TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/14 pp68–80
DOI: 10.26599/TST.2019.9010004
Volume 25, Number 1, February 2020

Spotlight: Hot Target Discovery and Localization with Crowdsourced Photos

Jiaxi Gu, Jiliang Wang*, Lan Zhang, Zhiwen Yu, Xiaozhe Xin, and Yunhao Liu

Abstract: Camera-equipped mobile devices are encouraging people to take more photos and the development and growth of social networks is making it increasingly popular to share photos online. When objects appear in overlapping Fields Of View (FOV), this means that they are drawing much attention and thus indicates their popularity. Successfully discovering and locating these objects can be very useful for many applications, such as criminal investigations, event summaries, and crowdsourcing-based Geographical Information Systems (GIS). Existing methods require either prior knowledge of the environment or intentional photographing. In this paper, we propose a seamless approach called "Spotlight", which performs passive localization using crowdsourced photos. Using a graph-based model, we combine object images across multiple camera views. Within each set of combined object images, a photographing map is built on which object localization is performed using plane geometry. We evaluate the system's localization accuracy using photos taken in various scenarios, with the results showing our approach to be effective for passive object localization and to achieve a high level of accuracy.

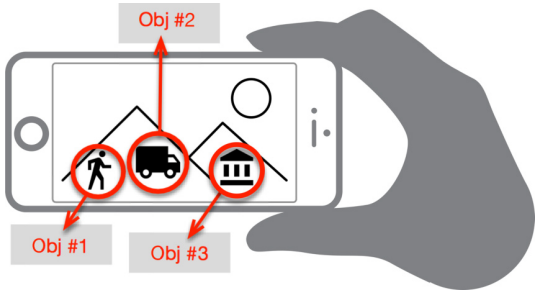
Key words: crowdsourcing; localization; multimedia; mobile computing

1 Introduction

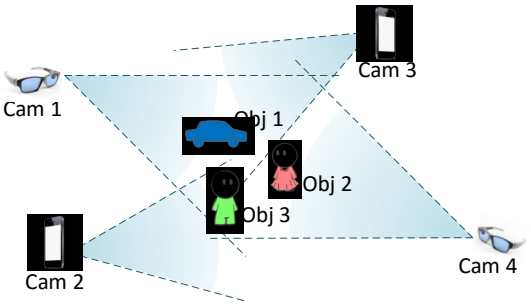
With the popularity of photographing with smart mobile devices, the past few years has seen explosive growth in mobile photo sharing. From Instagram to Snapchat, multimedia-based social networks have attracted much attention[1]. Every second, there are huge numbers of photos and videos shared on the Internet. In the case of popular tourist attractions, there are large numbers of photos and videos captured from different directions and locations. As seen in the motivating example shown as Fig. 1, where multiple photographers have concentrated on the same object, the various cameras have an overlapping Fields Of View (FOV). We call those objects that have attracted much attention "hot targets". Once multiple people have taken photos of them from different directions at the same time, hot targets can be localized using the geographical information provided by the various cameras.

Inspired by this observation, we propose an approach called Spotlight for hot target discovery and passive localization using shared photos. Discovering and then localizing hot targets is of great significance to many applications, such as criminal investigations, event summaries, and crowdsourcing-based Geographical Information Systems (GIS). Additionally, user-generated photos show a crowd's attention and thus provide information useful for abnormal event detection. For example, in tourist areas, there are some objects that are eye-catching but not listed in travel

• Jiaxi Gu and Zhiwen Yu are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: gujiaxi@mail.nwpu.edu.cn; zhiwenyu@nwpu.edu.cn.
• Jiliang Wang and Yunhao Liu are with the School of Software, Tsinghua University, Beijing 100084, China. E-mail: jiliangwang@tsinghua.edu.cn; yunhao@tsinghua.edu.cn.
• Lan Zhang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China. E-mail: zhanglan@ustc.edu.cn.
• Xiaozhe Xin is with Beijing Sogou Technology Development Co., Ltd., Beijing 100084, China. E-mail: 1135532804@qq.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2018-06-07; revised: 2018-07-07; accepted: 2019-01-12



(a) People take photos containing multiple objects.



(b) Cameras of various devices have more or less overlapped FOVs.

Fig. 1 A motivating example of hot target discovery and the passive localization with crowdsourced photos.

brochures; using Spotlight, these popular objects can be discovered and then localized.

Importantly, passive localization using crowdsourced photos is very lightweight and enables the efficient tracking of targets without deploying any extra devices or tags. This localization technology can therefore be categorized as Device-Free Passive Localization (DFPL)[2], as it works without placing devices or tags on targets and without targets needing to actively participate in the localization process. Most DFPL methods depend on wireless technology, such as the Received Signal Strength (RSS)[3–6], Radio-Frequency IDentification (RFID)[7–9], and Channel State Information (CSI)[10,11] methods. Such approaches require either the deployment of a large number of nodes or prior knowledge of the environment. Their performance is also influenced by scenario variability and signal interference. Recently, image-based DFPL methods have been proposed, such as image-assisted indoor mapping[12], building localization using a photo and a 2D map[13], and object localization with photos[14]. However, these methods require either the assistance of extra tools[12,13] or intentional photographing[14], which limits their usefulness.

Considering the problems with existing DFPL methods, we aim to propose a genuinely passive and seamless localization method using crowdsourced photos. The major design objectives are as follows. First, we do not require any prior deployment or intentional photographing; in contrast to existing methods, ours makes few assumptions and requires no prior knowledge. Second, photos are user-generated and crowdsourced. Third, objects to be localized can be discovered according to their visual appearance in photos. Fourth, the popularity of these objects can be revealed by the number of related photos, so that these objects can be further localized on the ground plane.

Crowdsourced photos have normally been taken by various mobile cameras at diverse positions. Identifying objects across this range of different views is a nontrivial task. One approach is to use the appearance information of the objects in the photos; however, one object may look quite different from various viewpoints. The variety in the types of objects and the occlusion between them makes this problem even more challenging. Considering these challenges, we propose our Spotlight method using the overlapping FOVs of crowdsourced photos to localize multiple objects that have drawn a crowd's attention. Spotlight achieves passive localization in three steps. First, we detect objects appearing in photos and calculate the corresponding directions from which they have been photographed. Second, we adopt a robust method based on a K-partite graph model for discovering objects across multiple views; objects appearing in multiple photos are discovered by combining visual appearances and spatial information. Finally, the discovered objects are used to build a photographing map, from which localization is then achieved using plane geometry calculations. We implement Spotlight using photos taken in various scenarios and evaluate its performance extensively. It shows a high level of robustness owing to crowdsourcing and achieves a high level of accuracy. The major contributions of this work are as follows.

