IEEE Access
Multidisciplinary ⁞ Rapid Review ⁞ Open Access Journal

Received January 1, 2020, accepted January 17, 2020, date of publication January 23, 2020, date of current version February 4, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2968982*

# Reverse Spatial Visual Top-*k* Query

**LEI ZHU[1], JIAYU SONG[1], WEIREN YU[2,3], CHENGYUAN ZHANG[1], HAO YU[1], AND ZUPING ZHANG[1]**

[1]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[2]School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China
[3]Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.

Corresponding authors: Weiren Yu (ywr0708@hotmail.com) and Chengyuan Zhang (cyzhang@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702560, Grant 61379109, Grant 61836016, and Grant 61972203, in part by the Science and Technology Plan of Hunan Province under Project 2018JJ3691 and Project 2016JC2011, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190442, and in part by the Research and Innovation Project of Central South University Graduate Students under Grant 2018zzts177.

**ABSTRACT** With the wide application of mobile Internet techniques an location-based services (LBS), massive multimedia data with geo-tags has been generated and collected. In this paper, we investigate a novel type of spatial query problem, named reverse spatial visual top-*k* query (RSVQ$_k$) that aims to retrieve a set of geo-images that have the query as one of the most relevant geo-images in both geographical proximity and visual similarity. Existing approaches for reverse top-*k* queries are not suitable to address this problem because they cannot effectively process unstructured data, such as image. To this end, firstly we propose the definition of RSVQ$_k$ problem and introduce the similarity measurement. A novel hybrid index, named VR$^2$-Tree is designed, which is a combination of visual representation of geo-image and R-Tree. Besides, an extension of VR$^2$-Tree, called CVR$^2$-Tree is introduced and then we discuss the calculation of lower/upper bound, and then propose the optimization technique via CVR$^2$-Tree for further pruning. In addition, a search algorithm named RSVQ$_k$ algorithm is developed to support the efficient RSVQ$_k$ query. Comprehensive experiments are conducted on four geo-image datasets, and the results illustrate that our approach can address the RSVQ$_k$ problem effectively and efficiently.

**INDEX TERMS** Geo-image, reverse top-*k* query, spatial visual query, hybrid index.

## I. INTRODUCTION

With the wide application of mobile Internet techniques and location-based services (LBS), massive multimedia data with geo-tags (geo-multimedia for short) has been generated and collected by smartphones and tablets with local sensors, and then uploaded and stored on the Internet. On the one hand, the multimedia sharing platform and online social networking provide geo-multimedia storage and sharing service. For example, more than 95 million photos with location information captured by smartphones and digital cameras are stored on Flickr,[1] which is one of the largest picture sharing platforms. more than 140 million Twitter[2] users post 400 million tweets in the form of text and image with geo-location information (referred as geo-text and geo-image). In China, lots of users of WeChat,[3] the most popular mobile application, share texts, images and short videos with geo-tags every day. On the other hand, geo-multimedia data are used in many location-based services. For instance, Dianping[4] provides the rating and review services for finding restaurant, hotel, gym, cinema, etc. via sharing the geo-texts and geo-images uploaded by users. Another LBS application is Foursquare,[5] which helps users to share the places visited and find the best places nearby via geo-multimedia data. These geo-multimedia data is a fusion of multimedia content [1], [2] and geo-location information [3], which enables queries consider geographical proximity and multimedia content similarity simultaneously.

Spatial keyword query [4] is one of the significant problems that has attracted much attention in the spatial database and information retrieval community. This query aims to find

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

[1]http://www.flickr.com/
[2]http://www.twitter.com/
[3]https://web.wechat.com/
[4]https://www.dianping.com/
[5]https://foursquare.com/

