15] utilized the built-in cameras of smart devices to collect data of targets on the visual crowdsensing platform. CrowdTracker [11] is a target tracking system based on mobile crowdsensing, which recruits people to collaboratively take photos of the target. CrowdTracking [7] can rapidly locate a vehicle by using the photographing contexts and the road network, then estimate the vehicle’s speed according to two successive localization results.

On the one hand, deploying a video surveillance system is expensive but with low flexibility, although requisition is easy after the deployment. On the other hand, recruiting people results in higher incentive costs and takes no advantage of existing cameras. Different from the above work, we propose the idea of collaborative sensing that utilizes both stationary cameras and mobile phones to track moving targets. In such a way, we can make up for the defects of a single sensing source, and improve the quality of target tracking while reducing cost.

C. TASK ALLOCATION

Some work aims to maximize the quality of data collected via mobile crowdsensing while satisfying certain constraints. Reddy et al. [22] proposed a recruitment framework to determine the appropriate participants for data collection based on geographic and temporal availability. Singla et al. [23] proposed a new adaptive participant selection mechanism, which mainly considers the spatial coverage of the task to be maximized while satisfying the total incentive cost constraint. Xiong et al. [24] proposed a task assignment problem whose main optimization goal is to maximize the quality of collected data while meeting a certain cost budget. In addition, a portion of work seeks to minimize the total cost of completing the task while ensuring the quality of collected data.
Karaliopoulos et al. [25] studied the user selection problem based on opportunistic networks and minimized the total cost when all points of interest are covered. Hachem et al. [26] proposed a participant selection framework for mobile crowdsensing system. It predicts each user’s next location based on the user’s current location and historical trajectories, and then selects minimal users who can complete the task by working together.

Only a few work studies the problem of multi-task allocation. Guo et al. [27] proposed a framework for optimizing the multi-task allocation in a mobile crowdsourcing system. Xiao et al. [28] proposed an algorithm to select participants from mobile social networks to minimize the average task completion time. Liu et al. [29] studied platform-oriented multitasking, in which participants can complete tasks as much as possible under time constraints. Different from existing work, our multi-task allocation algorithm is based on collaborative sensing.

### III. CO-TRACKING FRAMEWORK AND WORKFLOW

Co-Tracking is a human-machine collaborative system for object tracking. In this system, two types of resources finish the object tracking tasks collaboratively, i.e., stationary surveillance cameras and people’s smart phones. The stationary cameras are responsible for finding the target from their videos with the machine power so that they can locate the object, while people find the target with the human power and report its position by smart phones.

As shown in Fig.1, Co-Tracking’s workflow consists of four parts, i.e., task release, location prediction, task allocation, and task execution, and it is introduced as follows.

- **Task release.** First of all, the police office releases the tracking/finding task with descriptions of a specific target, including witnessed time, location, and other identifiable features, such as the license plate number, color and size of a vehicle, i.e., initial information. If possible, reference pictures can be also included in this description. With these clues, both ordinary people and intelligent stationary cameras can quickly identify the target if it is in the field of vision.

- **Location prediction.** Secondly, Co-Tracking system predicts the target’s current location. Based on the target’s initial spatiotemporal information provided by the task, we propose MQLP algorithm to compute the target’s current location according to a lot of historical trajectories of objects whose mobility is similar to the target, i.e., taxi dataset. Since using the initial spatiotemporal information is not enough to accurately locate the target, we use intermediate queries to handle this problem. An intermediate query consists of a series of places, and each of them is estimated to be the probable location of the target in a certain duration which is between the witnessed moment and the current moment, i.e., time interval. These places in the intermediate query are chosen to check whether the target has ever appeared there through searching the target from surveillance videos.
captured by cameras in these places. The intermediate query result is then served as a constraint to filter historical trajectories and train or update the prediction models. The result of the location prediction is a 1km × 1km area in this paper.

