IEEE Open Journal of
Vehicular Technology

Received 8 August 2022; accepted 1 September 2022. Date of publication 27 October 2022;
date of current version 31 October 2022. The review of this article was coordinated by Editor Koji Yamamoto.

Digital Object Identifier 10.1109/OJVT.2022.3205422

Hybrid Data Selection With Context and Content Features for Visual Crowdsensing

WEI SONG (Senior Member, IEEE)

Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada

This work was supported in part by New Brunswick Innovation Foundation (NBIF) under Grant RAI-2019-033 and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RGPIN-2015-04641.

ABSTRACT Visual crowdsensing (VCS) is becoming predominant in mobile crowdsensing, but there still exist various unique challenges, including large sizes of visual data, multidimensional requirements, and intensive processing demands. As a key research problem in VCS, data selection filters out redundant data and only retains most representative samples, which can effectively reduce the complexity and cost for VCS. In this paper, we study a phase-by-phase data selection approach, in which metadata are first used to pre-select collected photos and then only selected ones are sent to a backend server for further processing based on content features. As such, the initial selection can be completed on nearby edge servers in mobile edge computing (MEC), while more intensive content processing can be done in a remote cloud. We evaluate different initial data selection algorithms using traditional performance measures as well as adapted clustering indices as quality metrics. Moreover, we formulate an integer linear program (ILP) problem for the final data selection based on the scale-invariant feature transform (SIFT) feature. This content-based selection can complement the initial data selection based on contextual metadata. The simulation results show the differences of these selection algorithms and provide guidance on how to choose an appropriate one according to application needs.

INDEX TERMS Mobile crowdsensing, visual crowdsensing, edge computing, data selection, hierarchical clustering, maximum coverage problem.

I. INTRODUCTION

Mobile crowdsensing (MCS) is a cost-effective approach for data collection [1]. It leverages smart devices' built-in sensors and the inherent mobility of device holders to obtain comprehensive knowledge of interesting targets. In particular, visual crowdsensing (VCS) allows a large group of mobile users to share visual data (in the form of photos and videos) acquired by their devices [2]. User-contributed data can be further aggregated to generate insights with greater breadth and depth. With the benefits of low costs, high scalability, and high energy efficiency, VCS finds wide applications in virtual tours, smart cities, environment monitoring, emergency management, and disaster relief [2]. For example, smart vehicles or unmanned vehicles can be used to collect photos of certain regions or road segments in special events.

Though visual data provide rich information, their large sizes and multidimensionality can cause overwhelming demands for processing and transmission [2]. It is also highlighted in [2] that if redundant or irrelevant data can be filtered out and only most representative samples are retained, it can effectively reduce the complexity and cost for VCS. This data selection problem is a key research problem in VCS. In the literature, there have been many existing studies in this area such as [3], [4], [5]. In these works, both context and content features have been considered in similarity and redundancy measurements for data selection. For example, the work in [3] focuses on context features also known as *metadata*. In [4], [6], context and content features are jointly considered.

In fact, the metadata and the actual visual data are complementary. On one hand, a variety of metadata can be accessed easily from a sensor-rich smart device or directly provided by users for a captured photo, such as the location, shoot angle, view coverage, and timestamp. These lightweight metadata can be processed locally. On the other hand, visual content features provide richer information than contextual metadata but require more computational resources for processing. In addition, the transmission of content data to a remote server can be costly and time-consuming. To compare and relate visual objects more effectively, the raw pixel data from the visual contents are often processed to derive new features, such as color histograms and features from scale-invariant feature transform (SIFT) [7]. A content-based method needs to extract such features from the visual contents and further makes use of them for classification, detection, selection or other tasks. The feature extraction and comparison are often computationally intensive, especially, when a large number of visual objects are involved. Therefore, a content-based method may not be responsive enough to some time-critical sensing tasks.

In the literature, some existing works consider both contextual metadata and visual content data together in data selection during the participants-to-server stage (e.g., [8]) or the server-to-requester stage (e.g., [9]). They often require the participants to upload the collected photos or their thumbnails to the server, so that the visual features are extracted and utilized in photo selection. In this paper, we adopt a phase-by-phase data selection approach, in which metadata are first used to pre-select crowdsourced photos. As the metadata sizes are very small and the pre-processing with only metadata is not computationally intensive, this can be completed on nearby edge servers in mobile edge computing (MEC). After that, only selected data samples will be sent to a backend server, which may be hosted in a remote cloud, for further processing based on content features. With this phase-by-phase approach, the close proximity of distributed edge servers can enable real-time data analysis for better timeliness than one-time centralized processing at a remote backend server.

Specifically, the main contributions of this work are summarized as follows:

- We consider a phase-by-phase data selection framework, which first pre-screens photos with validity constraints, then selects promising photo candidates based on context metadata, and last finalizes selection leveraging features extracted from visual contents.
- For the initial data selection based on metadata, we extend the approaches in [3], [4]. Also, for comparison purpose, we consider a clustering-based benchmark approach which is often used in the literature to deal with redundant data. Unsupervised clustering can group data samples according to their features so that close samples are placed into the same cluster but separated from other distant samples. If each photo in the data selection problem is considered as a data sample in clustering, we can group all photos into clusters and select the most representative photo from each cluster. Each selected photo is expected to be similar to the rest of photos in the same cluster but dissimilar to other photos in different clusters. Thus, only these selected photos are retained and others can be screened out.
- We evaluate the performance of several data selection methods on various metrics. The simulation results demonstrate the performance of the methods in different aspects. Existing studies on data selection often use customized quality metrics, such as k-depth coverage [10], quality-aware coverage [11], and temporal-spatial coverage [12]. In addition to such traditional metrics, we also exploit a variety of clustering indices [13] for quality validation and comparison, which is a side benefit of mapping the data selection problem to a clustering problem. Compared with these customized quality metrics, the clustering indices can better integrate the multidimensional attributes of visual data in both the context and content domains and evaluate the overall performance more comprehensively.
- Last, we formulate an integer linear program (ILP) problem for the final data selection based on visual features. A heuristic algorithm is developed to solve a large-scale instance of the ILP problem. The performance of the heuristic algorithm is validated by comparison with that of the optimal solution.

The rest of the paper is organized as follows. In Section II, we explore the related works. In Section III, we present the system model and the data selection problem under study. Section IV further analyzes the problem and gives several candidate solutions. Section V evaluates these solutions and discusses observations from the simulation results. Last, Section VI concludes this paper.

II. RELATED WORKS

Compared to traditional MCS, VCS needs to tackle some unique challenges, such as large data amounts, multidimensional coverage requirements, and complex quality assessments. Consequently, there are some key research problems in VCS, such as diversity-oriented task allocation, efficient data transmission, and representative data selection [2].

In [14], the authors studied how to assign workers to tasks to maximize completion reliability and the spatial/temporal diversities of tasks. In particular, this work defines spatial and temporal diversities in terms of entropies and combines them with a weighted sum. The formulated problem is proved to be NP-hard and solved by using effective approximation approaches, including the greedy, sampling, and divide-and-conquer approaches. Similar temporal-spatial constraints are considered in [12] for multi-task allocation. It defines the temporal-spatial coverage to measure the target sensing quality and further evaluates the overall utility accordingly. A descent greedy approach is adopted to determine the task-worker pairs so that the overall utility is maximized while individual task quality is assured in the meantime.

Data transmission is also essential to VCS due to the inherently large sizes of visual data. A variety of communication techniques have been considered to address the transmission challenges such as opportunistic transmission [15], MEC [16], and disruption-tolerant networks (DTNs) [6]. In [15], the authors proposed a cooperative and selective picture forwarding framework, called CooperSense. In this framework, a tree model called PicTree is used to structure the picture collection from a crowdsensing participant based on the metadata of pictures. When two participants encounter, they exchange their PicTrees at a low transmission cost and then select high-quality pictures by merging these PicTrees. Thus, they only need to selectively forward certain useful pictures to each other.

In [16], the authors proposed a solution where a worker can upload crowdsourced data by leveraging the redundant resources of edge nodes in MEC. Due to the large sizes of visual data, the solution incorporates collaborations among multiple edge nodes to satisfy the transmission demand of one worker. Meanwhile, it requires that the data held by a worker be uploaded completely or not transmitted at all, since part of a captured photo may not provide useful knowledge. As this data transmission problem is proved to be NP-hard, an efficient method based on Lagrangian relaxation is used to obtain an approximate solution.

Furthermore, data selection is another key research problem for VCS. In [3], the authors proposed a framework, called SmartPhoto, to assess crowdsourced photos based on only metadata and select a given number of photos to maximize the total utility. Here, the utility of a photo mainly depends on the number of its covered aspects, which can be conveniently calculated from the metadata. Accordingly, four different optimization problems are studied to trade-off between photo quality and resource constraints.

In [4], the authors proposed an online data selection approach based on a pyramid tree (PTree). The proposed method PicPick can dynamically select an optimal set of pictures from workers based on multidimensional constraints. Contextual metadata and content-based visual features are used together to assess data redundancy in real time. As a result, the thumbnails of incoming pictures need to be sent to the backend server for data selection. In [5], the PTree approach is further used for data grouping at the macro-diversity level based on multidimensional semantic attributes such as location, shooting angle, and shot size. At the micro-diversity level, three data selection schemes are developed for different prioritization needs. In [8], the PTree model is also integrated into a generic framework, called CrowdPic, to solve the multiconstraint-driven data selection problem. A PTree-based data stream clustering method is used in CrowdPic to dynamically divide a data stream into microclusters and select pictures from each microcluster to form the maximum diversified subset (MDS). When a new picture arrives, it is placed into the PTree according to the matching and branching algorithms, while the MDS can get updated.