(1) We propose Spotlight to achieve passive localization using crowdsourced photos without any prior knowledge or intentional deployment.

(2) We design a photo fusion method to derive object location by combining object images across multiple views.

(3) We implement Spotlight on mobile phones and evaluate its performance using photos taken in various scenarios. The results show that it achieves a high level of accuracy.

The rest of this paper is organized as follows.

Section 2 formally defines the problem. Section 3 introduces the workflow of Spotlight, divided into three main phases. In Section 4, we define the problem and propose a graph-based model for combining object images across multiple camera views. In Section 5, we describe the building up of a photographing map and the process of passive localization. In Section 6, we discuss the implementation details of Spotlight and examine its performance. Section 7 covers the related work and Section 8 concludes the paper.

2 Problem Definition

2.1 Object detection

The first step of object localization is to determine all of the candidate objects. Our goal is to detect as many objects as possible, such that they can be further analyzed with the help of other clues. At this stage, the only basis we have to rely on is the visual information from photos. Thus, this is a typical object detection problem of the kind that has been studied for decades. While a great deal of methods have been proposed over this time in the field of computer vision, there still remain some challenges and open issues in this field. The accuracy of detection is influenced to different degrees by various factors, such as object occlusion, image resolution, and background complexity. There is also a need to trade some performance for efficiency when there are a large number of photos to process.

2.2 Object azimuth

Object azimuth is the direction from the camera taking a photo to an object's physical entity. The azimuth value is the degrees from north on the ground plane. This information is difficult to obtain from visual information only. Fortunately, smart devices can record GPS and mobile sensor data while taking photographs. The location and orientation of the camera with which the photo is taken with can be computed using GPS in combination with magnetometer sensor data. However, objects appearing in a same photo may have different azimuths because of their different positions in the photo. As a result, the camera's photographing direction cannot represent the azimuth of all objects appearing in a photo. Calculating object azimuths is another challenging problem, as they are influenced by many factors such as photographing distance, device parameters, and camera postures. The azimuth is critical to the final localization accuracy and a small error in the azimuth is likely to lead to a very poor localization result, especially in large-scale scenarios.

2.3 Object images and entities

We use the term "object image" to represent a sub-image extracted from the original photo according to the bounding box of object detection, and "object entity" to represent an individual object in its physical sense. Obviously, each object image corresponds to one unique object entity, whereas one object entity can have multiple object images representing it. Localization makes sense only for object entities. The crowdsourced data from multiple cameras need to be used collaboratively, so we combine object images to distinguish them according to the different object entities they correspond to. A certain object, however, may have a different appearance across different camera views. One of the challenges in combining object images is that visual appearance is influenced by many factors, such as lightness, shadows, and saturation. A further challenge arises when there are some objects with a similar appearance, as is very common in photographs of sporting events in which players of the same team are dressed similarly. As a matter of fact, there are many types of information extracted from photos, and visual appearance is just one of the features. Given the challenges involved, there is potential for improvement in object image combination that can be realized by adopting another feature in addition to visual appearance.

3 System Overview

Figure 2 shows the workflow of Spotlight. There are three main phases involved between inputting sensor-rich photos to outputting location results. Phase 1 takes raw crowdsourced photos as input, detects objects within photos, and calculates their azimuths by combining the multiple cameras' sensor data. In Phase 2, we use the object images generated in the first phase to construct a graph model for combining object images. Based on this model, multiple object cliques are computed, each containing object images corresponding to the same object entity. In Phase 3, we build a photographing map for each object clique and conduct localization on the ground plane.

3.1 Object detection and azimuth calculation

In Phase 1, we extract object-related information from photos. For object detection, we adopt Faster R-CNN[15] to strike the right balance between efficiency and accuracy. We use a pre-trained model that can be used

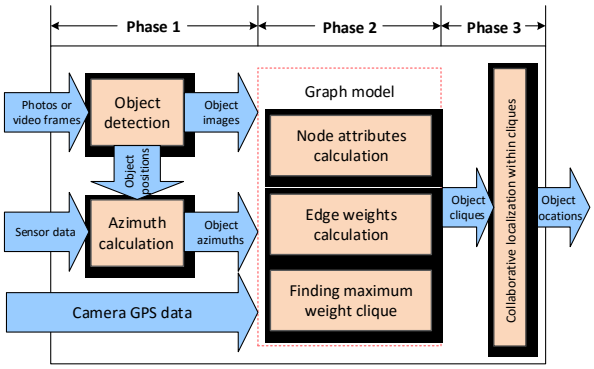


Fig. 2 Spotlight consists of three main phases responsible for object detection, object image combination, and collaborative localization.

for detecting common objects (person, bicycle, car, etc.) This model is employed to extract all of the objects within each photo or video frame. For calculating the object azimuth, we indirectly leverage photo content and camera posture to derive the photographing direction. The azimuth can be further computed by combining sensor data and the positions of objects in the photos. We index photos as $\{1, 2, \ldots, K\}$. For each photo k, there are an uncertain number of objects, so we denote them as a set $V_k = \{1, 2, \ldots\}$ $(1 \leqslant k \leqslant K)$. The j-th object in the k-th photo can be denoted as $v_{k,j}$ $(1 \leqslant k \leqslant K, j \in V_k)$. We then use the camera imaging model to calculate the azimuth for each object in the photos, as illustrated in Fig. 3.