- **Task allocation.** Thirdly, tracking tasks are assigned to right persons and right surveillance cameras. This process refers to the camera dataset and check-in dataset to complete the experimental evaluation. In order to tracking the target that might appear in several places, multiple tracking tasks with spatial-temporal constraints are created and then assigned. The tracking task will require the assignee (i.e., the phone’s camera or the stationary camera) to check whether the target appears at the corresponding intersection that the people or the stationary camera is (denoted by latitude and longitude) at current moment. Every tracking task has an importance level according to the intersection’s characteristic in the road network. Every tracking task also has a trust level according to the probability of being accomplished in the future, which mostly relates to the assignee’s position. Then the task allocation algorithm S-Maximum is performed based on all tasks’ importance levels and trust levels to both minimize the number of recruited assignees and maximize the number of accomplished tasks.

- **Task execution.** Finally, the task assignees get ready (move if not in position) and pay attention to suspicious objects. Once an assignee sees the target, a picture/video of the target is taken and uploaded to the cloud. As a result, this corresponding tracking task is successfully accomplished. In order to track the object, once the location of the target is refreshed, new tasks will be created and another round of object tracking will begin. Otherwise, if the time of finding the object exceeds the time threshold, the tracking task is considered failed.

Abbreviations and frequently-used notations are listed in Table 1.

### IV. MIDDLE QUERY LOCATION PREDICTION

Middle Query Location Prediction (MQLP) algorithm has two steps. Firstly, we use the Hidden Markov Model (HMM)-based location prediction algorithm to predict multiple locations that the target might has appeared. Secondly, the location prediction algorithm with intermediate queries is used to refine the prediction result and lower the tracking task number.

#### A. HMM-BASED LOCATION PREDICTION ALGORITHM

The Hidden Markov Model (HMM) [30] is used to address location prediction problem and can be easily trained by using large-scale trajectory data. We use the HMM-based location prediction algorithm to roughly predict the location of a moving object after a specific time interval and it is called Time-specific Location Prediction (TLP) algorithm.

| Abbreviation | Explanation |
|--------------|-------------|
| MQLP | Location Prediction Algorithm with Intermediate Queries |
| TLP | Basic Location Prediction Algorithm Based on Markov Model |
| S-Maximum | A heuristic task allocation algorithm |
| l<sub>i</sub> | A location, l<sub>i</sub> ∈ L |
| T<sub>i</sub> | A target-witnessing time point. |
| ΔΤ | A time interval. |
| TPM | The transition matrix of a Markov Model. |
| TR | Vehicle trajectory |
| s<sub>i</sub> | A road intersection. |
| MQLP | Location Prediction Algorithm with Intermediate Queries |

TABLE 1. Abbreviations and frequently-used notations and explanations.

The TLP algorithm considers locations as different states, and the state transition probabilities (i.e., from one location to another location) are only dependent on current state (i.e., current location) for the first-order Markov model. Locations are denoted by L = {l₁, ..., lₙ, ...}. The transition probability of l<sub>i</sub> to l<sub>j</sub> is the fraction of trajectories that happen to pass location l<sub>j</sub> out of all trajectories that pass location l<sub>i</sub>, which is calculated by Equation 1. Equation 2 calculates the transition matrix during the time interval ΔΤ.

\[
P (l_i, l_j) | ΔΤ = \frac{\sum TR_i ∈ TR_l \cup TR_{l+ΔΤ} ∈ TR_j}{\sum TR_i ∈ TR_l}
\]

\[
TPM_{ΔΤ}^{l₁×|L|} = \{p (l_i, l_j) | ΔΤ\}
\]

Once the initial information (including witnessed time T₀ and location l₁) of a moving object is obtained, ΔΤ is calculated by subtracting T₀ from current time. \(p (l_i, l_j) | ΔΤ\) represents the probability that the target moves from location l<sub>i</sub> to l<sub>j</sub> at a certain time ΔΤ. In Equation 2, \(TR_i ∈ TR_l\) represents the target’s trajectory in which contains location l<sub>i</sub> at time \(t\). \(TR_{l+ΔΤ} ∈ TR_j\) represents the target’s trajectory in which contains location l<sub>j</sub> at the next time \(t + ΔΤ\). Finally, we define the \(\sum TR_i ∈ TR_l \cup TR_{l+ΔΤ} ∈ TR_j\) represents the number of target’s trajectories in which the target contains location l<sub>i</sub> at time \(t\) and contains location l<sub>j</sub> at the next time \(t + ΔΤ\). In TPM, one dimension represents the state of the target at l<sub>i</sub> at \(t\), and the other dimension represents the state of the target at l<sub>j</sub> in \(t + ΔΤ\), then \(TPM_{ij}\) represents the probability when the target moves from l<sub>i</sub> in \(t\) to l<sub>j</sub> in \(t + ΔΤ\). The transition matrix denoted by TPM is trained with historical trajectories, then the i-th row of the transition probability matrix is the location prediction result, and we can sort locations according to the object appearing probabilities. The location with the largest probability is the next location predicted by TLP.
B. LOCATION PREDICTION ALGORITHM WITH INTERMEDIATE QUERIES