All the works in [3], [4], [5], [8] focus on pre-data selection during the participants-to-server stage. In contrast, a novel server-to-requester photo selection problem is investigated in [9], [17]. The authors studied how the server selects a subset of high-quality photos from the raw set for the requester to better meet the requester's expectations. In [17], the proposed approach leverages simple metadata information (i.e., GPS location) and SIFT features extracted from photos. A utility measure is designed to assess the quality of a photo set, which integrates an entropy-based spatial diversity factor and a content influence factor based on visual similarities. A greedy approximation algorithm is proposed to solve the NP-hard utility-based photo selection problem. The experiments with real-world datasets show high performance of the proposed approach in terms of photo coverage and view quality.

In [9], the authors further extended the study on server-to-requester photo selection in [17]. They considered a more realistic photo coverage model by taking into account aspects of point of interests (PoIs). Moreover, speeded up robust features (SURF) [18], a speeded-up version of SIFT, are used to assess visual similarities in the calculation of the content influence factor. In addition to a greedy-based algorithm, termed BasicSelection (BPS), a PoI number-aware photo selection scheme (termed PAPS) is further proposed. PAPS first constructs a similarity graph over all photos and partitions it into clusters via spectral clustering, provided that the number of PoIs is known a priori. Then, BPS is run on each cluster to select photos from each cluster independently. PAPS is shown to outperform BPS in photo coverage but performs similarly as BPS in view quality.

In this work, we focus on data selection during the participants-to-server stage as in [3], [4], [5], [8]. Similar to [4], [8], we consider multifacet contextual metadata as well as visual content-based features. This is different from the work in [3], which is only based on geographical and geometrical metadata obtained from built-in cameras of smartphones. In [4], [8], the metadata and visual features are processed together to construct different layers of a PTree for clustering. Thus, thumbnails of all pictures need to be uploaded and processed at the server. In contrast, we consider a phase-by-phase approach, which pre-screens photos based on metadata first and only requires upload of pre-selected photos for final selection based on visual contents.

This work is also different from [9], [17], which focus on the server-to-requester photo selection problem. Because the works in [9], [17] intend to reduce the burden on participants in metadata collection to encourage participation, only simple location information is required. In contrast, our work, as well as [3], [4], [8], study how to exploit a variety of metadata in photo selection when they are available. Another difference between this work and [9], [17] lies in the use of features extracted from visual contents. In [9], [17], the similarity between two photos are defined as the ratio of the number of matched features over all features extracted from them. The sum of visual similarities between a subset of selected photos and its complement is used to evaluate the representativeness of the selected subset. Differently, we directly use the mean distance of good matches identified between two photos to measure their dissimilarity. Then, the sum of pairwise distances of a subset of photos can also assess the distinctness of the selected photos therein.

III. SYSTEM MODELLING

A. DATA SELECTION FOR VISUAL CROWDSENSING

Consider a visual crowdsensing scenario shown in Fig. 1, where a crowdsensing task (denoted by G) is published to collect photos for some interesting object, such as a landmark

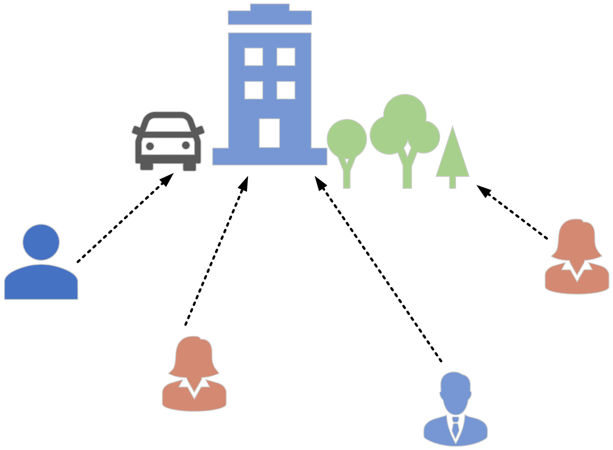


FIGURE 1. Visual crowdsensing scenario.

building, a street segment, or a celebration ceremony. The task and the sensing target need to be specified clearly, e.g., in terms of its temporal, spatial, and quality constraints. A group of mobile users are recruited to accomplish the task by taking and uploading photos (denoted by $\mathcal{P}$) with their smart devices. Due to the large number of photos and the potential redundancy therein, they will be processed to select the most representative ones, for example, to further reconstruct a virtual display.

Foremost, the selected photos should meet the task constraints to be valid. Moreover, the data selection should balance between cost and quality. The number of selected photos is often limited by B to bound the processing and storage costs. Ideally, these selected photos should be as diverse as possible, so that the sensing target is covered comprehensively. Meanwhile, they should have high similarity or redundancy with those photos that are filtered out, so that minimal information is lost after the data selection.

B. TASK MODEL

Referring some previous work [3], [4], we consider a task model that specifies a sensing task and target by the following tuple: $G = \{[g_x, g_y], [t_s, t_e], [\theta_{\min}, \theta_{\max}], \varphi_{\min}, d_{\max}\}$. Here, $[l_x, l_y]$ is the location of the sensing target, $[t_s, t_e]$ is the valid period of performing the task, $[\theta_{\min}, \theta_{\max}]$ is the valid view angles with respect to the target, $\varphi_{\min}$ is the minimum coverage span within the above range required for a valid photo, and $d_{\max}$ is the maximum acceptable distance between the shoot location and the target. As seen, a valid sensing photo for a task has to be taken within the valid period and distance, while it must cover sufficient views within the expected range.

C. DATA FEATURES

To select a limited number of qualified photos for a task, we consider two types of data features, namely, the context metadata recorded with a photo, and certain visual features extracted from the content data. Let F denote the number of context-based metadata features. Specifically, the metadata associated with photo $P_i \in \mathcal{P}$ are given by $M_i = \{[x_i, y_i], t_i, [\alpha_i, \beta_i], \varphi_i, d_i\}$. Here, $[x_i, y_i]$ and t_i are where and when the photo is taken, respectively. Accordingly, the shoot distance to the target d_i and the shoot angle φ_i can be derived. Last, $[\alpha_i, \beta_i]$ represents the range of views covered by the photo. A valid photo for the sensing task has to meet the conditions that $d_i \leq d_{\max}$, $t_s \leq t_i \leq t_e$, $|\beta_i - \alpha_i| \geq \varphi_{\min}$, and $[\alpha_i, \beta_i] \subseteq [\theta_{\min}, \theta_{\max}]$.

In addition, we can extract visual features from the captured photos. One powerful technique is the feature detection algorithm, SIFT [7]. A good property of SIFT is that it is invariant to the size or orientation of a photo. SIFT has been shown to be robust in identifying objects even in presence of clutter and occlusion [7]. In [19], SIFT is also tested with kinds of image distortions, including scaling, rotation, fisheye, shearing and salt and pepper noise. It is found that SIFT performs the best in most scenarios and achieves high matching rates.

Using SIFT, we can locate a collection of image features commonly known as “keypoints,” which are scale and rotation invariant. Further, a “descriptor” is generated from the samples in the neighbourhood of each keypoint, as a unique fingerprint for the keypoint. Here, we denote the visual content feature of photo P_i by $V_i = \{\mathcal{K}_i, \mathcal{D}_i\}$, where $\mathcal{K}_i$ and $\mathcal{D}_i$ represent the sets of keypoints and descriptors generated from photo P_i, respectively.

D. QUALITY METRICS

To bound the transmission and processing costs, we intend to select certain most representative photos from the photo set. The selected photos are expected to be dissimilar to each other but more similar to the photos that are filtered out. To quantify the selection effectiveness, there are various metrics proposed in the literature.

For example, the coverage metric and its variants have been widely used, such as k-depth coverage [10] and quality-aware coverage [11]. In [4], the proposed approach aims to maximize the coverage by the selected photos. An eliminated photo is counted as covered by a selected photo if the distances between the two photos with respect to all features fall within the corresponding similarity thresholds. For instance, if the Euclidean distance between their shoot locations is no more than a threshold, say 5 meters, one photo is considered covered in terms of the shoot distance. If an eliminated photo is similar to a selected photo on all features, the eliminated photo is said to be redundant and covered. It is ideal that the set of selected photos can maximize the coverage and even provide full coverage. As seen, this coverage metric is simple and easy to compute. However, this 0/1 count of coverage cannot quantify redundancy elimination in a finer granularity. Moreover, this coverage metric cannot characterize the diversity among the selected photos.

Differently, the work in [3] defines a utility measure, which is proportional to the range of aspects covered by a photo. For instance, a photo with a coverage interval $[40^\circ, 110^\circ]$ has a utility that is twice that of another photo with a coverage interval $[50^\circ, 85^\circ]$. Clearly, the larger the overlap between two photos' coverage intervals, the higher the similarity and redundancy between them. Moreover, in the above case, their coverage intervals not only overlap, but also the coverage of the latter photo is a subset of that of the former photo. Then, the latter photo can be discarded assuming that it does not provide new information. As seen, this utility measure can quantify the similarity between two photos in a scale finer than a $0/1$ count. However, the work in [3] mainly focuses on maximizing the covered aspects, while the similarity and redundancy among a set of photos depend on more features such as shoot time, shoot distance, and visual content.

To select a number of diverse photos from a whole set, a natural idea is to cluster the photo set into groups and choose the one that is the closest to the centroid of each cluster, which is presumably the most representative photo of each group. Ideally, each cluster should be dense within the cluster but well separated in between. To evaluate the clustering performance, many metrics have been implemented in scikit-learn [20] when there is no ground truth, e.g., the silhouette coefficient, the Davies-Bouldin index, and the Calinski-Harabasz index (also known as the variance ratio criterion). We can extend these metrics to evaluate the data selection performance.