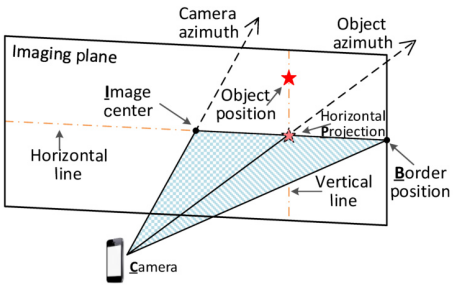
Figure 3a shows the imaging model from the camera. We regard a photo as an imaging plane. The red star represents the position of an object, which we project to the horizon. Figure 3b is the top-down view of the imaging model. The direction from the camera to the image center shows the camera's azimuth. For a given camera view, there are three fixed positions: C is the camera position, I is the image center, and B is the border position. The viewing angle of a camera can also be obtained from device parameters; $\angle ICB$ denoted by θ is equal to half the viewing angle. A given object in the photo is projected to the horizon and its horizontal projection is denoted as P. The declination angle of an object is denoted as α; it turns negative when P is on the left half of the frame. From this we arrive at the following equation according to plane geometry,

$$\alpha = \arctan\left(\frac{\overline{IP}}{\overline{IB}}\tan\theta\right).$$

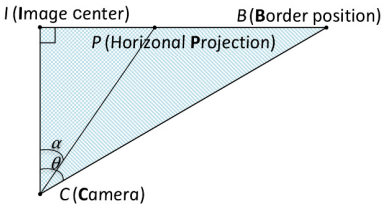
Denoting camera azimuth as a_{cam} and object azimuth as a_{obj}, we then have

$$a_{\text{obj}} = a_{\text{cam}} + \alpha = a_{\text{cam}} + \arctan\left(\frac{\overline{IP}}{\overline{IB}}\tan\theta\right)$$

Finally, as $\overline{IP}$, $\overline{IB}$, and θ can be obtained from the photos, we can obtain the object azimuth a_{obj}.



(a) Each object is projected to horizon for the calculation of its azimuth.



(b) Top view of the imaging model after projecting object position on horizon.

Fig. 3 Object azimuth is calculated by projecting object position to horizon of the photo so that there is an offset to the picture center.

3.2 Object entity discovery

In Phase 2, we use the object images to discover object entities. A single object image without depth of field is insufficient to describe the location of an object. For an object appearing in multiple photos, we can obtain a number of corresponding object images. However, it is hard to determine what object entity an object image corresponds to. An object entity may show itself with more or less different colors or shapes when photographed from diverse directions and/or at multiple distances. Object image combination is critical since the views of a single object from multiple cameras overlap. We can derive the identity of an object entity collaboratively using potentially related object images. Furthermore, the location and direction of photographing actually provide extra evidence for combining object images. Once the combination problem is solved, an object entity can be described by its combined object images. Also, we can find out the popularity of an object entity based on the number of images in which it appears.

3.3 Object entity localization

The goal of Phase 3 is to compute the location of each object entity using the information from its

corresponding object images. Our focus in this phase is not on the visual content of an image but rather on where an image was taken and its photographing direction. Within a set of object images, we can compute the camera's GPS and object's azimuth, and then build a photographing map. The location of the object entity on the ground plane can be gradually narrowed down using the combined object images. Thus, the greater the number of images representing an object entity, the higher will be the accuracy of the final localization result.

4 Object Image Combination

Figure 4 illustrates the object image combination problem. The tiles of various colors and shapes represent object images extracted from original photos after object detection. A specific object entity has different visual appearances across different views. The goal of this phase is to find the correct combinations of these object images. Object images that correspond to the same object entity should be gathered together, as illustrated on the right side of the figure.

Combining object images across multiple views is a very important step for object tracking or localization. Most existing works formalize this as a path search problem[16, 17]. Those methods work when there are only small numbers of cameras and objects, but they are not feasible as solutions to our problem because they are too aggressive in combining similar object images and will give a high rate of false positive errors. An aggressive strategy is inappropriate for our problem because an incorrectly combined object image (marked in Fig. 4 as a dashed red circle) produces a significant error in localization. In contrast, an incomplete combined object image (marked in Fig. 4 as a dashed blue circle) has less negative impact. When the number of object images is large, the absence of a single object image will make little difference to the final localization accuracy. To serve our need for a stricter combining strategy, we propose a graph-based model, in which vertices represent object images and edges indicate combining possibilities. Based on this model, we formalize object image combination as a maximum weight clique search problem.

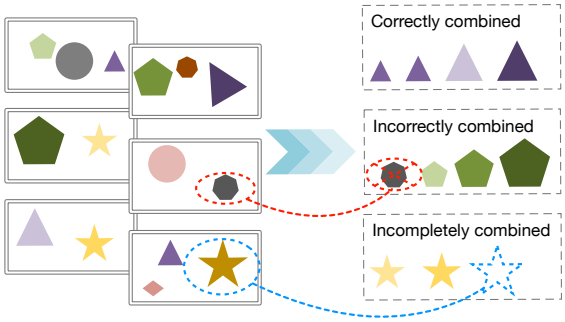


Fig. 4 Object images are combined according to their visual appearances. Incompletely combined ones are better than incorrectly combined ones.

4.1 Definition of object image

In addition to spatial information, including GPS and object azimuth, we introduce image features for use in distinguishing object images. The color histogram is adopted because it is a widely used image feature to solve combination problems. In comparison with other features, using the color histogram shows much more robust performance in the face of variations in posture and viewpoint. To calculate the color histogram, we adopt the HSV color space[18] and Ref. [19], showing that HSV can greatly reduce the color variation across camera views by separating the lightness component. Given an object image v, we denote its color histogram as m_v, the GPS of its corresponding camera as c_v, and the object azimuth as a_v. Thus, an object image v can be represented by (m_v, c_v, a_v).

4.2 Model for object image combination

In order to achieve localization, object images need to be mapped to their corresponding object entities. In formalizing this as a clustering problem involving the attributes of object images, we make the following three observations:

(1) There is zero probability that object images appearing in the same photo correspond to the same object entity;

(2) There is a low probability that object images whose photographing paths have no intersection point correspond to the same object entity; and

(3) Visually similar object images are more likely to correspond to the same object entity.

Based on the above observations, we formalize the structure of object images as a graph G. Given K photos in total, vertex $v_{k,j}$ represents the j-th object image in the k-th photo, and edge weight $w(u, v)$ indicates the similarity between object images, u and v. Table 1 shows the notations and their corresponding definitions. Based on the first observation, there are no edges between vertices in the same photo. Therefore, this graph further evolves to a K-partite graph where K

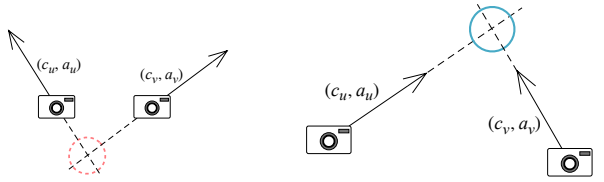
Table 1 Notations and definitions.