The newer the object’s location we know, the more accurate the next location is predicted. Although stationary cameras and mobile phones are dense enough to catch the object’s past clues by coincidence, these image/video records are massive and with general purpose. For instance, they are not indexed to find whether this object is in a record or not. Therefore, the target-finding is only available when we “query” the record of a location of a time and MQLP is proposed for this purpose.

For the purpose of verifying the estimated target’s trajectory, $\Delta T'$ is the intermediate time to query. We also need to know which roads should be queried. TLP algorithm returns $N$ most promising locations, denoted by $L$, hence both image and video records of cameras that cover these locations are queried. The query result of the location set $L$ and the time interval $\Delta T'$ is denoted as $q(L, \Delta T')$, as Equation 3 shows.

$$q(L, \Delta T') = \begin{cases} 
1, & \text{the target has appeared in a location in } L \text{ at } \Delta T' \\
0, & \text{the target has not appeared in any location in } L \text{ at } \Delta T'
\end{cases} \tag{3}$$

The query result is taken as the condition, and each training trajectory is classified according to whether it meets the same condition. We will train a transition matrix based on trajectories of each class by Equation 4. As Algorithm 1 show, the transition matrix will be used to predict the object’s location at $\Delta T$ of the testing trajectory. Based on these trained models, Co-Tracking predicts the next location of a target if the current location is obtained, and then the tracking tasks will be created and assigned, which will be introduced in the next section.

$$p(l_i, l_j|\Delta T, q(l, \Delta T')) = \frac{\sum_{TR_j \in TR_i \cup TR_{t+\Delta T}} q(l, \Delta T') = \{1 \lor 0\}}{\sum_{TR_j \in TR_i \cup q(l, \Delta T') = \{1 \lor 0\}}, t = t_1, t_2, \ldots} \tag{4}$$

To use MQLP for prediction, we first need to make a query to determine whether the target satisfies the query conditions at the time $\Delta T'$. When the value of the query result is 1, we substitute the target position into a model that matches the “query to” result to predict. When the query result is 0, the target position is substituted into the “not queried” model for prediction. Its prediction algorithm is given in Algorithm 2.

V. HUMAN-MACHINE COLLABORATIVE TRACKING

The human-machine collaborative tracking requires a specific task allocation algorithm that takes advantage of both smart phones and stationary cameras. This algorithm is based on the importance level of intersections and the trust level of either stationary cameras or mobile phones. A heuristic algorithm will be proposed to optimize the task allocation in this section.

![Algorithm 1 MQLP-Training](image)

A. IMPORTANCE LEVEL AND TRUST LEVEL

1) MODELING THE IMPORTANCE LEVEL OF INTERSECTION

We assume that objects are moving along with the road network. The road network is consisted of roads (i.e., edges) and intersections (i.e., nodes). We only track objects at intersections.

In order to cover the predicted location area, different intersections have different importance, which is determined by their geographic characteristics, including: the number of connected roads (e.g., an intersection connecting four roads is more important than that connecting three roads), the grade of connected roads (e.g., the higher grade, the more important), and the length of the connected road (e.g., the longer, the more important). Considering the above factors, we use Equation 5 to calculate the importance level of an intersection [31], denoted by $s_i$.

$$M(s_i) = \sum_{t=1}^{k(i)} c_t \left(1 + \omega \frac{l_i - l_{\min}}{l_{\max} - l_{\min}}\right) \tag{5}$$

where $k(i)$ is the number of roads connected by $s_i$, $c_t$ and $l_i$ are the grade and the length of the $t$-th road, $l_{\max}$ and $l_{\min}$ are the maximum and minimum road length of the entire road network in the considered location area, and $\omega$ is the contribution weight of the road length to the intersection importance.