The silhouette coefficient of each point x_i in a dataset can be written as [21]

$$S_i = \frac{b_i - a_i}{\max\{b_i, a_i\}}, \ -1 \leq S_i \leq 1. \tag{1}$$

Here, b_i is the nearest-cluster distance, i.e., the average distance between point i and all points in the nearest cluster. In other words, b_i quantifies the cluster separation, since it is the smallest average distance of x_i to all points in any other cluster, of which x_i is not a member. The cluster cohesion a_i gives the average distance between point x_i and all other points in the same cluster. As seen, b_i captures how dissimilar a point is to other clusters, and a_i tells how similar it is to the other points in its own cluster. The mean silhouette coefficient of all points is the silhouette score of the clustering. The score is close to 1 when clusters are dense and well separated.

The Davies-Bouldin index computes the average similarity of each cluster with its most similar one. The similarity of cluster j and cluster k can be evaluated by [21]

$$R_{jk} = \frac{s_j + s_k}{d_{jk}} \tag{2}$$

where s_k is the average distance between each point of cluster k and its centroid, also known as the cluster diameter, and d_{jk} is the distance between the centroids of clusters j and k. As seen, the similarity measure is the ratio of the within-cluster distance to the between-clusters distance. Then, the Davies-Bouldin index of K clusters can be expressed as [21]

$$DB = \frac{1}{K}\sum_{k=1}^{K} \max_{j \neq k} R_{jk}. \tag{3}$$

Clearly, the Davies-Bouldin score is close to 0 when the clusters are far apart.

In the data selection problem, the selected photos are expected to be dissimilar to each other but similar to screened out photos. When we evaluate any given data selection solution, a selected photo is not necessarily the centroid of a cluster or its mean point. It may not even be the point that is the closest to the centroid. However, the Davies-Bouldin index is based on the distances to the centroid, which may not represent well the features of a selected photo. Hence, the Davies-Bouldin index cannot evaluate the performance of any data selection solution accurately. Therefore, we extend the above definition of the Davies-Bouldin index by generalizing the use of centroids in the calculation. Instead of taking the mean point of each cluster as the centroid, we can use any given point as the centroid of the cluster, which is mapped to the photo selected from each cluster. As such, the Davies-Bouldin index measures the similarity with respect to a real given photo instead of a virtual mathematic centroid of the cluster.

Last, the Calinski-Harabasz index is the ratio of the sum of between-clusters dispersion and the sum of within-cluster dispersion of all clusters, while dispersion is measured by the sum of squared distances [21]. Specifically, the Calinski-Harabasz index of K clusters for a dataset of totally N points can be written as [22]

$$CH = \frac{\sum_{k=1}^{K} \frac{n_k}{N-1} \|\mu_k - \mu\|^2}{\sum_{k=1}^{K} \frac{1}{N-K} \sum_{i=1}^{n_k} \|x_i - \mu_k\|^2}. \tag{4}$$

Here, for any cluster k, there are centroid μ_k and n_k points denoted by x_i, where $i = 1, \ldots, n_k$, while μ is the global centroid. The separation of clusters in the numerator is based on the distances of the cluster centroids from the global centroid, while the internal cluster cohesion in the denominator is estimated by the distances from the data points within a cluster to its cluster centroid. Hence, a higher score indicates better clustering that is dense within each cluster but well separated between clusters. Due to the similar reason for adapting the Davies-Bouldin index, we also extend the Calinski-Harabasz index by using any given point to replace a centroid. Then, we can better map a data selection solution to a clustering result for quality evaluation.

IV. PROBLEM ANALYSIS AND SOLUTION

Given a sensing task G, a group of participants collect a set of photos $\mathcal{P}$. To select no more than B photos from the set, we take a three-phase approach. First, all participants upload the metadata of their photos to the crowdsensing server. The cost is acceptable considering the small sizes of the metadata. The server then pre-screens the photos based on the metadata and identifies the valid ones that meet the temporal, spatial, and quality constraints given in Section III-B. The group of valid photos after pre-screening is denoted by $\mathcal{P}_v$.

Assuming that there are more than B valid candidate photos, we need to proceed and refine the selection. Based on the context metadata, we select γB $(1 \leq \gamma \leq |\mathcal{P}_v|/B)$ photos from the valid candidates in the second phase. These γB photos should be as diverse as possible, while the other eliminated photos are redundant with respect to the selected ones. In the last phase, the server requests the original files of these γB selected photos from the corresponding participants and selects B photos therein by further processing the visual contents. With the initial selection based on the lightweight metadata, this step now becomes more manageable. Moreover, the ratio γ can be adjusted according to the available resources for transmission and computation. When a larger cost is affordable, the server can set a larger value for γ and choose more candidates based on the metadata.

A. INITIAL SELECTION BASED ON CONTEXT METADATA

1) SMARTPHOTO$^+$ FOR INITIAL SELECTION

For the initial selection based on the photo metadata, we consider three candidate approaches. First, we can extend the scheme SmartPhoto proposed in [3]. For ease of comprehension, let us introduce the main steps of SmartPhoto in the following. Consider a single target G. Each photo P_i covers a range of aspects of the target, represented by an interval $[\alpha_i, \beta_i]$. With the coverage intervals of all photos in set $\mathcal{P}_v$, we can split the whole range of full coverage $[0, 360^\circ)$ into a number of sub-intervals, by taking the boundaries of each interval as the dividing points. These sub-intervals form a universe set of elements, where the length of the sub-interval for an element is defined as its weight. Then, each photo covers a subset of elements in the universe set. The total utility of a photo selection result is the total weight of the elements covered by the selected photos. Thus, the data selection problem is formulated as an instance of the maximum coverage problem [23], which is to find a bounded number of subsets to maximize the total weight of the elements covered by the selected subsets. As the maximum coverage problem is known to be NP-hard, a greedy selection algorithm is used in [3] for the photo selection. The basic idea is to iteratively select an unselected subset, which has the highest incremental contribution to the total utility. That is, the elements in the new subset that are uncovered so far have the largest total weight. The selection process continues until the required number of subsets (photos) have been selected or every aspect of the target has been covered.

As the original algorithm focuses on the aspect coverage, we need to incorporate more context features, including the shoot angle, the shoot time, and the distance to the sensing target. Algorithm 1 shows the extended version, subsequently referred to as SmartPhoto$^+$. As seen, there are two main procedures. In Lines 1–6, the whole set of elements are defined with respect to each feature in the context metadata, and each candidate photo is mapped to a subset of such elements. While the aspect coverage is specified by a range of view angles, the other features are single values. For each single-valued feature, we just divide its data scale to a number of uniformly distributed intervals, which represent the set of elements for this feature. If a photo's feature falls into an interval, it means that the photo covers the element corresponding to this interval. Since these intervals are uniformly distributed, each element has the same weight 1. The aspect coverage is defined by two end points for the view angles, e.g., $[30^\circ, 70^\circ]$. The end points of all photos divide the entire coverage interval $[0, 360^\circ)$ to a set of sub-intervals. As considered in [3], each sub-interval is added to the universe set of elements, while its weight is proportional to the length of the sub-interval. Moreover, we normalize the weight to [0, 1] to be comparable with the elements from other features. For each photo, a subset of elements is generated according to which coverage sub-intervals it falls into.

Algorithm 1: SmartPhoto$^+$: An Algorithm for Initial Photo Selection Based on Metadata Extended from [3].

```
Input: Sensing task $G$, metadata for valid photo set:
       $\{M_i : \forall P_i \in \mathcal{P}_v\}$, maximum photo number for
       initial selection $\gamma B$
Output: Initial selection $x = \{x_i : \forall P_i \in \mathcal{P}_v\}$, cluster
        labelling $y = \{y_i : \forall P_i \in \mathcal{P}_v\}$
1  begin Define elements and their weights in universe
   set and subsets of elements of all photos
2      for each context feature $f$ do
3          Define the whole set of elements w.r.t. current
           feature $f$;
4          Set and normalize the weight of each element
           in the whole set;
5      for each photo $P_i \in \mathcal{P}_v$ do
6          Obtain subset of elements covered by $P_i$;
7  begin Select subset of photos and cluster unselected
   photos accordingly
8      Select $\gamma B$ photos using the greedy algorithm in [3] for
       the maximum coverage problem, record solution in $x$;
9      Normalize metadata features of all valid photos;
10     Set each selected photo as the centroid of a cluster;
11     for each unselected photo $P_i \in \mathcal{P}_v$ and $x_i = 0$ do
12         Compute distances of $P_i$ to all centroids;
13         Assign $P_i$ to the cluster with the smallest
           distance to its centroid, record clustering in $y_i$;
14 Return $x$, $y$;
```

After defining the elements of the universe set and their weights and the subset of elements covered by each photo, Lines 7–13 select a limited number of photos and label all photos into the clusters. First, the greedy algorithm in [3] is used to solve an instance of the maximum coverage problem, which gives γB representative photos to achieve a high total utility. Then, each selected photo is taken as a cluster centroid, while each remaining unselected photo is classified into the cluster with the nearest centroid. For ease of distance calculation, we normalize the context features before the classification.

The algorithm SmartPhoto$^+$ is a potential solution to the initial selection of candidate photos only based on the context metadata. Compared to the original approach SmartPhoto, we further take into account multiple features. Accordingly, we need to properly weight and normalize the feature elements. In addition, we map the selection result to a straightforward clustering so that it can be evaluated with various clustering indices discussed in Section III-D.

2) PICPICK$^+$ FOR INITIAL SELECTION

Another interesting approach for the initial photo selection is PicPick proposed in [4]. PicPick uses a pyramid tree (PTree) structure to group pictures with multidimensional features. PTree has $(F + 2)$ layers, where F is the number of features under consideration. The root node is at Layer-0, while Layer-1 to Layer-F is mapped to one of the F features. The bottom Layer-$(F + 1)$ consists of the leaf nodes, each of which corresponds to a picture instance. Each subset of leaf nodes on the bottom layer descending from the same node on Layer-F naturally forms a cluster of pictures, and one representative picture can be selected from each cluster. Here, we re-define the label of each node in the tree. The label of each node is given by $L\langle\ell\rangle$-$N\langle n\rangle$, where $\langle\ell\rangle$ is the layer of the node and $\langle n\rangle$ is simply the order of the node added to the layer.