Notation	Definition
K	Total number of photos
V_k	Set of object images in photo k, $1 \leqslant k \leqslant K$
$v_{k,j}$	The j-th object image in k-th photo
V	Set of all object images, $V = \{v_{k,j} \mid 1 \leqslant k \leqslant K, 1 \leqslant j \leqslant \lvert V_k \rvert\}$
m_v	Visual feature of object image, $v \in V$
c_v	Camera GPS corresponding to object image, $v \in V$
a_v	Azimuth of object image, $v \in V$
$w(u, v)$	Similarity between object images u and v, $u, v \in V$

represents the total number of photos.

Edge weight, representing the similarity between object images is defined based on both spatial information and visual features. To begin, we consider cases of spatial information. In the definition of an object image (m_v, c_v, a_v), c_v and a_v represent the camera GPS and azimuth, respectively. Our second observation points out that it is unlikely for two object images with no intersection point in their photographing paths to correspond to the same object entity. Therefore, given two vertices u and v in the graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. We determine their combination firstly based on their spatial information. As Fig. 5 shows, we can determine whether they have intersection points on the ground plane using GPS and azimuth information. A pair of (c, a) represents the camera GPS and the object's azimuth from the camera, so it can be regarded as a ray on the ground plane.

The edge between u and v exists, that is $(u, v) \in E$, only if there is an intersection point calculated with (c_u, a_u) and (c_v, a_v). Edge weight is determined by some specific visual feature of object images, i.e., m_u and m_v. Since we apply the color histogram as a visual feature, image similarity can be represented by the Bhattacharyya distance. The function for calculating



(a) In this case, these two object images are unlikely to have any intersection due to diverging photographing directions.

(b) There may be an intersection point in this case though they may correspond to different object entities eventually.

Fig. 5 Camera location and object azimuth are used for determining whether two object images are likely to have any intersection on the ground plane.

Bhattacharyya distance is denoted as $B(\cdot)$. For consistency, we assign the reciprocal of $B(\cdot)$ to edge weight $w(u, v)$ in

$$w(u, v) = B^{-1}(m_u, m_v) \tag{1}$$

therefore, a high weight indicates a strong combination.

Based on this, edges with low weights are removed. We set a threshold ϵ and remove the edges which meet $w(u, v) < \epsilon$; that is, $E \backslash \{(u, v) | w(u, v) < \epsilon\}$.

4.3 Maximum weight clique problem

Our goal is to place all of the detected object images into different clusters according to the object entity each of them corresponds to; the basis of such clustering is the similarity between object images. Therefore, we formalize our clustering as a typical clique problem, the statement of which can be presented as follows.

Problem 1 Given an undirected K-partite edge-weighted graph $G = (V, E)$, iteratively find all the maximum weight cliques such that the cardinality of each clique is larger than two.

Two vertices are said to be adjacent if they are connected with an edge. A clique q of $G = (V, E)$ is a subset of V so that each pair of vertices in q is mutually adjacent (i.e., a complete subgraph). The cardinality of a clique represents the number of vertices it contains. The Maximum Clique Problem (MCP) is to find a clique with the maximum number of vertices, equating to maximum cardinality. For devising a weighted-graph model for our work, we formalize the problem as a Maximum Weight Clique Problem (MWCP), which is a generalization of an MCP. The problem is to find a clique with the highest total weight. An MCP is a special case in which all the edge weights are equal. In our problem, the goal is to find the maximum weight cliques iteratively. After each iteration, the vertices and edges that have been previously worked out are removed; hence, the scale of the problem is gradually decreased. In practice, maximum weight cliques which have only two vertices are useless, since localization is unreliable with just two images. The termination condition of our method is based on this fact—we define a triplet of vertices as three connected vertices in a graph, and the iteration stops when there are no triplets of vertices in the graph.

One particular feature of the graph model we propose in this paper is worth noting. The graph is K-partite originally, so it can be colored with K colors. In other words, vertices in the graph can be partitioned into K

partitions with no adjacent vertices in each partition. It was pointed out in Ref. [20] that coloring can bring a considerable reduction of computation in an MCP. For a given K-partite graph, at most one vertex in each tier can be chosen to constitute a clique. This permits the solution space to be reduced dramatically. For efficiency, we adopt a branch-and-bound method to find the optimal solution for the MWCP in every iteration. We define the index of the tier only containing root as 0. Children of each vertex represent vertex choices in the graph. Since at most one vertex will be chosen in each tier, the total number of children for any vertex in tier k is $|V_{k+1}| + 1$, including an empty choice for where no vertex is chosen in the tier. By definition, each leaf vertex is a potential solution. By comparing the computed weights of cliques, we find the solution of the MWCP in the current graph. Furthermore, we cut those sub-trees based on their potential maximum weight in comparison with the current optimal solution.

The algorithm for the branch-and-bound method is presented in Algorithm 1. We use a stack to store live vertices, each of which is expanded with its children as branches. The bounding strategy is based on two requirements. First, the newly added vertex has to be fully connected with the current clique. Second, the maximum weight of potential solutions in the sub-trees should be more than the weight of current solution; otherwise, the children are not pushed into the stack of live vertices.

In each iteration, we conduct this algorithm to obtain a clique, then we remove the clique from the graph. The iterations stop when there are no triplets of vertices in the graph. In this way, we find all the maximum weight cliques that meet our requirements.

Algorithm 1 Branch-and-bound on MWCP

Input: $q^{\text{opt}} \leftarrow \varnothing$ // Potential max weight clique
Output: $S^{LN} \leftarrow \{n^{\text{root}}\}$ // Stack of live vertices
for while $S^{LN} \neq \varnothing$ **do**
  $v \leftarrow S^{LN}.pop()$
  if v is a leaf vertex **then**
    // Compute clique weight about v
    if $Weight(v) > Weight(q^{\text{opt}})$ **then**
      $q^{\text{opt}} \leftarrow v$
    end if
  end if
  // Generate all the children of v
  for each child u in $CHI(v)$ **do**
    // Whether it is a live vertex
    if complete graph ***OR*** potential optimal **then**
      $S^{LN}.push(u)$
    end if
  end for
end for

5 Collaborative Localization

After combining object images, we place them into cliques each of which corresponds to an object entity. For passive localization of a specific object entity, we focus on object images in the same clique. A photographing map is built on the ground plane based on camera GPS and object image azimuth as illustrated in Fig. 6. As proposed above, each object image can be regarded as a ray described by its corresponding camera GPS and object azimuth. Thus, object image v can be denoted as $b_v = (c_v, a_v)$ after removing its visual feature m_v. In Fig. 6, black dots with rays represent these object images. The position of a dot is determined by c_v, while the direction of its ray is determined by a_v. Thus, for every two object images, a potential intersection point can be calculated as

$$r_{u,v} = IN(b_u, b_v) \tag{2}$$

in which $r_{u,v}$ is a location representing the intersection point of b_u and b_v, and $IN(\cdot)$ is a function for computing the potential intersection point of two rays. Obviously, two rays may have no intersection points, in which case, no result is desired or returned. After working out all of the potential intersection points on the photographing map, we denote a set of intersection points as **R**, which can be treated as a cluster. Within **R**, we use the centroid of the cluster as the final location of the corresponding object entity.