2) MODELING THE TRUST LEVEL OF STATIONARY CAMERA

A stationary camera is used to monitor an intersection or a road. The trust level of a camera to an intersection refers to
Intersection monitoring of a stationary camera.

The smallest distance ($Y$) and the largest distance ($T$) that the camera can monitor the intersection effectively are demonstrated in Fig.2. Denote the wide of the intersection ($H$), the performance parameters of the camera such as the focal length ($f$), the image wide ($w$) and the horizontal resolution ($r$ pixels), and the recognizability requirement (at least $P_{\text{min}}$ pixels for the intersection in the image).

We assume the angle of the camera is direct from the camera to the intersection. According to the imaging principle of a lens, we have the following equations:

$$\frac{Y}{H} = \frac{f}{w}$$  \hspace{1cm} (6)

$$\frac{T}{X} = \frac{f}{w}$$  \hspace{1cm} (7)

$$X = \frac{H}{r}$$  \hspace{1cm} (8)

where $X$ is a temporary variable that means the covered wide of the camera at the distance of $T$. Then we get $Y = \frac{f + h}{w}$ and $T = \frac{r + f - h}{w}P_{\text{min}}$.

When the distance between the camera and the intersection $d < Y$, the covered width of the camera is $h = \frac{wd}{f}$. The monitoring probability is reasonably the ratio of $h$ and the intersection width $H$. Thus the trust level of this camera is:

$$\frac{h}{H} = \frac{wd}{f} \Rightarrow \frac{d}{Y}$$  \hspace{1cm} (9)

When $Y < d < T$, the recognised ability of the intersection in the image cannot meet the minimum requirement, but the monitoring probability may be not 0, because sometimes, the camera is on a road linked to the intersection, which can be considered as monitoring the intersection indirectly. For simplicity, the possibility is equally divided among the $K$ roads which linked to the same intersection, i.e. $1/K$.

In summary, the trust level of a stationary camera $c_j$ to an intersection $s_i$ can be calculated as:

$$B_i(c_j, s_i) = \begin{cases} 
\frac{d}{Y} & \text{if } d \leq Y \\
1 & \text{if } Y < d \leq T \\
\frac{1}{K} & \text{if } d > T, \text{ but } c_j \text{ is on a road linked to } s_i \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (10)

3) MODELING THE TRUST LEVEL OF MOBILE PHONE

The trust level of a mobile phone refers the possibility to take a picture of the target if the target is going through the intersection. This possibility is also most related to the distance (denoted as $d_1$) between the intersection and the mobile phone, because if the distance is too large, the owner of the mobile phone is reluctant to move to the intersection, or cannot reach the intersection in time. The distance threshold it denoted as $D_{\text{max}}$.

Gravity model is usually used to simulate how distance influence the willingness that people moving from one position to another. We adopt $e^{-\beta d_1}$ to calculate the trust level of a mobile phone to an intersection if $d_1 \leq D_{\text{max}}$, where $\beta$ is the attenuation coefficient. Like cameras, if the mobile phone cannot reach an intersection in time (i.e., $d_1 > D_{\text{max}}$), it can try to reach a road instead, but with discounted possibility to monitor the intersection indirectly. Let the distance between the mobile phone and the road is $d_2$, if $d_2 \leq D$, $e^{-\beta d_2}/K$ is adopted, where $K$ is the number of roads that linked to the same intersection.

Algorithm 2 MQLP-Prediction

Input: QTMP: Transition probability matrix with query result 1, QTMP': Transition probability matrix with query result 0, $\Delta T$: time interval, $\Delta T'$: Query time, (lon,lat): Initial position