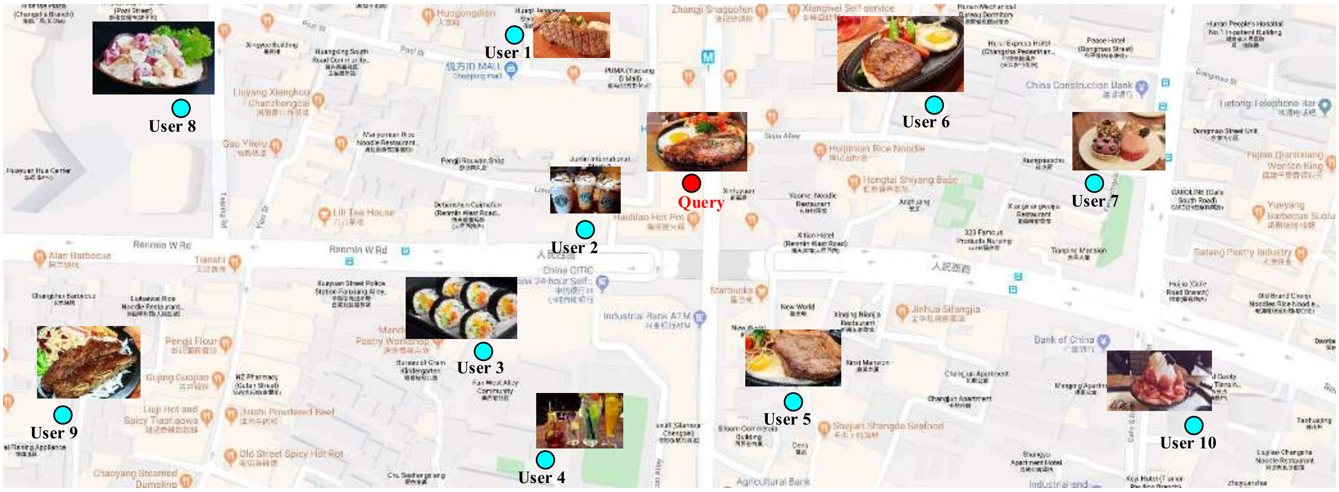


**FIGURE 1.** An example of reverse spatial visual top-*k* query.

spatial objects by taking into account both spatial proximity and relevance of keywords. Several types of spatial keyword search, i.e., collective spatial keyword query [5], *m*-closest keywords search [6], best keyword cover search [7], group top-*k* spatial keyword query [8] and etc., are studied deeply and applied widely in many scenarios to provide efficient spatial keyword query.

### A. MOTIVATION

Reverse spatial keyword query [9] is another important search problem, which is to find a set of geo-objects that have the query as one of the most relevant objects in both geographical proximity and textual similarity. Many researches [10]–[13] propose efficient algorithms to speed up the reverse search on Euclidean space and road network space. However, previous works only focus on keyword search, which are suitable for unstructured data, such as geo-image. In other words, these techniques cannot be applied directly to the reverse spatial query for geo-multimedia data. To this end, this paper consider geo-image that is the most common type of geo-multimedia. Thus, we propose a novel type of reverse top-*k* query, named *reverse spatial visual top-k query* (RSVQ$_k$ for short), which takes into account both geo-location proximity and visual similarity between images. In other words, users can submit a reverse query with geo-images, rather than keywords. To the best of our knowledge, this is the first time to investigate RSVQ$_k$ problem. To introduce this problem more intuitively, we provide an example of reverse spatial visual top-*k* query as follows:

*Example 1:* As shown in Fig. 1, a manager of a steak house wants to know the consumer preferences of people nearby so as to carry out more accurate advertising. She submits a reverse spatial visual top-*k* query by taking a picture of steak with a smartphone in this steak house. The system will return the users who have this steak house as one of the *k* most desirable restaurants in both aspects of geographical proximity and the visual similarity between their posed images and the query image.

### B. OUR METHOD

To overcome this challenge, firstly, this paper defines reverse spatial visual top-*k* query in formal, and introduces the relevant notions, i.e., the geographical proximity measurement and visual similarity measurement. As far as we know, this is the first time to propose the definition of RSVQ$_k$ and no existing approach has been proposed for this problem. Thus, a baseline that uses R-Tree and the threshold algorithm [14] is proposed. To organizing the geo-image data more efficiently, we careful design a novel hybrid index, named VR$^2$-Tree, which is a integration of the visual representation of geo-images and R-Tree. The visual representation of an image in this work is a vector of visual words. Two operations of visual words vector, namely Weight OR and Weight AND are proposed to support the generation of the non-leaf nodes of VR$^2$-Tree. Besides, an extension of VR$^2$-Tree, named CVR$^2$-Tree is developed to enhance the pruning power by its specific entry in tree node, namely CEntry set. Furthermore, we discuss the calculation of lower bound and upper bound via CVR$^2$-Tree, and then introduce the optimization technique via CVR$^2$-Tree to tighter the bounds. In addition, the CVR$^2$-Tree based query processing algorithm with the optimization is introduced.