Different degrees of error arise from poor GPS

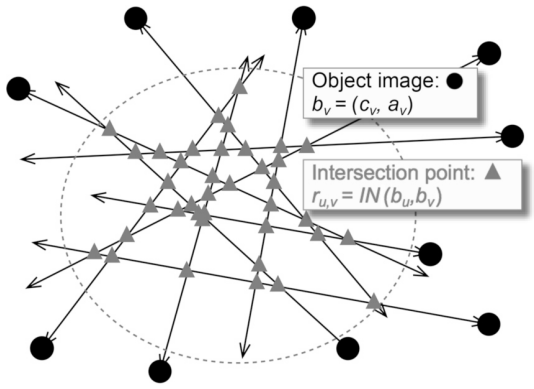


Fig. 6 Intersection points are generated based on GPS and azimuth information of object images in the same clique. Furthermore, they are used for computing the object entity locations.

signal reception, photographing jitter, geomagnetic disturbance, etc. However, the influence of such errors can be decreased by taking advantage of crowdsourcing. As long as there are enough photos covering the object, and therefore the maximum weight clique contains a high enough number of vertices, the final error rate of Spotlight can be controlled to fall into an acceptable range.

6 Implementation and Evaluation

6.1 Experimental setup

Spotlight relies on both multimedia data and sensor data of mobile photos. Instead of directly using the native camera application in mobile phones, we develop a mobile App which can record real-time sensor data while taking photos and videos. Considering the diversity of mobile phones, a range of phones made by various manufacturers are used to take photos and videos for experimental purposes. The standard Android API is used for building the mobile App for our system. The resolution of photos and videos varies from 640×520 to 1920×1080.

The Spotlight process, including object detection runs offline on a Windows PC with a GeForce GTX TITAN X. The processing time varies with the number of photos. We find that the object detection process accounts for a great proportion of overall processing time, even though Faster R-CNN is an outstanding method for object detection. The overall execution time can be reduced if object detection can be achieved in a reliable and effective way.

In order to evaluate the influences of the surrounding environment, we conduct our experiments in various conditions, considering weather, lightness, background, and device models. On the other hand, there are indeed various types of objects in real environment, but object detection of various objects is not the main focus of our work. As a result, we recruit volunteers as representative "objects". Human bodies are typically non-rigid objects and detecting them is challenging for computer vision.

6.2 Ground truth

For every location of either camera or object entity, we use Google Earth for calibration and record the coordinates as ground truth. We also update the location information in real time as we predefine the roadmap. We use GeographicLib[21] for distance and azimuth calculation. Therefore, we can actually calculate the location of all the points and the location relation among them. The latitude and longitude are both in WGS84 geodetic coordinates. Object entities are pre-defined during our experiments, and the movement traces of our object entities are regulated.

6.3 Azimuth accuracy

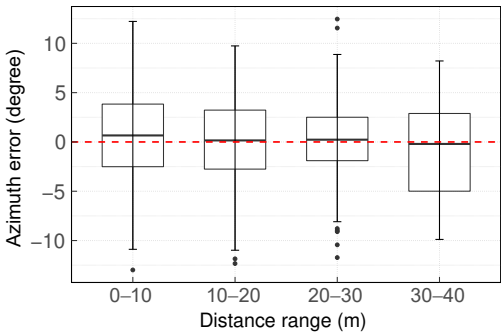
The GPS accuracy of mobile phones is beyond our control, so we use the readings of the GPS module on the mobile phone directly as the location of the camera. The average GPS error in our experiments is around 7.327 meters, the minimum and maximum are 2.432 meters and 13.208 meters, respectively.

As for the object azimuth calculation, we consider three components. The first is the z-axis azimuth of the mobile phone, which can be directly obtained using the Android API drawing on the accelerometer and magnetometer. The second component is the magnetic declination, which represents the horizontal component of the magnetic field from true north and varies both from place to place and with the passage of time. We obtain its value, which is around $-5.5566°$ in our experiments, along with real-time GPS readings via built-in sensors. The final component for azimuth calculation is the position of the object in the photos, as explained above.

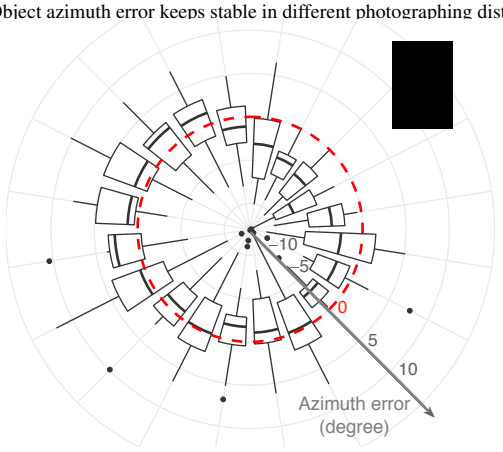
Figure 7 shows the object azimuth error for different distances and devices. To evaluate the calculation of the object azimuth, we take photos and videos using the same phone at different distances from target objects. Figure 7a shows that the azimuth error remains within 10° in most cases, indicating that photographing distance has little influence on azimuth calculation. To evaluate the influence of different devices, we ask our volunteers to stand around the target object and take photos and videos using different phones. Figure 7b shows the results of this evaluation, with the red dashed circle representing zero error. Again, we find that the azimuth error stays within 10°.

6.4 Performance of combining object images

Figure 8 shows the visual appearances of some of the objects in our experiments. As Fig. 8a shows, a different number of objects are detected in each photo. Each object image represents one view of an object entity. Some objects may appear in only a single camera's FOV; we do not consider these cases because it is both unnecessary and infeasible to achieve accurate localization on such "unpopular" objects. Instead, we focus on those objects appearing in as many photos as



(a) Object azimuth error keeps stable in different photographing distances.