Output: Top N prediction regions

1. list $\leftarrow$ null; //The list is used to save predicted results
2. FOR EACH target trajectory $T_R \in TR$:
3. initial location $l_i$, and after $\Delta T'$ the location is $l_j$;
4. get the state $q(L, \Delta T')$ of the target at time $\Delta T'$ according to TLP;
5. IF $q(L, \Delta T') = 1$ THEN do
6. Find the row-data corresponding to $l_i$-th row in QTMP and save it to the list;
7. ELSE
8. Find the row-data corresponding to $l_j$-th row in QTMP' and save it to the list;
9. ENDIF
10. Sort list; // The column numbers corresponding to the first N elements in the list are the first N prediction regions
In summary, the trust level of a mobile phone \( p_j \) to an intersection \( s_i \) can be calculated as:

\[
(p_j, s_i) = \begin{cases} 
  e^{-\beta d_1}, & \text{if } d_1 \leq D_{\text{max}} \\
  e^{-\beta d_2} \cdot \frac{1}{K}, & \text{if } d_1 > D_{\text{max}} \text{ AND } d_2 \leq D_{\text{max}} \\
  0, & \text{if } d_2 > D_{\text{max}}
\end{cases}
\]  \( (11) \)

B. COLLABORATIVE TASK ALLOCATION ALGORITHM

1) OPTIMIZATION OBJECTIVE

In the predicted location area, the road network is consisted of roads and intersections \( (S = \{s_1, s_2, \ldots, s_i, \ldots\}) \), and for each intersection there is a tracking task that need to be assigned. We aim to complete more tasks with less mobile phones (since we ignore the cost of repositioning a stationary camera). Completing more tasks not only means the number and the possibility, but also the importance. To this end, the cost effectiveness (see Equation 12) is adopted as the metric to evaluate the performance of different task allocation algorithms. Let \( C = \{c_1, c_2, \ldots, c_j, \ldots\} \) the camera set, and \( P = \{p_1, p_2, \ldots, p_i, \ldots\} \) the phone set. \( \hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_i, \ldots\} \) is the selected mobile phones. Apparently all cameras should be selected to maximize the completed tasks.

\[
B \left( s_i \right) = \begin{cases} 
  B_s(c_j, s_i), & \text{if } c_j \text{ monitors } s_i \\
  B_p(\hat{p}_j, s_i), & \text{if } \hat{p}_j \text{ monitors } s_i 
\end{cases}
\]

\[
\text{CostEff} = \frac{\sum_i M(s_i) \times B(s_i)}{|P|}
\]  \( (12) \)

In order to maximize CostEff, a heuristic task allocation algorithm called S-Maximum are proposed.

2) HEURISTIC ALGORITHM

Obviously, it is necessary for Co-Tracking to get the positions of pedestrians’ mobile phones (from check-in dataset in this paper) and the positions of stationary cameras (from camera dataset in this paper). For any task/intersection that is within a camera’s field of vision, we assign it to that camera first. Those tasks outside the range of cameras are then assigned to mobile phones. This is how to achieve collaborative sensing. The owners of the selected mobile phones will (or might not) move to the tasks’ intersection to execute tasks respectively, if there are distance between their own positions and the tasks’ intersections.

As shown in Algorithm 3, S-Maximum ensures that mobile phones are preferentially assigned to important intersections.

Firstly, select the intersection \( s_i \) with the largest importance level from all intersections \( S \). Then, the set \( UT = \{u_{i1}, u_{i2}, u_{i3} \ldots\} \) of the completed task \( i_t \) is obtained, and the task \( i_t \) is deleted in the task set, and if \( u_{ij} \) is a candidate, the candidate is deleted in the candidate set. Similar to the process of selecting assignees for subtask \( i_t \), iteratively selects assignees for each subtask until all subtasks are executed. Finally, we obtain the optimal set of candidates to perform the task. The threshold \( t_e \) means only stationary cameras or mobile phones with trust level higher than \( t_e \) are taken into consideration.

A baseline algorithm for task allocation is S-Random. It is the same as S-Maximum except in each iteration, we firstly select an intersection randomly, rather than that with the largest importance level.

VI. EVALUATION

In this section, we conducted extensive experiments to evaluate the performance of Co-Tracking system. We validate Co-Tracking from two aspects: the performance of the location prediction algorithm MQLP (in terms of accuracy) and the performance of the task allocation algorithm S-Maximum (in terms of cost effectiveness, the number of completed tasks, the number of selected mobile phones, and the running time).