### C. CONTRIBUTIONS

The main contributions of this work are summarized as follows:

- We propose the definition of reverse spatial visual top-*k* query and the relevant notions. Besides, a baseline for reverse spatial visual search is introduced. To the best of our knowledge, this work is the first time to study RSVQ$_k$ problem.
- We present a novel hybrid index, named VR$^2$-Tree which is a combination of visual representations of

geo-images and R-Tree. In addition, an extension of $VR^2$-Tree, called $CVR^2$-Tree is designed, which can further improve the pruning power during the reverse search.

- We carefully develop the efficiency $RSVQ_k$ algorithm, which utilizes the optimization technique via $CVR^2$-Tree to enhance the search performance significantly.
- We have conducted extensive performance evaluation on four geo-image datasets. Experimental results demonstrate that ths proposed approach has really high performance.

### D. ROADMAP

In the remainder of this paper, we review the previous studies about this work in Section II. In Section III, we propose the definition of reverse spatial visual top-*k* query and the related notions. Besides, a baseline is introduced in this section. In Section IV, a novel hybrid index, named $VR^2$-Tree and its extension, i.e. $CVR^2$-Tree are proposed. Furthermore, an efficient reverse spatial visual search algorithm named RSVQ*k* is carefully designed. In Section V, we evaluate the proposed algorithms on four geo-image datasets. Finally, we conclude this paper in Section VI.

## II. RELATED WORK

In this section, we review the previous studies of image retrieval and collective spatial keyword query, which are related to our work. To the best of our knowledge, we are the first to study the problem of collective geo-image query.

### A. IMAGE RETRIEVAL

Image retrieval is one of the classical problems in the community of multimedia and computer vision, and it can be applied in versatile big data applications [15]–[23]. Lots of researches have been proposed to combat this challenge. As two powerful visual feature representation tools, Scale-Invariant Feature Transform (SIFT) [24], [25] and Bag-of-Visual-Word (BoVW) [26] are widely utilized. For example, Ke *et al.* [27] proposed an effective PCA-based local feature representation method called PCA-SIFT to improve the accuracy and efficiency. Mortensen *et al.* [28] proposed to augment the original SIFT descriptor by combining SIFT feature with a global context vector to enhance the matching rate. Li and Ma [29] improved SIFT descriptor by integrating color and global information which provides powerfully distinguishable information. Dimitrovski *et al.* [30] improved BoVW model by using predictive clustering trees to construct codebook to reduce the number of local descriptors.

More recently, with the rise of deep learning [31]–[33], lots of researchers employed more powerful tools such as CNN [34], RNN [35] and LSTM [36] to greatly hoist the image retrieval accuracy [37], [38]. In 2012, AlexNet [39] markedly improved the image classification accuracy and won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Matsuo *et al.* [40] proposed a CNN-based style vector that is transformed from style matrix with PCA dimension reduction. Gordo *et al.* [41] proposed a CNN-based global fixed-length representation for image retrieval, which is generated by a ranking framework. Tan *et al.* [42] utilized different CNN models to extract the multiple visual features that are fused into weighted average feature. Liu *et al.* [17] introduced a method that fuses high-level features from CNN and low-level features to generate two-layer codebook features. Seddati *et al.* [43] combined multi-scale and multi-layer feature extraction from improved RMAC approaches, which generates short descriptors and get better performance without the need of CNN fine tuning. Yang *et al.* [44] introduced a method in which a dynamic match kernel is constructed by calculating the matching thresholds between query and candidate.

It is obvious that the deep learning based methods have much better performance than the traditional hand-crafted feature based methods. In our previous works [45], [46], we proposed to combine the spatial search techniques and visual feature representations to solve geo-multimedia retrieval problem. However, as far as we know there is no existing image retrieval approach that is suitable to address the reverse spatial visual query (RSVQ) problem. In this work, we attempt to design efficient index structure and algorithm for RSVQ problem.