To generate a PTree, it begins with the root node. Then new nodes are added in according to a sequence of pictures. Each node at Layer-1 to Layer-F (excluding the top and bottom layers) is associated with an attribute with respect to the feature used on that layer (e.g., the shoot angle). This attribute is obtained from the features of the leaf nodes descending from the node. Specifically, it is the mean value of these leaf nodes for the feature of that layer, e.g., the mean shoot angle of leaf descendants on the bottom layer. Thus, each picture can be added into the tree by comparing its distances to the attributes of the nodes already in the tree.

PicPick originally uses context features and visual content features together. Since we intend to streamline the data selection procedure phase by phase, we can use PicPick for the initial photo selection based on only context metadata. However, one limitation of PicPick is that it does not restrict the number of node clusters on the bottom layer. If one picture is selected per cluster, we cannot limit the number of selected pictures. Hence, we need to extend PicPick on the generation of the PTree. Algorithm 2 shows the algorithm PicPick$^+$ extended from [4]. Similar to SmartPhoto$^+$, there are two main procedures. Lines 1–12 generate a PTree by processing the set of pictures one by one. For each picture, it traverses the current PTree from the root node, and compares the features of the new picture with the attributes of the existing non-leaf nodes on Layer-1 to Layer-F. If a non-leaf node is within a distance threshold to the new picture, it is called a match in [4]. Note that match here should be distinguished from match of SIFT features to be discussed in Section IV-B. Otherwise, a new non-leaf node is created on the current layer. It is worth noting that the distance thresholds are important to PicPick and PicPick+ and should be carefully selected according to the requirements of the crowdsensing task. Then, we move downward to the next layer through the matched node or the newly created node. Finally, the new picture is added to the bottom layer, i.e., Layer-$(F + 1)$, as a leaf node.

Algorithm 2: PicPick$^+$: An Algorithm for Initial Photo Selection Based on Metadata Extended from [4].

```
Input: Sensing task $G$, metadata for valid picture set:
       $\{M_i : \forall P_i \in \mathcal{P}_v\}$, maximum picture number
       for initial selection $\gamma B$
Output: Initial selection $x = \{x_i : \forall P_i \in \mathcal{P}_v\}$, cluster
        labelling $y = \{y_i : \forall P_i \in \mathcal{P}_v\}$
1  begin Generate PTree based on features of picture set
2      Set features for Layer-1 to Layer-$F$;
3      Create a directed graph and add root node;
4      for each picture $P_i \in \mathcal{P}_v$ do
5          for each Layer-$\ell$, $\ell = 1, \cdots, F$ do
6              Compare distances between the feature of
               $P_i$ for Layer-$\ell$ and those of existing
               non-leaf nodes on Layer $\ell$;
7              if number of clusters is less than $\gamma B$, and
               no node on Layer-$\ell$ is within a distance
               threshold to $P_i$ then
8                  Add a new node to Layer-$\ell$ and set it as
                   parent;
9              else
10                 Find the node on Layer-$\ell$ that is the
                   closest to picture $P_i$ in terms of feature
                   of Layer-$\ell$ and set it as parent;
11             Move to next layer below the parent node;
12         Add picture $P_i$ as a new leaf-node under the parent
           node selected above for Layer-$F$;
13 begin Label clusters for all pictures based on the
   PTree and select a subset of pictures from the clusters
14     for each node $LF$-$Nj$ on Layer-$F$ do
15         Find the leaf nodes on Layer-$F + 1$ under node
           $LF$-$Nj$;
16         Assign cluster label $j$ to the group of pictures
           corresponding to these leaf nodes and record
           clustering in $y$;
17         Compute the centroid of this group of pictures;
18         Select the picture that is the closest to the
           centroid and record selection result in $x$;
19 Return $x$, $y$;
```

In the above procedure, there are two key aspects. First, we need to properly calculate the distance between a non-leaf node and a new picture based on the layer feature. For a single-valued feature, such as the shoot angle and shoot distance, we can simply use the Euclidean distance as in [4]. However, for the distance between two coverage intervals inspired by how coverage intervals are handled in [3], we first find their overlapping sub-interval if any, normalize the length of the sub-interval with the maximum range covered by the two intervals, and last take one minus the ratio. This distance can quantify the dissimilarity between the two intervals. Second, we need a match policy to match a new picture to a non-leaf node when traversing the PTree. In [4], the authors proposed two match policies, namely first-match and min-match. First-match just takes the first matched non-leaf node on Layer-1 to Layer-F if any exists, whereas min-match compares the distances of the new picture to all matched nodes on a layer and chooses the node with the shortest distance to the picture.

As mentioned earlier, one limitation of PicPick is that it does not restrict the number of clusters on the bottom layer, which makes it not applicable to our problem. Hence, we adapt the algorithm in Lines 7–10. Here, we only add a new non-leaf node on a layer if the number of existing clusters has not reached the limit. Moreover, we apply the min-match policy and always choose the closest non-leaf node as a parent, even when this node is not within the distance range and a maximum number of clusters have been created. As such, we do our best to group pictures into the PTree while meeting the selection constraint.

The second procedure in Lines 13–18 labels the clusters for all pictures according to the above generated PTree and selects a subset of pictures from the clusters. In the original algorithm PicPick, it simply picks the first picture added into each cluster for timely processing. Here, since we only consider metadata in the initial photo selection, it is affordable to choose a more representative picture for each cluster. Hence, in Line 18, referring to the mean-priority strategy in [5], we select the picture that is the closest to the centroid of each cluster in terms of the feature distances.

3) CLUSTERING ALGORITHMS FOR INITIAL SELECTION

In addition to SmartPhoto$^+$ and PicPick$^+$, for comparison purpose, we also consider a clustering-based benchmark algorithm which is often used in the literature to process redundant data. The idea is to consider the set of valid photos as data samples, use a clustering algorithm to group them into γB clusters, and then choose one representative photo from each cluster. Since a clustering tends to be dense within the cluster but well separated in between, the selected photos are therefore expected to be diverse but redundant to the eliminated ones. Algorithm 3 shows the approach for initial photo selection based on clustering and centroids. It first normalizes the features of all valid photos. Then, a clustering algorithm is run to separate the photos into γB clusters. In the literature, there have been a variety of clustering algorithms to choose from, such as k-means, mean shift, spectral clustering, density-based spatial clustering of applications with noise (DBSCAN), and agglomerative clustering. Last, it selects the photo that is the closest to the centroid of each cluster.

Algorithm 3: ClusterFirst: Initial Photo Selection Based on a Standard Clustering Algorithm.

```
Input: Sensing task $G$, metadata for valid photo set:
  $\{M_i : \forall P_i \in \mathcal{P}_v\}$, maximum photo number for initial
  selection $\gamma B$
Output: Initial selection $x = \{x_i : \forall P_i \in \mathcal{P}_v\}$, cluster
  labelling $y = \{y_i : \forall P_i \in \mathcal{P}_v\}$
Normalize features of all photos in $\mathcal{P}_v$;
Choose a clustering algorithm, e.g., agglomerative
  clustering;
Run the clustering algorithm to separate photos in $\mathcal{P}_v$
  into $\gamma B$ clusters;
Assign cluster labels to photos in $\mathcal{P}_v$ and record
  clustering in $y$;
Compute the centroid of each cluster;
Select the photo that is the closest to the centroid and
  record selection result in $x$;
Return $x$, $y$;
```

B. FINAL SELECTION BASED ON VISUAL CONTENT

After the valid photo set is initially examined based on context metadata, we end up with a much smaller photo subset, denoted by $\mathcal{P}_s$. Then, it becomes affordable to analyze their visual features and further select B most distinct photos therein. Here, we take the SIFT-based visual feature as one example, while this can be extended to other visual features as well. Note that the pre-selected photos may contain clutter, blur, occlusion or other distortions. Such noises may interfere with the visual feature extraction and matching, although SIFT has good robustness even in matching noisy images [7], [19]. If a noisy photo provides complementary information that is absent in other photos, it may be selected but not meet the requester's expectation on image quality. Therefore, if the requester has a requirement on the minimum image quality, we can use an image quality assessment (IQA) approach [24], [25] to filter out photos of low image quality first. As such, in the subsequent photo selection, we can avoid the interferences from noisy photos and do not need to deal with conflicting decisions with respect to photo uniqueness and image quality, of course, at the additional cost of IQA screening.

Algorithm 4 shows the final photo selection algorithm based on the SIFT feature. Here, we expect to select a subset of dissimilar photos that can provide complementary information. As seen, Algorithm 4 consists of two main procedures. In Lines 1–12, we extract the visual features of pre-selected photos and evaluate their pairwise distances. In Line 3, we derive the SIFT feature V_i from each pre-selected photo P_i, where $V_i = \{\mathcal{K}_i, \mathcal{D}_i\}$, respectively. Here, $\mathcal{K}_i$ and $\mathcal{D}_i$ represent the sets of keypoints and descriptors generated from photo P_i. Given that there are ρ keypoints identified for photo P_i, $\mathcal{K}_i$ contains ρ keypoint objects, while $\mathcal{D}_i$ is an $\rho \times 128$ array describing the 16×16 neighborhood of each keypoint. Then, in Line 6, we measure the distances between the descriptor features of each pair of photos to find possible match(es) if there is any. That is, each descriptor of one photo is compared with all descriptors of the other photo to find one or more matches with short distances. The distances are evaluated by Euclidean distances based on L2-norm.