A. DATA SETS

Data sets consist of taxi trajectories, road networks, positions of traffic cameras in Chengdu, China, and people’s check-ins on the social network in Chengdu. The taxi trajectory data set contains 19,000 taxis’ trajectories in 17 days in August 2014 covering 1,254 square kilometers in the city of Chengdu, China. This area is divided into 33 \( \times \) 38 grids, and each grid is a 1km \( \times \) 1km area. All points of trajectories, camera positions and check-ins’ locations are projected into grids.

B. THE EFFECTIVENESS OF MQLP

In the experiment, the intermediate query function can be realized by looking up the historical record of the test trajectory instead of the camera network data in actual application. In other words, we get the query results based on the history of the test trajectory.

In order to verify the performance of the MQLP, the time interval \( \Delta T \) is taken as 10, 20, 30, 40, 50, and 60 minutes, and the midpoint of the time interval is used as the query time, and the number of the queried regions is \( N = 5 \)
FIGURE 3. Change of the predicted time.

(i.e., top 5 most likely regions output by TLP). Compared with the TLP algorithm, the results are shown in Fig.3 (a). We can see MQLP is better than TLP, and the smaller query time, the greater accuracy increase.

The performance of the MQLP algorithm depends on the number of test cases with the query result being 1 (i.e., the object is in one of the top 5 regions). When the query result is 1, the time interval corresponding to the prediction becomes $\Delta T - \Delta T'$. We set $\Delta T$ to 30 minutes, $\Delta T'$ are 5, 10, 15, 20, 25 minutes respectively. In the case where the query result is 1, the predicted time is equivalent to 25, 20, 15, 10, 5 minutes. The performance of MQLP is compared to the TLP method as shown in Fig.3 (b). The results show that when the query result is 1, the prediction accuracy of MQLP is also better than the TLP algorithm.

For the MQLP algorithm, the impact of the query time $\Delta T'$ and the number of query regions on the performance of the algorithm has instructive significance for people to search missing object. Therefore, it is necessary to discuss the influence of the query time and the number of query regions on the searching effect.

The first is the influence of the number of different query regions on the prediction results. As shown in Fig.4 (a), the accuracy of the algorithm when $\Delta T$ sets to 10 minutes and 20 minutes respectively, and $\Delta T'$ sets to the midpoint. The prediction accuracy increases as the number query regions increases. When fixing $\Delta T$ to 20 minutes, and $\Delta T'$ to 10 minutes, and the number of query regions N is 1, 2, 3, 4, 5 respectively, the running time of the algorithm is shown in Fig.4 (b). As the value of N increases, the running time increase linearly.

In order to compare the impact of different query times on the query effect, the query time $\Delta T'$ is set to 5, 10, 15, 20, 25 minutes, respectively, and the object’s region after 30 minutes is predicted. Fig.5(a) shows the total accuracy of different query times. In general, when the number of query regions is the same, the closer the query time is to the predicted time, the better the prediction effect. Fig.5(b) considers the accuracy of MQLP when the query result is 1. When the query time $\Delta T'$ is 5, 10, 15, 20, 25 minutes, the predicted time is 30 minutes, the query result is 1, and the number of query regions is the same, the prediction performance is better as the query time is closer to the prediction time. Compared to the number of query regions, the query time has a greater impact on the query effectiveness.

C. THE PERFORMANCE OF TASK ALLOCATION ALGORITHMS

We assume that the road is three-lane with each lane width of 3.5 m. The camera has a horizontal length of 4.8 mm and a focal length of 16 mm. The horizontal resolution is 2448. The license plate width is set to 0.44 m and the horizontal pixel is 75. The pedestrians’ speed is 1 m/s. The predicted time T is set to 10 minutes.

The more candidates of assignees (stationary cameras or mobile phones), the more opportunities you can choose, and the greater the chances of choosing the best assignees. The threshold $t_s$ is set to 0.6, and different number of candidates are randomly selected from the check-in data. The Fig.6 shows task completion, the number of assignees, the ratio of task completion to the number of people is getting larger. S-Maximum is more stable, while S-Random is fluctuating. When the number of candidates is small, prioritizing the assignment of more important tasks can improve the overall performance. When the number of candidates is large, S-Maximum and S-Random have similar performance. This is because when there are a large number of candidates, it is possible to cover all tasks. S-Random takes a bit less time. Therefore, in practical applications, using S-Maximum when the number of candidates is small can improve the performance of task allocation. When the
number of candidates is large, S-Random can be used if the running time is crucial.