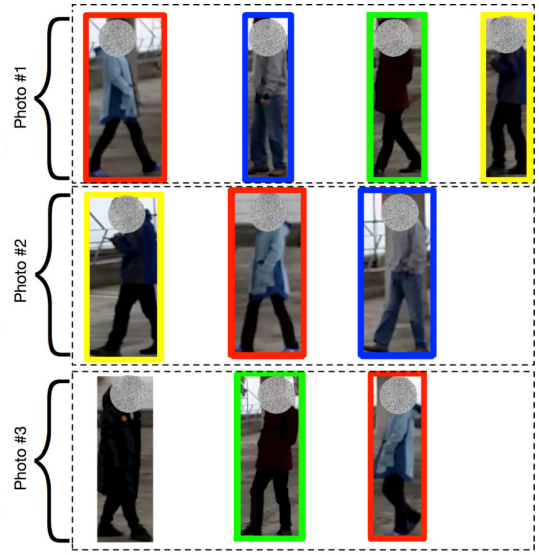
(b) Object azimuth error is acceptable when using various devices in different directions.

Fig. 7 Object azimuth error.

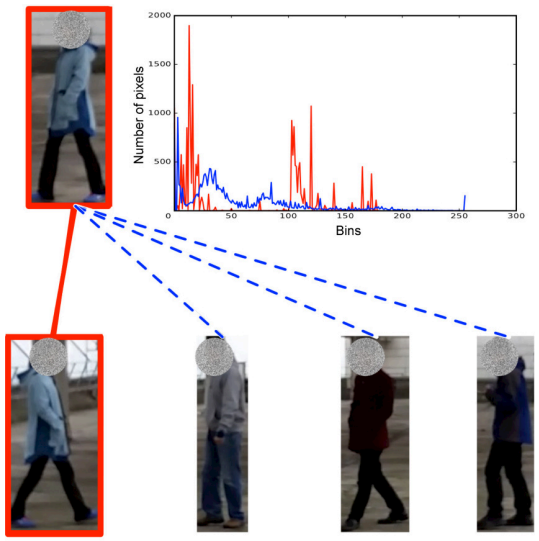
possible. As proposed in Section 4, in addition to spatial information, we combine object images across multiple camera views using their visual features. In Fig. 8b, for example, each object image is used for comparison with all the object images in another photo. We use the color histogram of hue and saturation as a visual feature to identify objects, as shown in the line chart in Fig. 8b. The blue dashed lines between object images represent a weak combination while the red line indicates a strong combination.

We evaluate the accuracy of object detection in five different scenarios. For each scenario, we choose one target object. For video quality, the resolution is 1920×1080 and the framerate is 30 fps. The total length of video for each scenario is around 100 seconds. The results are shown in Fig. 9. Object detection accuracy is quite high in all of these scenarios; it reaches the highest level of accuracy in the fifth scenario because the view of the rooftop is good and the background is clear.

In our proposed graph model, vertices represent object images and edge weights represent similarities between object images. In order to calculate similarities, we remove the edges between those object images that are unlikely to have intersection



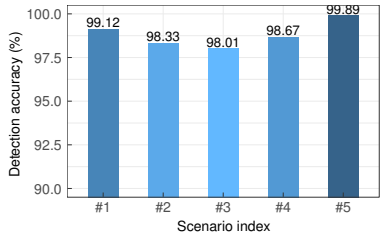
(a) Multiple objects are detected in every photo and object entity shows different visual appearances across different views.



(b) Color histogram of hue and saturation is a determining characteristic for combining object images.

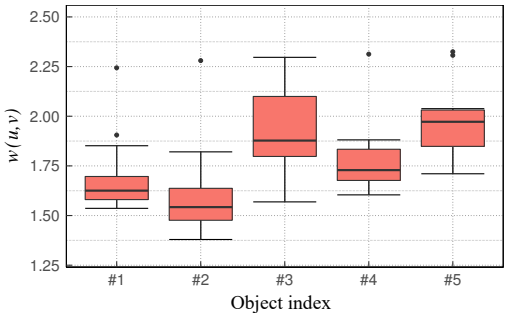
Fig. 8 Object image combination is determined by their similarities in visual appearances.

points on the ground plane. We adopt the reciprocal of Bhattacharyya distance to quantify the similarity which is the edge weight in the graph ($w(u, v)$ in Eq. (1)). We conduct experiments on five different objects, with the results shown in Fig. 10a. We find that the results are basically stable. In order to compute an appropriate threshold, five different objects are used as test objects. For each object, we extract 30 object images from photos taken in different conditions. Figure 10b shows the error rates of object image combination using different thresholds. We adopt $\epsilon = 1.5$ as an appropriate threshold so that we can

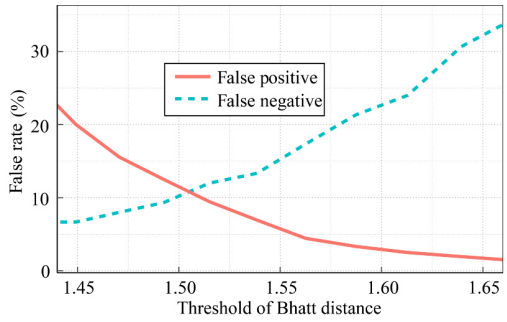


Scenario	Description
#1	Football field on a sunny day
#2	Building site on a cloudy day
#3	City street on a cloudy day
#4	Country field on a rainy day
#5	Rooftop on a cloudy day

Fig. 9 Objection detection accuracy in various scenarios.



(a) For different object entities, $w(u,v)$ between their corresponding object images is calculated.



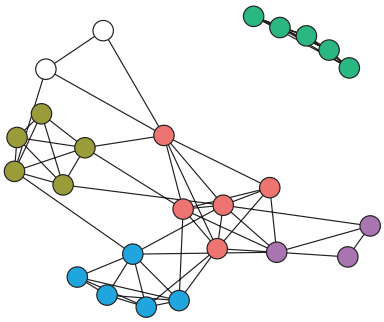
(b) Error rates of object image combination with different thresholds.

Fig. 10 Performance of the object image combination.