As the k-nearest neighbour (KNN) matching algorithm was shown to be generally robust, it is considered in Algorithm 4. If there are ρ descriptors from photo P_i, the KNN algorithm outputs $\rho \times k$ matches from photo P_i to photo P_j. Among the k matches for each descriptor, Line 7 further takes the ratio test [7] to decide whether to keep the top match. Only if the shortest distance of the top match is less than ϕ ($0 < \phi < 1$, e.g., $\phi = 0.75$) of that of the second match is the top match considered as a good match. As such, only those matches with

```
Algorithm 4: Final Photo Selection Based on SIFT Visual Features.
   Input: Sensing task $G$, pre-selected photo set $\mathcal{P}_s$,
          maximum photo number for final selection $B$
   Output: Final photo selection $x = \{x_i : \forall P_i \in \mathcal{P}_s\}$
1  begin Extract visual features of pre-selected photos
   and evaluate pairwise distances
2      for each photo $P_i \in \mathcal{P}_s$ do
3          Extract keypoints $\mathcal{K}_i$ and descriptors $\mathcal{D}_i$ from
           photo $P_i$;
4      for each photo $P_i \in \mathcal{P}_s$ do
5          for each photo $P_j \in \mathcal{P}_s$ do
6              Evaluate L2-norm distances from
               descriptors of $P_i$ to those of $P_j$ and find $k$
               matches for each descriptor with KNN;
7              Take ratio test among $k$ matches with
               distances $d_i$ $(i = 1, 2, ..., k)$ for each
               descriptor, keep top match only if
               $d_1 < \phi \cdot d_2$;
8              if there exists at least one final match then
9                  Take mean distance of the match(es)
                   from $P_i$ to $P_j$ and set it to $W_{ij}$ of
                   distance matrix;
10             else
11                 Set $W_{ij}$ to a large constant;
12     Set each pair of elements $W_{ij}$ and $W_{ji}$ of distance
       matrix to their maximum value;
13 begin Select a subset of $B$ photos that are most distant
   in between
14     Use distance matrix $W$ to formulate an ILP
       problem;
15     if problem instance is relatively small-scale then
16         Solve the problem with an ILP solver and
           record solution in $x$;
17     else // Solve the instance with a heuristic algorithm
18         Initialize $x \leftarrow \{x_i = 0 : \forall P_i \in \mathcal{P}_s\}$, selected
           photo set $\mathcal{S} \leftarrow \varnothing$, eliminated photo set
           $\mathcal{U} \leftarrow \mathcal{P}_s$;
19         Based on distance matrix $W$, compute mean
           distance of each photo in $\mathcal{P}_s$ to others in $\mathcal{P}_s$;
20         Select photo $P_{i^*} \in \mathcal{P}_s$ with the highest mean:
           $\mathcal{S} \leftarrow \mathcal{S} \cup P_{i^*}, \mathcal{U} \leftarrow \mathcal{U} \setminus P_{i^*}, x_{i^*} \leftarrow 1$;
21         while $\sum_i x_i < B$ do
22             Compute mean distance of each photo in $\mathcal{U}$
               to those in subset $\mathcal{S}$;
23             Select photo $P_{i^*} \in \mathcal{U}$ with the highest
               mean distance: $\mathcal{S} \leftarrow \mathcal{S} \cup P_{i^*}, \mathcal{U} \leftarrow \mathcal{U} \setminus P_{i^*}$,
               $x_{i^*} \leftarrow 1$;
24 Return $x$;
```

sufficiently small distances are retained, and there are at most ρ such good matches identified for two photos, where there is at most one good match for each descriptor.

After finding the good matches between two pre-selected photos, the mean distance of the matches is recorded as a corresponding element in a distance matrix (Line 9). If no good match is found, a large constant is used in the distance matrix (Line 11), which indicates that the two corresponding photos are very distinct. As the so-generated distance matrix may not be symmetric, we further take the maximum value of each pair of elements W_{ij} and W_{ji} to reset their values (Line 12). Then, the distance matrix becomes symmetric and better quantify the dissimilarity of each pair of photos.

In the second procedure in Lines 13–23, we use the above generated distance matrix W to select B most distinct photos from set $\mathcal{P}_s$. Naturally, we expect that the selected photos are as dissimilar as possible. Such a subset of diverse photos can cover the sensing target comprehensively. Translating this idea to a mathematical problem, we can formulate the photo selection as the following optimization problem:

$$(P) \quad \max_x . \sum_i \sum_{j>i} x_i x_j W_{ij} \tag{5a}$$

$$\text{s.t.} \quad \sum_i x_i \le B. \tag{5b}$$

Here, x_i is a binary decision variable, indicating whether photo $P_i \in \mathcal{P}_s$ is selected, and W_{ij} is the distance between photos P_i and P_j estimated in the above procedure. As the distance matrix is symmetric, we only need to consider the upper triangular portion of the distance matrix above the diagonal. The only constraint is to limit the total number of selected photos by B.

To solve this optimization problem, we can reformulate it as an ILP problem as follows:

$$(D) \quad \max_x . \sum_i \sum_{j>i} z_{ij} W_{ij} \tag{6a}$$

$$\text{s.t.} \quad z_{ij} \le x_i, \ \forall i, j \text{ and } j > i \tag{6b}$$

$$z_{ij} \le x_j, \ \forall i, j \text{ and } j > i \tag{6c}$$

$$x_i + x_j - 1 \le z_{ij}, \ \forall i, j \text{ and } j > i \tag{6d}$$

$$\sum_i x_i \le B. \tag{6e}$$

In (6), one additional binary variable z_{ij} is introduced, which can be interpreted as an indicator on whether distance W_{ij} between photos P_i and P_j is counted in the objective value. Obviously, z_{ij} is related to x_i and x_j. As defined in constraints (6b)–(6d), $z_{ij} = 1$ only if both $x_i = 1$ and $x_j = 1$. That is, to have $z_{ij} = 1$ and count distance W_{ij}, both photos P_i and P_j should be selected.

When the ILP problem has a small scale, it can be solved by modern solvers, such as Gurobi [26] and the GNU linear programming kit (GLPK) [27]. However, due to the high complexity, it can be slow to solve a large-scale instance. Hence, we also use a heuristic algorithm given in Lines 17–23 to solve it. Initially, it chooses the first photo that is the most distant to all others. Then, it iteratively selects a new photo with the largest mean distance to other already selected photos. The subsets of selected photos and eliminated photos are updated

TABLE 1. Experiment Parameters

Parameter	Value
Sensing region	$[50, 50]$
Sensing period	$[0, 80]$
Range of max. valid sensing distance to target	$[30, 60]$
Range of valid shoot duration of target	$[20, 60]$
Range of valid views of target $[\theta_{\min}, \theta_{\max}]$	$[0^\circ, 360^\circ]$
Range of minimum valid coverage span of target	$[20^\circ, 90^\circ]$
Range of coverage span of photos	$[30^\circ, 120^\circ]$
Range of split ratio for range of views of photos	$[0.1, 0.9]$
Similarity threshold for shoot distance	10
Similarity threshold for shoot time	15
Similarity threshold for range of views	0.3
Cluster radius of Matérn process	$\frac{1}{3}$ sensing region width
Cluster size of Matérn process (maximum number of points per cluster)	$\frac{2\times\text{total num. of points}}{\text{number of clusters}}$
Number of initially selected photos	20
Number of finally selected photos	7
Total number of photos	500

accordingly. This procedure continues until B photos are selected.

V. SIMULATION RESULTS

In this section, we evaluate the data selection approaches introduced in Section IV. First, we introduce the datasets used in the performance evaluation, including those for the contextual metadata and the visual content data. Then, we present the results for the initial photo selection based on metadata. Last, we show the performance of the final photo selection.

A. DATASETS

1) DATASETS FOR CONTEXT METADATA

In the following experiments, we use both synthetic and real datasets to evaluate the data selection approaches. The key parameters are listed in Table 1.

As given in Section III-B, a sensing task target is specified by tuple $G = \{[g_x, g_y], [t_s, t_e], [\theta_{\min}, \theta_{\max}], \varphi_{\min}, d_{\max}\}$. We generate the target location $[g_x, g_y]$ uniformly within a rectangular region. The valid start time t_s is randomly picked within a maximum sensing period, and the task lasts for a random duration within a range, which gives the valid end time t_e bounded by the maximum sensing period. The range of valid views $[\theta_{\min}, \theta_{\max}]$ is simply set to $[0^\circ, 360^\circ]$. The minimum valid coverage span $\varphi_{\min}$ is taken uniformly in a given range. The maximum acceptable shoot distance $d_{\max}$ is also randomly selected from a range. According to the task information, we can set the similarity thresholds, which are used in the PicPick-based approaches and for the coverage metric. Some example values are given in Table 1.

For each photo $P_i \in \mathcal{P}$, its context-based metadata are given by $M_i = \{[x_i, y_i], t_i, [\alpha_i, \beta_i], \varphi_i, d_i\}$. The shoot time t_i is set uniformly within the whole sensing period. The sensing location $[x_i, y_i]$ can be taken from a real dataset or generated as a clustered random point process within a region slightly larger than the sensing region (to be discussed in detail). Based on the sensing location, the shoot distance d_i to the target and the view angle φ_i can be derived. The range of views covered by the photo is given by $[\alpha_i, \beta_i]$. We randomly generate a width for the view coverage and a ratio less than 1. The ratio defines how the view angle φ_i splits the width of the coverage in $[\alpha_i, \beta_i]$. That is, the width of the sub-interval $[\alpha_i, \varphi_i]$ over that of the entire interval $[\alpha_i, \beta_i]$ is given by the ratio.