In the process of task allocation, the larger the threshold $t_s$ means that the choice requires greater trust level, and more assignees should be selected to complete a subtask. A smaller threshold means fewer assignees are able to complete a subtask. Therefore, the difference in threshold size will affect the final task assignment effect. Since the number of candidates has a great impact, the number of candidates sets to 100 and 400 respectively, and experiments are performed under different thresholds $t_s$. The Fig.7 and 8 show task completion, the number of assignees, the ratio of task completion to the number of assignees (i.e., cost effectiveness), and running time for different thresholds.

When there are fewer candidates, the smaller the threshold, the better the effect of the allocation. As shown in the Fig.7, as the threshold increases, the ratio of task completion to the number of assignees decreases. The number of assignees and running time are increasing. When there are fewer candidates, the threshold $t_s$ has a greater impact on the results. This is due to the probability of ensuring that each task is completed. The number of assignees who completed a single task was increased, resulting in no candidates available for the tasks remained.

As shown in the Fig.8, as the threshold increases, the task completion increases, and the number of assignees increases. Specially, when $t_s$ exceeds 0.7, it increases faster. The ratio of task completion to the number of assignees first increase and then decrease. At 0.6, the maximum value is reached, and it increases slowly before 0.6, and then drops rapidly after 0.6.

Combined with the above experimental results, when the number of candidates is small, a smaller threshold should be taken. When the number of candidates is large, the threshold should be increased, but not as large as possible. Experiments show that the value had better be between 0.6 and 0.8.

There are different distributions of roads in different regions in urban road networks. In order to verify the effectiveness of the algorithm, 13 regions were randomly selected in all regions (see Table 2), and experiments are performed using S-Maximum in these regions. Fig.9 shows the results of task completion in each region. Above 85% tasks are completed. Generally, more cameras, more pedestrians and less intersections (tasks) means we can complete tasks easier.

**D. DISCUSSIONS**

According to the above experiments, we find that the performance of Co-Tracking system would be influenced by
some factors such as the object movement prediction and the number of candidates. In this section, we discuss the limitations of our current work.

1) REGIONS FOR LOCATION PREDICTION
In real life, moving targets are usually on the road, dividing the city into grids will destroy the structure of the city's road network.

2) THE MEASURE OF TASK IMPORTANCE
When calculating a task importance, it is also possible to consider the characteristics of the traffic flow. On the other hand, depending on the topology nature of the road network, some tasks can be reduced.

3) THE MEASURE OF TRUST LEVEL
When calculating the trust level of a camera, the direction and the parameters of the camera are assumed. However, in actual scenarios, they are different. When calculating the trust level of a pedestrian, it is regarded as following the power law distribution of the distance. However, the trust level of a pedestrian is also related to his/her historical trustiness record. In addition, according to the historical trajectory of the pedestrians, predicting his/her destination can increase the practicality.

VII. CONCLUSION AND FUTURE WORK
This paper introduced the Co-Tracking system, which uses fixed nodes (road camera) and mobile nodes (pedestrians and their mobile phones) to collaborate on taking photos or videos to track specific moving objects. First of all, we proposed the location prediction MQLP model for a specific time. Next, the calculation model of trust degree is proposed to estimate the probability of a candidate executing a task, and two allocation algorithms are proposed for human/machine resources to track the object in the predicted region. Finally, we evaluated the Co-Tracking system using large-scale real-world dataset. Experimental results indicate that the proposed system can effectively track moving objects with low incentive costs.

The merits of collaborative sensing include not only to compensate the coverage area of each sensing approach, but also compensate in the quality, features etc. we will explore these later. In the future we intend to conduct user studies to evaluate the performance of the algorithm in the real environment.

REFERENCES

[1] L. Liu, X. Zhang, and H. Ma, “Dynamic node collaboration for mobile target tracking in wireless camera sensor networks,” in Proc. IEEE 28th Conf. Comput. Commun. (INFOCOM), Apr. 2009, pp. 1188–1196.