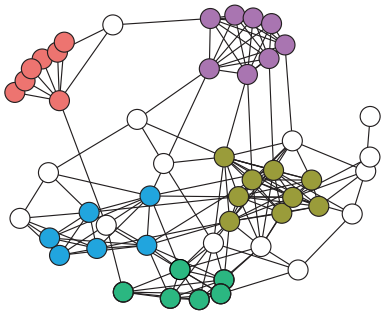
reach almost 10% in both false positive rate and false negative rate. Moreover, according to our algorithm, we construct a K-partite graph with different values of K, where K represents the number of photos covering an overlapping FOV. Figure 11 shows our fully constructed graph model.

6.5 Localization performance

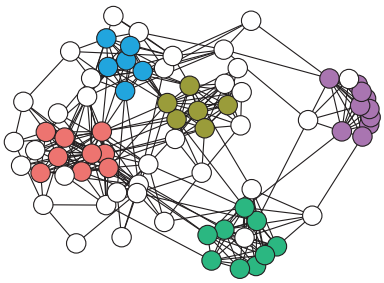
As mentioned above, the performance of Spotlight relies on crowdsourcing, and the final results are greatly influenced by the number of valid photos which have



(a) $K = 5$



(b) $K = 10$



(c) $K = 15$

Fig. 11 Generated graphs using different numbers of valid photos.

a partially or fully overlapping FOV. Hence, to fully evaluate the performance of localization, we conduct experiments with different numbers of valid photos with overlapping FOV. Some invalid photos are also present in our experiments. Figure 12 is a Cumulative Distribution Function (CDF) figure showing the results with 5, 10, and 15 valid photos. When there are 5 valid photos, the localization error of Spotlight is within 15 meters; the error is reduced to 5 meters when there are 15 valid photos.

7 Related Work

Object image combination. This is a significant yet challenging problem in object tracking and re-identification. Similarities between object images have been sought from various perspectives, including position[22], appearance[23], and motion[24]. Researchers have also considered a fusion of multiple features,

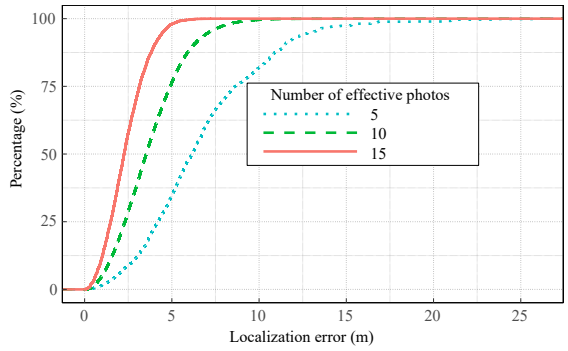


Fig. 12 CDF figure of localization error with different numbers of valid photos.

including position, speed, shape, and chromatic characteristics[25]. Hamid et al.[17] used static cameras at fixed positions for football player localization; they used a complete K-partite graph and solved the combination problem by finding minimum weight K-length cycles.

Hot target discovery. Salient object detection within images, a special case of hot target discovery in spatial scenarios, has been researched for a number of years in computer vision studies[26–29]. Going beyond these purely vision-based methods, Peng et al.[30] proposed a "Spatial+Visual" framework which combined spatial information with visual information to find places-of-interest in photos. Moreover, they explored the adoption of a deep neural network[31] and Bayesian method[32] in their system to improve the spatial recognition accuracy.

Device-free passive localization. Wireless technology inspires many DFPL methods based on various characteristics of signals, such as RSS[3–6, 33] and CSI[10, 11]. RFID is another similar technology used for DFPL[7–9]. Image-based methods have advantages because they are lightweight and flexible. For example, CamLoc[34] enables immediate object localization with only two photos; although the accuracy is high, it has the limitation that those two photos have to be taken at a fixed location. Hotspotting[35] is another image-based localization system; it turns localization to a crowdsourced image annotation task and needs more than a few participants to answer various questions.

8 Conclusion

Crowdsourcing has instinctive advantages in reflecting a crowd's attention. Moreover, with the popularity of mobile photo sharing, objects drawing people's attention can be further localized. In this paper, we propose Spotlight, a novel system for discovering and localizing multiple objects using crowdsourced photos. It requires no intentional deployment or prior knowledge of the environment. The target objects are seamlessly discovered by combining their corresponding images across multiple camera views; these objects can be then further localized passively. Extensive experiments using photos taken in various scenarios show Spotlight to be effective in object discovery and to obtain a high localization accuracy. Spotlight has various potential applications in criminal investigations, event summaries, crowdsourcing-based GIS, etc.

References

[1] M. Duggan, Photo and video sharing grow online, https://www.pewinternet.org/2013/10/28/photo-and-video-sharing-grow-online/, 2013.

[2] M. Youssef, M. Mah, and A. Agrawala, Challenges: Device-free passive localization for wireless environments, in *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking*, Montréal, Canada, 2007, pp. 222–229.

[3] X. Zheng, J. Yang, Y. Chen, and Y. Gan, Adaptive device-free passive localization coping with dynamic target speed, in *Proceedings of the 32nd International Conference on Computer Communication*, Turin, Italy, 2013, pp. 485–489.

[4] H. Aly and M. Youssef, New insights into wifi-based device-free localization, in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Zurich, Switzerland, 2013, pp. 541–548.

[5] J. Wang, D. Fang, X. Chen, Z. Yang, T. Xing, and L. Cai, LCS: Compressive sensing based device-free localization for multiple targets in sensor networks, in *Proceedings of the 32nd International Conference on Computer Communication*, Turin, Italy, 2013, pp. 145–149.

[6] C. Liu, D. Fang, Z. Yang, H. Jiang, X. Chen, W. Wang, T. Xing, and L. Cai, RSS distribution-based passive localization and its application in sensor networks, *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2883–2895, 2016.

[7] L. M. Ni, D. Zhang, and M. R. Souryal, RFID-based localization and tracking technologies, *IEEE Wireless Communications*, vol. 18, no. 2, pp. 45–51, 2011.

[8] P. Yang, W. Wu, M. Moniri, and C. C. Chibelushi, Efficient object localization using sparsely distributed passive RFID tags, *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5914–5924, 2013.

[9] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, W. Xi, Z. Jiang, and Z. Wang, Twins: Device-free object tracking using passive tags, *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1605–1617, 2016.

[10] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, Fila: Fine-grained indoor localization, in *Proceedings of the 31st*

International Conference on Computer Communication, Orlando, FL, USA, 2012, pp. 2210–2218.

[11] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, Pilot: Passive device-free indoor localization using channel state information, in *Proceedings of the 33rd International Conference on Distributed Computing Systems*, Philadelphia, PA, USA, 2013, pp. 236–245.