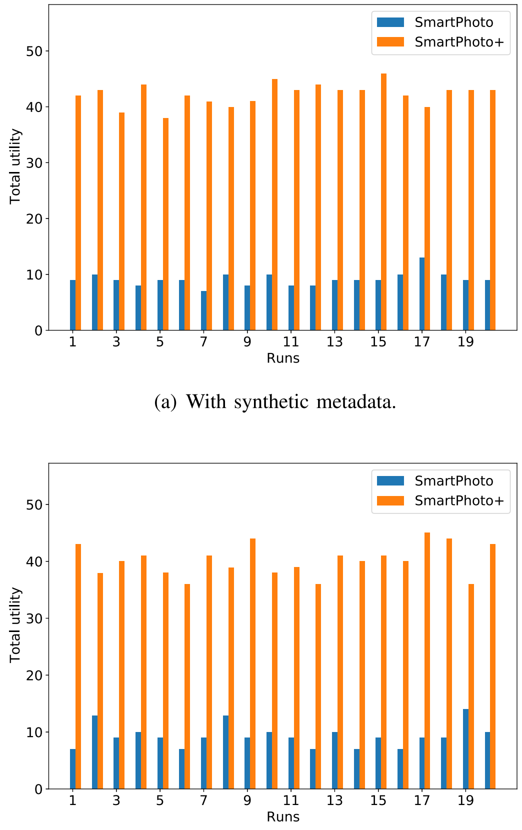
To get synthetic location data, we can use a cluster point process such as the Matérn process to model the locations. For a Matérn process, the cluster centres are generated according to a homogeneous Poisson point process (PPP), while each cluster contains a random number of children points that are uniformly distributed within a circular area of a given radius and a centre from the preceding PPP. To just generate a total number of location points, we also limit the maximum number of points in each cluster, as shown in Table 1. Since the sensing locations tend to cluster in the surrounding environment of the target, this Matérn-like process can capture the clustering effect while meeting our simulation needs.

To get more realistic location information, we can also use some existing datasets such as the New York City (NYC) and Tokyo (TKO) check-in datasets [28]. They were crawled from Foursquare [29], which is a location data platform and mobile app. These datasets include check-in data at venues in NYC and TKO collected from April 12, 2012 to February 16, 2013. Each check-in record is a tuple (user ID, venue ID, latitude, longitude, date and time). The latitude and longitude are the GPS coordinates obtained from the check-in user device.

The datasets contain 227,428 check-ins in NYC and 573,703 check-ins in TKO. As users may visit a venue repeatedly, it is time-consuming and unnecessary to consider all records in the datasets. Instead, we pre-process the data as proposed in [30] to get a required number of location records. Specifically, we sample 500 records from the TKO dataset in each run and they are sufficient to capture the data characteristics that our experiments need. Here, we first load the check-in records and filter them by keeping the newest check-in record for each mobile user but removing duplicate user IDs and venue IDs. Then, we limit the scales in space and time to select a required subset of samples that meet the distance and time constraints from the above filtered records. Finally, the GPS locations of the selected records are mapped to 2D Cartesian locations for ease of calculation. We also normalize the location coordinates and scale them into the given sensing region.

2) DATASETS FOR VISUAL CONTENT

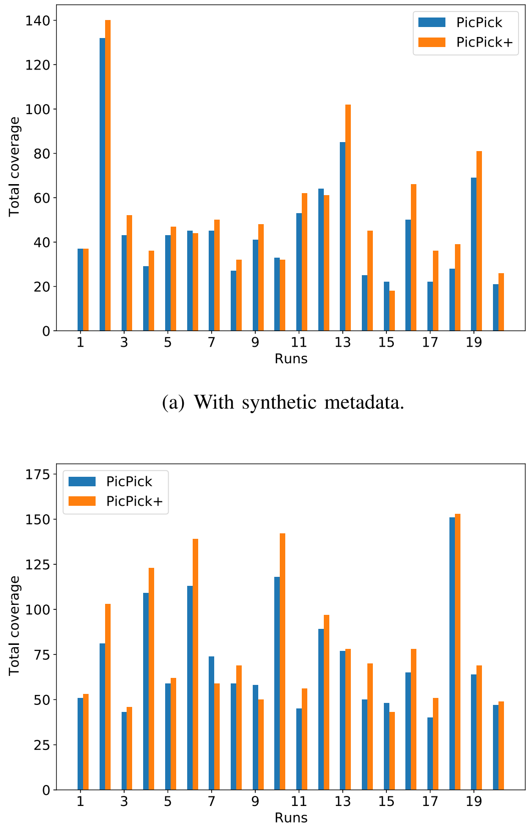
It is challenging to create a usable photo set with a variety of metadata under consideration. First of all, the photos must be toward the same target and also significantly vary in certain aspects such as shoot time, shoot angle, and coverage of views. Hence, the photo set needs to be sufficiently large so that there remain a good number of valid candidates after the dataset is pre-screened with the validity constraints.



(a) With synthetic metadata.

(b) With real location data.

FIGURE 2. Total utility.



(a) With synthetic metadata.

(b) With real location data.

FIGURE 3. Total coverage.

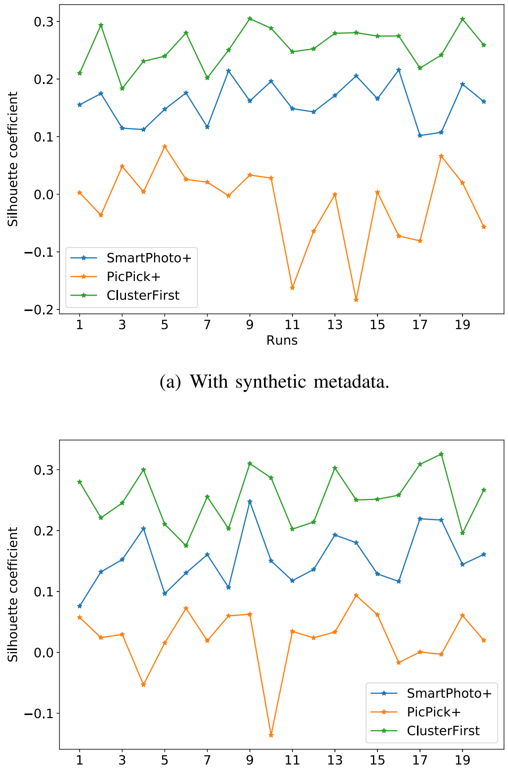
Unfortunately, we are not able to create or find such an ideal photo set. Nonetheless, since some key aspects used in photo selection are the angle, span and range of views, we find that the Columbia University Image Library (COIL-100) [31] can meet our needs in a degree. This library contains a database of color images of 100 objects, which were shot through 360 degrees at pose intervals of 5 degrees. We will use this image database to test the effectiveness of Algorithm 4 for final photo selection with visual features. It is worth mentioning that Algorithm 4 is not limited to the COIL-100 database or SIFT features. It would be interesting future work to test Algorithm 4 with other appropriate photo datasets and visual features.

B. RESULTS OF INITIAL SELECTION WITH METADATA

According to the settings in Section V-A, we evaluate the initial selection methods introduced in Section IV-A with different datasets. To investigate their performance extensively, we test each method for multiple runs. In each run, each method takes a new dataset of 500 samples as its input. Fig. 2 shows the total utility achieved by the photo selection with SmartPhoto [3] and the extended algorithm SmartPhoto$^+$ in Algorithm 1. Clearly, SmartPhoto$^+$ significantly improves the total utility by selecting photos according to multiple features instead of just the coverage aspect.

The photo selection approach PicPick [4] aims to maximize the total coverage of the selected photos by using a PTree to group the photo set. As shown in Fig. 3, PicPick$^+$ in Algorithm 2 slightly enhances this goal. To generate Fig. 3, the original PicPick is slightly adapted to be comparable to PicPick+ and meet the limit on the number of selected pictures by stopping selection after the limit is reached. The average coverage is improved by 15.32% with the synthetic metadata, while it is improved by 10.34% with the dataset of real locations. These results demonstrate that our extension does well in keeping the clustering structure when pruning the branches of the PTree.

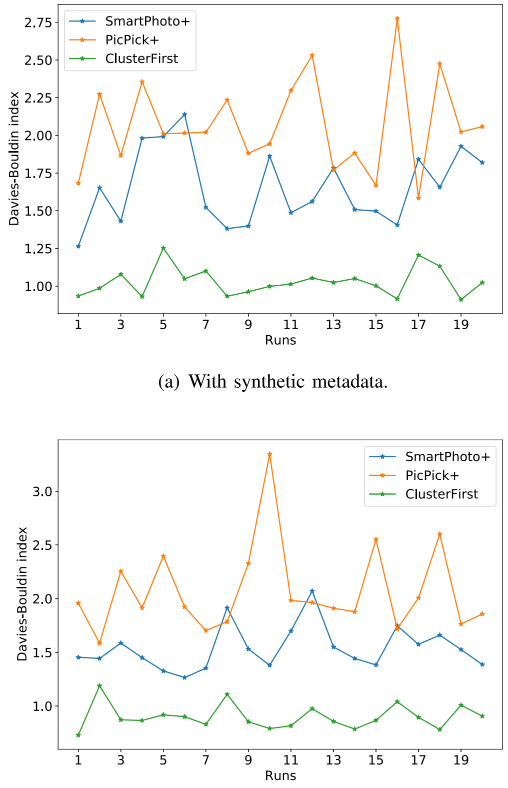
Figs. 4–6 compare the three photo selection approaches discussed in Section IV in terms of the three clustering metrics given in Section III-D. For the approach ClusterFirst, we use agglomerative clustering in Algorithm 3. Fig. 4 shows the silhouette coefficients of the three approaches. As seen, if we select photos using the agglomerative clustering algorithm in Algorithm 3, the resulting clustering can achieve the highest silhouette coefficients. It is known that the silhouette coefficient is larger when the clusters are denser and better separated. Similarly, it is seen in Fig. 5 that the Davies-Bouldin indices of ClusterFirst are the lowest among the three approaches. Different from the silhouette coefficient, a lower Davies-Bouldin index indicates a better clustering result.



(a) With synthetic metadata.

(b) With real location data.