[2] H. Ma, M. Yang, D. Li, Y. Hong, and W. Chen, “Minimum camera barrier coverage in wireless camera sensor networks,” in Proc. IEEE INFOCOM, Mar. 2012, pp. 217–225.

[3] Y. Wang, D. Wang, and W. Fang, “Automatic node selection and target tracking in wireless camera sensor networks,” Comput. Electr. Eng., vol. 40, no. 2, pp. 484–493, Feb. 2014.

[4] L. Chen, M. Lv, and G. Chen, “A system for destination and future routine prediction based on trajectory mining,” Pervas. Mobile Comput., vol. 6, no. 6, pp. 657–676, Dec. 2010.

[5] R. Y. Yu, X. Y. Xia, and X. Li, “Position prediction algorithm for mobile users based on social relations in participatory perception systems,” Chin. J. Comput., vol. 38, no. 2, pp. 374–385, 2015.

[6] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu, “Destination prediction by sub-trajectory synthesis and privacy protection against such prediction,” in Proc. IEEE 29th Int. Conf. Data Eng. (ICDE), Apr. 2013, pp. 254–265.

[7] H. Chen, B. Guo, Z. Yu, and Q. Han, “CrowdTracking: Real-time vehicle tracking through mobile crowdsensing,” IEEE Internet Things J., vol. 6, no. 5, pp. 7570–7583, Oct. 2019.

[8] L. Wang, Z. Yu, B. Guo, T. Ku, and F. Yi, “Moving destination prediction using sparse dataset,” ACM Trans. Knowl. Discovery from Data, vol. 11, no. 3, pp. 1–33, Apr. 2017.

[9] D. Lian et al., “Analyzing location predictability on location-based social networks,” in Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science), vol. 8443, V. S. Tseng, T. B. Ho, Z. H. Zhou, A. L. P. Chen, and H. Y. Kao, Eds. Cham, Switzerland: Springer, 2014.

[10] D. Lian, X. Xie, V. W. Zheng, N. J. Yuan, F. Zhang, and E. Chen, “CEPR: A collaborative exploration and periodically returning model for location prediction,” ACM Trans. Intell. Syst. Technol., vol. 6, no. 1, pp. 1–27, Apr. 2015.

[11] Y. Jing, B. Guo, Y. Liu, Z. Wang, Z. Yu, and X. Zhou, “CrowdTracker: Object tracking using mobile crowd sensing,” in Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. Proc. ACM Int. Symp. Wearable Comput. (UbiComp). New York, NY, USA: ACM, 2017, pp. 85–88.
C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, “TrajCompress: Target Tracking via Collaborative Sensing of Stationary Cameras and Mobile Phones,” IEEE Access, vol. 8, 2020.

ZHIYONG YU received the M.E. and Ph.D. degrees in computer science and technology from Northwestern Polytechnical University, Xi’an, China, in 2007 and 2011, respectively. He is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He was also a Visiting Student at Kyoto University, Kyoto, Japan, from 2007 to 2009, and a Visiting Researcher at the Institut Mines-Telecom, TELECOM SudParis, Evry, France, from 2012 to 2013. His current research interests include pervasive computing, mobile social networks, and mobile crowd sensing.

QI AN received the M.E. degree in computer application technology from Northwestern Polytechnical University, Xi’an, China, in 2019.

HUIHUI CHEN received the Ph.D. degree from Northwestern Polytechnical University, China, in 2017. She is currently a Lecturer at Foshan University, China. Her research interests include ubiquitous computing and mobile crowdsensing.
HOUCHUN YIN is currently pursuing the master’s degree with Northwestern Polytechnical University, China. His research interests include ubiquitous computing and mobile crowdsensing.

ZHIWEN YU received the M.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi’an, China, in 2003 and 2005, respectively. He visited the Institute of Information and Communication, Singapore, from 2004 to 2005. From 2006 to 2009, he was a Postdoctoral Researcher at Nagoya University and a Special Researcher at Kyoto University, Japan. From November 2009 to October 2010, he was funded by the German Humboldt Foundation and went to the University of Mannheim, Germany, for collaborative research. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University. His current research interests include pervasive computing, mobile crowdsensing, the Internet of Things, and intelligent information technology.

* * *