[12] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, Enhancing wifi-based localization with visual clues, in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Osaka, Japan, 2015, pp. 963–974.

[13] X. Xiong, Z. Yang, L. Shangguan, Y. Fei, M. Stojmenovic, and Y. Liu, Smartguide: Towards single-image building localization with smartphone, in *Proceedings of the 16th International Symposium on Mobile Ad Hoc Networking and Computing*, Hangzhou, China, 2015, pp. 117–126.

[14] L. Shangguan, Z. Zhou, Z. Yang, K. Liu, Z. Li, X. Zhao, and Y. Liu, Towards accurate object localization with smartphones, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2731–2742, 2014.

[15] S. Ren, K. He, R. B. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[16] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke, Tracking a large number of objects from multiple views, in *Proceedings of the IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, 2009, pp. 1546–1553.

[17] R. Hamid, R. Kumar, J. Hodgins, and I. Essa, A visualization framework for team sports captured using multiple static cameras, *Computer Vision and Image Understanding*, vol. 118, pp. 171–183, 2014.

[18] E. D. Cheng and M. Piccardi, Matching of objects moving across disjoint cameras, in *Proceedings of the International Conference on Image Processing*, Atlanta, GA, USA, 2006, pp. 1769–1772.

[19] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. H. Tu, Shape and appearance context modeling, in *Proceedings of the 11th International Conference on Computer Vision*, Janeiro, Brazil, 2007, pp. 1–8.

[20] P. R. Östergård, A fast algorithm for the maximum clique problem, *Discrete Applied Mathematics*, vol. 120, nos. 1–3, pp. 197–207, 2002.

[21] C. F. Karney, Algorithms for geodesics, *Journal of Geodesy*, vol. 87, no. 1, pp. 43–55, 2013.

[22] K. Kim and L. S. Davis, Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering, in *Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, 2006, pp. 98–109.

[23] X. Dai and S. Payandeh, Tracked object association in multi-camera surveillance network, in *Proceedings of the International Conference on Systems, Man, and Cybernetics*, Manchester, UK, 2013, pp. 4248–4253.

[24] Q. Zhai, S. Ding, X. Li, F. Yang, J. Teng, J. Zhu, D. Xuan, Y. F. Zheng, and W. Zhao, VM-tracking: Visual-motion sensing integration for real-time human tracking, in *Proceedings of the 34th International Conference on Computer Communication*, Hong Kong, China, 2015, pp. 711–719.

[25] S. Piva, A. Calbi, D. Angiati, and C. S. Regazzoni, A multi-feature object association framework for overlapped field of view multi-camera video surveillance systems, in *Proceedings of the International Conference on Advanced Video and Signal-based Surveillance*, Como, Italy, 2005, pp. 505–510.

[26] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, Learning to detect a salient object, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[27] X. Shen and Y. Wu, A unified approach to salient object detection via low rank matrix recovery, in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 853–860.

[28] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, Unsupervised joint object discovery and segmentation in internet images, in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 1939–1946.

[29] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, Salient object detection: A benchmark, *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[30] P. Peng, L. Shou, K. Chen, G. Chen, and S. Wu, The knowing camera 2: Recognizing and annotating places-of-interest in smartphone photos, in *Proceedings of the 37th International Conference on Research and Development in Information Retrieval*, Gold Coast, Australia, 2014, pp. 707–716.

[31] P. Peng, H. Chen, L. Shou, K. Chen, G. Chen, and C. Xu, DeepCamera: A unified framework for recognizing places-of-interest based on deep convnets, in *Proceedings of the 24th International Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 1891–1894.

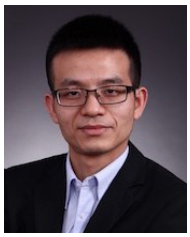
[32] P. Peng, L. Shou, K. Chen, G. Chen, and S. Wu, KISS: Knowing camera prototype system for recognizing and annotating places-of-interest, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 994–1006, 2016.

[33] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, Nuzzer: A large-scale device-free passive localization system for wireless environments, *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, 2013.

[34] S. J. Hwang and K. Grauman, Reading between the lines: Object localization using implicit cues from image tags, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1145–1158, 2012.

[35] M. Salek, Y. Bachrach, and P. Key, Hotspotting—A probabilistic graphical model for image object localization through crowdsourcing, in *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Bellevue, WA, USA, 2013. pp. 1156–1162.

Jiaxi Gu is currently a PhD student in Northwestern Polytechnical University, China. He received the BE degree from Northwestern Polytechnical University. His research interests include mobile computing and privacy-aware video streaming.

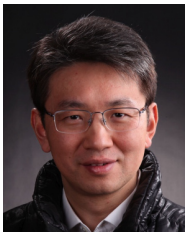


Jiliang Wang is currently an associate professor in School of Software and Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, China. He received the BE degree from University of Science and Technology of China and the PhD degree from Hong Kong University of Science and Technology, respectively. His research interests include wireless and sensor networks, Internet of Things, and mobile computing. He is a member of the IEEE.

Lan Zhang received the bachelor degree from Tsinghua University, China, in 2007, and the PhD degree from Tsinghua University, China, in 2014. She is currently a distinguished researcher at the School of Computer Science and Technology, University of Science and Technology of China. Her research interests include privacy protection, secure multi-party computation, and mobile computing.

Zhiwen Yu received the PhD degree from Northwestern Polytechnical University, Xi'an, China. He is currently a professor and vice-dean with the School of Computer Science, Northwestern Polytechnical University. He had worked as an Alexander Von Humboldt fellow with Mannheim University, Germany, from November 2009 to October 2010, and a research fellow with Kyoto University, Japan, from February 2007 to January 2009. His research interests include pervasive computing and human-computer interaction.

Xiaozhe Xin works at Beijing Sogou Technology Development Co., Ltd, as a researcher. He received the bachelor degree from Tianjin University of Science and Technology in 2014, and the master degree from Tsinghua University in 2017. His research interests include image process and deep learning.



Yunhao Liu received the BS degree from Tsinghua University, China, in 1995, and the MS and PhD degrees from Michigan State University in 2003 and 2004, respectively. He is now a Changjiang professor and dean of School of Software, Tsinghua University, China. His research interests include sensor network and pervasive computing, peer-to-peer computing, the Internet of things (IOT), and supply chain. He is a fellow of the IEEE and ACM.