### B. SPATIAL KEYWORD QUERY

Spatial keyword query [47], [48] is a significant problem in the community of spatial database [49]–[51], which is well studied by researchers in recent years. It aims to returns spatial-textual objects that are spatially and textually relevant to the query. Several spatial indexing structures such as R-Tree [52], $R^*$-Tree [53], IR-Tree [54], [58], $KR^*$-Tree [55], IL-Quadtree [56], etc. have been proposed to improve the spatial keyword search effectively.

Felipe *et al.* [57] proposed to address the top-*k* spatial keyword queries by using a novel index called Information Retrieval R-Tree ($IR^2$-Tree) that is a combination of R-Tree and superimposed text signatures. Cong *et al.* [58] introduced a new indexing framework, in which the inverted file is employed for text retrieval and R-tree for spatial proximity search. Rocha-Junior *et al.* [59] proposed a novel index named Spatial Inverted Index (S2I) which maps each distinct term to a set of objects containing the term. Zhang *et al.* [60] developed $I^3$ that is an integrated inverted index with quadtree to partition the data space into cells in a hierarchical manner. In another work of them [61], they modeled the top-*k* distance-sensitive spatial keyword query as top-*k* aggregation problem, and then an extension of CA algorithm, called Rank-aware CA algorithm, to enhance the search.

Unfortunately, These researches whether in European space or road network space can only be applied to structured data, e.g., keywords. That means they are not suitable to cope with spatial unstructured data, such as geo-image. To the best of our knowledge, this paper is the first time to develop effective and efficient technique for the geo-image search task.

### C. REVERSE QUERY PROCESSING

Reverse query [11], [62], [63] is another significant problem in the area of spatial-textual search, which is from the perspective of point of interest (POI), e.g. restaurant, supermarket, store, tourist attraction, etc, rather than users. More specifically, it aims to retrieve the users for which the query objects is one of the most preferences, such as geographical proximity [64]–[67]. Reverse query can be applied in lots of applications, e.g., advertising, recommendation, marketing, etc.

Many researches have been proposed to combat this challenge in the last decade. Vlachou *et al.* [64] proposed the reverse top-*k* query and introduced two versions, namely monochromatic and bi-chromatic. In their another work [68], they proposed distance-based reverse top-*k* query problem which can be applied in the mobile environment. For the reverse *k* nearest neighbor (R*k*NN) problem, Cheema *et al.* [69] proposed a novel notion, named influence zone, which is the area such that every point inside this area is the results of R*k*NN query and every point outside it is not the results. Yu *et al.* [70] studied the reverse top-*k* search by using random walk with restart in large graphs. In road network space, Wang *et al.* [71] investigated continuous monitoring of RkNN queries. They utilized the influence zone to boost the search.

Not only the proximity of space distance, the textual similarity is considered into the reverse query. For example, Lin *et al.* [9] proposed the reverse keyword search for spatio-textual top-*k* queries (RSTQ) at the first time and developed a novel hybrid index, called KcR-tree, to store and summarize the spatial and textual information of objects. Yang *et al.* [11] proposed to extend half-space-based pruning technique to solve the spatial reverse top-*k* queries and introduced a novel regions-based pruning algorithm according to SLICE [72] that is a regions-based pruning algorithm for reverse *k* nearest neighbors queries to improve the efficiency. Instead in the Euclidean space, Luo *et al.* [73] investigated reverse spatial and textual *k* nearest neighbor queries on road networks. Besides, they proposed several spatial keyword pruning techniques to speed up the search. Gao *et al.* [10] introduced another novel query paradigm, called reverse top-*k* Boolean spatial keyword (RkBSK) retrieval on Road Networks that considers both spatial and textual information. To boost the system performance significantly, they developed a new data structure named count tree to overcome the drawback of the count list.

However, these solutions for reverse queries cannot be extended to the geo-image query problem since they are not suitable for unstructured data such as image. To combat this limit, in this work we propose to address the reverse spatial visual top-*k* query problem that takes into account both visual similarity and geographical proximity simultaneously. To the best of our knowledge, we are the first to propose this query paradigm and try to solve it effectively and efficiently.