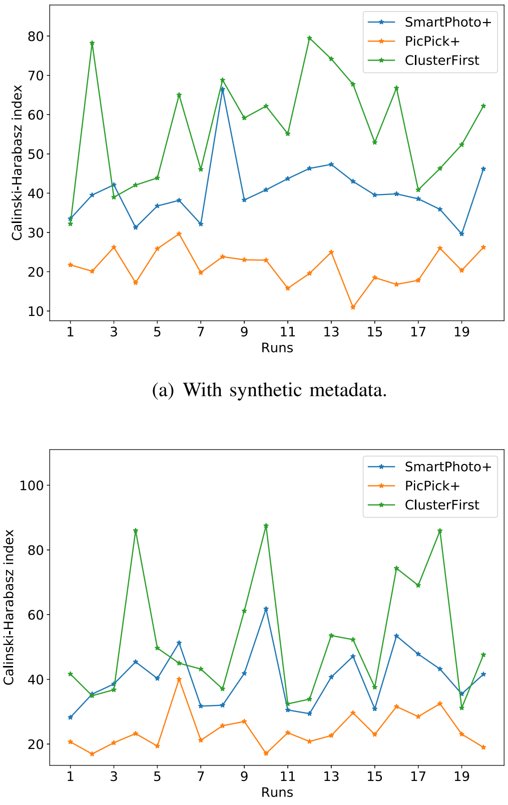
FIGURE 4. Silhouette coefficient.



(a) With synthetic metadata.

(b) With real location data.

FIGURE 5. Davies-Bouldin index.



(a) With synthetic metadata.

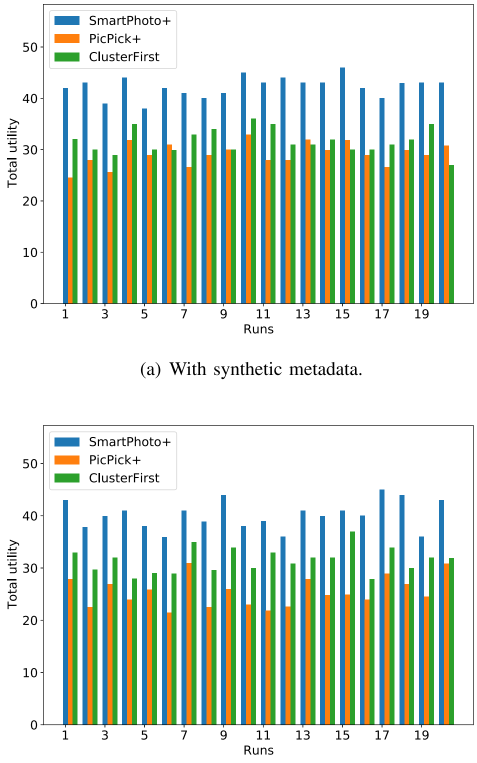
(b) With real location data.

FIGURE 6. Calinski-Harabasz index.

Fig. 6 shows the Calinski-Harabasz index of the three photo selection approaches. It is observed that ClusterFirst achieves the highest Calinski-Harabasz indices. Similar to the silhouette coefficient, a higher Calinski-Harabasz index indicates a better clustering result. With the synthetic metadata, the average Calinski-Harabasz index of ClusterFirst is 40.22% higher than that of SmartPhoto$^+$ and 1.655 times higher than that of PicPick$^+$. With the dataset of real locations, SmartPhoto$^+$ performs closer to ClusterFirst in some cases. However, the average Calinski-Harabasz index of ClusterFirst is still 29.01% higher than that of SmartPhoto$^+$. Also, both ClusterFirst and SmartPhoto$^+$ significantly outperform PicPick$^+$ in terms of the Calinski-Harabasz index.

Figs. 7–8 compare the three photo selection approaches in terms of other performance metrics such as total utility and total coverage. Since SmartPhoto and its extension SmartPhoto$^+$ aim to maximize the total utility, it is not surprising to see in Fig. 7 that SmartPhoto$^+$ achieves the highest total utility. ClusterFirst with agglomerative clustering comes next, while the total utility of PicPick$^+$ is the lowest. This is because PicPick$^+$ selects photos to maximize the number of other photos covered by the selected ones.

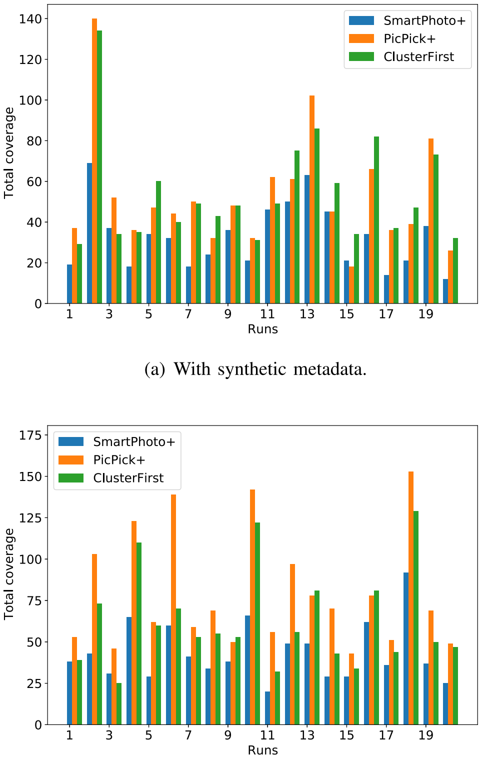
As shown in Fig. 8, the total coverage of PicPick$^+$ is the highest on average among the three approaches. We can see in Fig. 8(a) that, with the synthetic dataset, the total coverage of PicPick$^+$ is 61.66% higher than that of SmartPhoto$^+$,



(a) With synthetic metadata.

(b) With real location data.

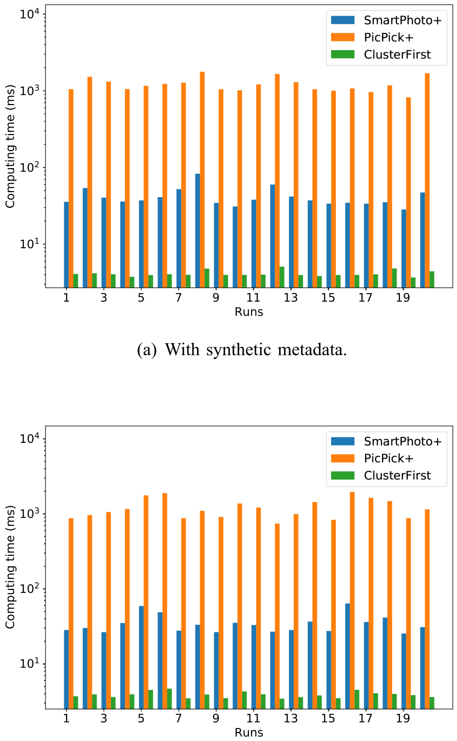
FIGURE 7. **Total utility.**



(a) With synthetic metadata.

(b) With real location data.

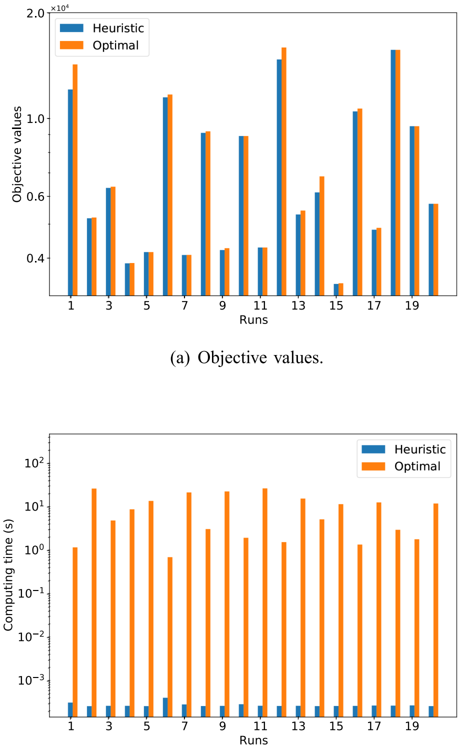
FIGURE 8. **Total coverage.**



(a) With synthetic metadata.

(b) With real location data.

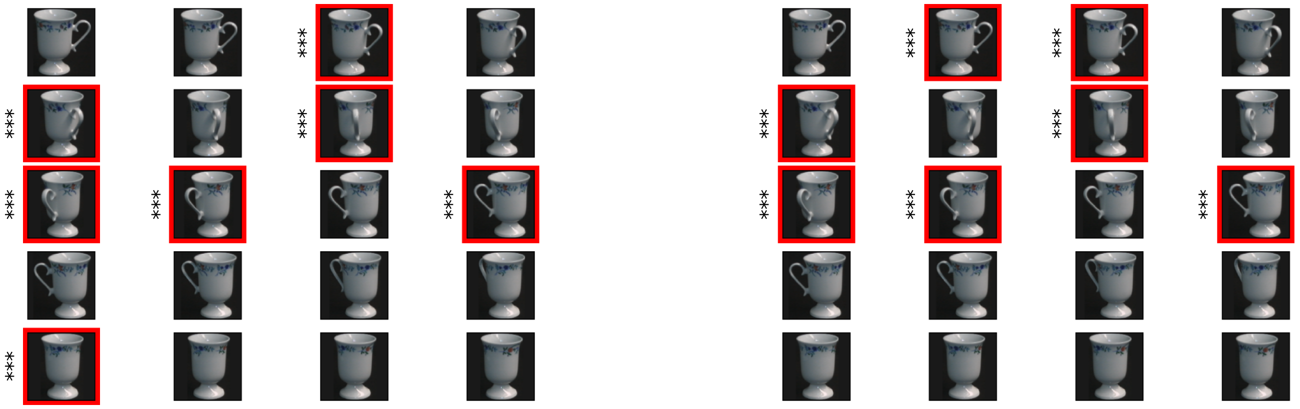
FIGURE 9. **Computing time.**



(a) Objective values.

(b) Computing time.

FIGURE 10. **The optimal solution vs. the heuristic solution for final photo selection.**



(a) The optimal solution.

(b) The heuristic solution.

FIGURE 11. Final photo selection results based on SIFT features.

but it performs very closely to ClusterFirst. Similar trends are observed in Fig. 8(b). PicPick$^+$ performs the best since it targets at maximizing the total coverage. ClusterFirst with agglomerative clustering achieves total coverage that is slightly lower that of PicPick$^+$, but it performs better than SmartPhoto$^+$.

As the edge servers in MEC usually have limited computing resources, the computational cost of the data selection approaches is an important factor to ensure feasible implementation. In Fig. 9, we further evaluate the computing time (in milliseconds) of PicPick$^+$, SmartPhoto$^+$, and ClusterFirst. They are implemented in Python and run on a Mac mini station with a processor of 3.2 GHz 6-core Intel Core i7 and memory of 32 GB. As seen, all three methods can run on the order of seconds. ClusterFirst is the fastest with computing time of several milliseconds. PicPick$^+$ is the slowest but the average computing time of all runs is still less than 2 seconds. One reason for the longer time is that PicPick$^+$ needs to constantly maintain a complex data structure, PTree, which organizes the picture set according to their features. The results in Fig. 9 demonstrate that these three methods are computationally efficient and it is viable to deploy them on the edge servers in MEC.

C. RESULTS OF FINAL SELECTION WITH VISUAL DATA

As mentioned in Section IV, we take a three-phase approach to select a limited number of photos. After the pre-screening with the validity constraints and the initial photo selection based on the context metadata, we need to further process the visual content data of the pre-selected photos and choose a given number of the best ones among them. In Section IV-B, we give Algorithm 4 for the final photo selection based on SIFT. First, the SIFT features are extracted to evaluate the pairwise distances of the pre-selected photos. Then, we solve the optimization problem in (6) with an ILP solver or a heuristic approach.

Using the COIL-100 image library, we compare the heuristic solution with the optimal solution obtained by the ILP solver CVXOPT [32] with the GLPK extension [27]. Here, we randomly choose γB photos for an object and select B most distinct photos among them using the optimal solution and the heuristic solution. Similar to the above experiments, we compare these solutions in multiple runs. In each run, we randomly select a new subset of photos from the image library and apply the selection solutions over them. Fig. 10 compares the objective values achieved by the two solutions and their computing time. As seen in Fig. 10(a), the heuristic solution performs very closely to the optimal solution. The average approximation ratio of the objective value achieved by the optimal solution to that of the heuristic solution is 0.968. Nonetheless, Fig. 10(b) shows that the heuristic solution takes significantly less computing than the optimal solution.

To further examine the differences between the optimal solution and the heuristic solution, Fig. 11 shows the photos selected by the two approaches for a case where the heuristic solution has an approximation ratio 0.9879. The selected photos are annotated with *** and framed by red borders. As seen, both approaches successfully choose the photos with fairly distinct shoot angles. These photos are the representative ones for the whole set. Specifically, the heuristic approach selects two photos in the first row with close views. In contrast, the optimal solution selects one photo in the last row with a side view, which is clearly a better choice. However, the heuristic approach performs closely to the optimal solution overall, and it is much faster. Especially, the ILP solver becomes quite slow when there are more than 30 candidates in the photo set.

VI. CONCLUSION

In this paper, we study the data selection problem in VCS. A phase-by-phase hybrid framework is considered, which first filters collected pictures based on the metadata and then selects the final pictures using the content features. Particularly, we extend SmartPhoto [3] and PicPick [4] to be applicable to the hybrid framework. We also consider a benchmark approach with a standard clustering algorithm for comprison. In addition, we evaluate different selection approaches using adapted clustering indices as well as traditional metrics such as total utility and coverage.

Extensive simulations are conducted with both synthetic and real datasets. The results show that, the extended algorithm SmartPhoto$^+$ performs significantly better than the original algorithm SmartPhoto in terms of total utility. PicPick$^+$ slightly improves the total coverage in comparison with PicPick when the number of selected pictures is constrained. Among the three algorithms (SmartPhoto$^+$, PicPick$^+$, and ClusterFirst), SmartPhoto$^+$ achieves the highest total utility, while PicPick$^+$ performs the best in terms of total coverage but fairly close to ClusterFirst. Regarding the clustering indices, ClusterFirst shows the best results. In practice, we can choose an appropriate solution among these candidates according to specific application needs and target performance.

REFERENCES

[1] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, “Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey,” *IEEE Commun. Surv. Tuts.*, vol. 18, no. 4, pp. 2546–2590, Oct.–Dec. 2016.

[2] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu, “The emergence of visual crowdsensing: Challenges and opportunities,” *IEEE Commun. Surv. Tuts.*, vol. 19, no. 4, pp. 2526–2543, Oct.–Dec. 2017.

[3] Y. Wu, Y. Wang, W. Hu, and G. Cao, “SmartPhoto: A resource-aware crowdsourcing approach for image sensing with smartphones,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1249–1263, May 2016.

[4] B. Guo, H. Chen, Z. Yu, X. Xie, and D. Zhang, “PicPick: A generic data selection framework for mobile crowd photography,” *Pers. Ubiquitous Comput.*, vol. 20, pp. 325–335, 2016.

[5] B. Guo, H. Chen, Q. Han, Z. Yu, D. Zhang, and Y. Wang, “Worker-contributed data utility measurement for visual crowdsensing systems,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2379–2391, Aug. 2017.

[6] H. Wang, M. Uddin, G.-J. Qi, T. Huang, T. Abdelzaher, and G. Cao, “PhotoNet: A similarity-aware image delivery service for situation awareness,” in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2011, pp. 135–136.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[8] H. Chen, B. Guo, Z. Yu, L. Chen, and X. Ma, “A generic framework for constraint-driven data selection in mobile crowd photographing,” *IEEE Internet Things J.*, vol. 4, no. 1, pp. 284–296, Feb. 2017.

[9] T. Zhou, B. Xiao, Z. Cai, and M. Xu, “A utility model for photo selection in mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 48–62, Jan. 2021.

[10] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, “iCrowd: Near-optimal task allocation for piggyback crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.

[11] M. Zhang et al., “Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7698–7707, Sep. 2016.

[12] J. Wang et al., “Multi-task allocation in mobile crowd sensing with individual task quality assurance,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2101–2113, Sep. 2018.

[13] B. Desgraupes, “Clustering indices,” Lab Modal’X, University Paris Ouest, 2017, pp. 1–34.

[14] P. Cheng et al., “Reliable diversity-based spatial crowdsourcing by moving workers,” *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1022–1033, 2015.

[15] H. Chen, B. Guo, and Z. Yu, “CooperSense: A cooperative and selective picture forwarding framework based on tree fusion,” *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 4, pp. 1–13, 2016.

[16] C. Xu and W. Song, “Efficient data uploading for mobile crowdsensing via team collaborating and matching,” *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 645–654, Mar. 2022.

[17] T. Zhou, B. Xiao, Z. Cai, M. Xu, and X. Liu, “From uncertain photos to certain coverage: A novel photo selection approach to mobile crowdsensing,” in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1979–1987.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[19] E. Karami, S. Prasad, and M. S. Shehata, “Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images,” 2017, *arXiv:1710.02726*.

[20] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[21] Scikit-learn developers, “2.3 Clustering,” Accessed: Aug. 2022. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html

[22] Yufeng, “Three performance evaluation metrics of clustering when ground truth labels are not available,” Accessed: Aug. 2022. [Online]. Available: https://towardsdatascience.com/three-performance-evaluation-metricsof-clustering-when-ground-truth-labels-are-not-available-ee08cb3ff4fb

[23] D. S. Hochbaum, Ed., *Approximation Algorithms for NP-Hard Problems*. Boston, MA, USA: PWS Publishing Company, 1996.

[24] Z. Liu, B. Ferry, and S. Lacasse, “A scalable deep neural network to detect low quality images without a reference,” in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 1–7.

[25] S. Bosse, D. Maniry, T. Wiegand, and W. Samek, “A deep neural network for image quality assessment,” in *Proc. Int. Conf. Image Process.*, 2016, pp. 3773–3777.

[26] Gurobi Optimization, “ The Gurobi optimizer,” 2008. [Online]. Available: https://www.gurobi.com/products/gurobi-optimizer/

[27] GNU Project, “ GNU linear programming kit (GLPK),” 2000. [Online]. Available: https://www.gnu.org/software/glpk/

[28] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, “Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[29] Foursquare Labs, “ Foursquare city guide,” 2009. [Online]. Available: https://foursquare.com

[30] C. Xu, “Intelligent data collection and data dissemination application design via mobile crowd sensing,” Univ. New Brunswick, Fredericton, Canada, Tech. Rep., 2019.

[31] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (COIL-100),” Columbia University, Tech. Rep. CUCS-006-96, 1996.

[32] M. Andersen and L. Vandenberghe, “ CVXOPT: Python software for convex optimization,” 2004. [Online]. Available: https://cvxopt.org

WEI SONG (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2007. In 2009, she joined the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, where she is currently working as a Full Professor. Her research interests include Internet of Things, mobile edge computing, mobile crowdsensing, and device-to-device communications. She was the recipient of Best Paper Award from the 2018 IEEE ICC, 2014 UNB Merit Award, Best Student Paper Award from the 2013 IEEE CCNC, Top 10% Award from the 2009 IEEE MMSP, and Best Paper Award from the 2007 IEEE WCNC. She has been the Chair of the Joint Computer and Communications Chapter of IEEE New Brunswick Section from 2014 to 2020. She Co-chaired tracks/symposiums for IEEE VTC Fall 2010, IWCMC 2011, IEEE GLOBECOM 2011, IEEE ICC 2014, IEEE VTC Fall 2016, and IEEE VTC Fall 2017.