## III. PRELIMINARY

In this section, for the first time, we formulate the definition of reverse spatial visual top-*k* query problem and introduce the relevant notions. Then we propose the baseline to combat this challenge. Table 1 summarizes the notations frequently used throughout this paper to facilitate the discussion.

**TABLE 1.** The summary of notations.

| Notation | Definition |
|---|---|
| $\mathcal{I}$ | a geo-image dataset |
| $I$ | a geo-image |
| $I.\boldsymbol{\lambda}$ | the geo-location descriptor of $I$ |
| $X$ | the value of longitude |
| $Y$ | the value of latitude |
| $I.\boldsymbol{\nu}$ | the visual descriptor of $I$ |
| $Q$ | the reverse spatial visual top-$k$ query |
| $\mu$ | the balance parameter in similarity measurement |
| $W(\cdot)$ | the weight function of visual word |
| $\bigotimes$ | the weight AND operator |
| $\bigoplus$ | the weight OR operator |
| $\mathcal{C}_k$ | the $k$-th cluster |
| $\mathcal{S}_C$ | the CEntry set |
| $\mathcal{E}_C$ | a CEntry |
| $C_{id}$ | the id of a cluster |
| $\biguplus$ | the CEntry set sum operator |
| $\mathfrak{T}$ | a CVR$^2$-Tree |
| $\mathcal{T}$ | a tuple in a CVR$^2$-Tree |
| $\lfloor\mathcal{T}\rfloor$ | the lower bound of similarity between $\mathcal{T}$ and $k$-th most similar geo-image |
| $\lceil\mathcal{T}\rceil$ | the upper bound of similarity between $\mathcal{T}$ and $k$-th most similar geo-image |
| $\Psi_L$ | a lower bound determinant queue |
| $\Psi_U$ | a upper bound determinant queue |

### A. PROBLEM DEFINITION

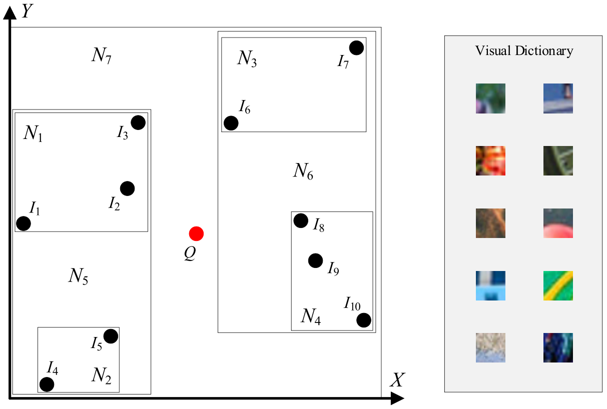
Before defining the reverse spatial visual top-*k* query problem, we introduce the notion of geo-image that contains two aspects of information, i.e., geo-location and visual content.

Let $\mathcal{I} = \left\{I_1, I_2, \ldots I_{|\mathcal{I}|}\right\}$ be a geo-image dataset. Each geo-image $I \in \mathcal{I}$ is represented by a tuple $\langle I.\boldsymbol{\lambda}, I.\boldsymbol{\nu}\rangle$, where $I.\boldsymbol{\lambda}$ is the geo-location descriptor that is a 2-dimensional vector to represent the geographical information in the form of longitude $X$ and latitude $Y$, i.e., $I.\boldsymbol{\lambda} = (X, Y)$. $I.\boldsymbol{\nu}$ is the visual descriptor which is a $\gamma$-dimensional vector to represent the visual features of the image, i.e., $I.\boldsymbol{\nu} = \left(\nu^{(1)}, \nu^{(2)}, \ldots \nu^{(\gamma)}\right)$. In this paper, we employ BoVW [26] model to construct the visual descriptor, thus each item $\nu$ represents a visual word.

*Definition 1 (Reverse Spatial Visual Top-k Query):* Given a geo-image dataset $\mathcal{I}$ and a query $Q = \langle Q.\boldsymbol{\lambda}, Q.\boldsymbol{\nu}\rangle$. A reverse spatial visual top-*k* query (RSVQ$_k$) aims to retrieve all the geo-images in $\mathcal{I}$ that consider the query $Q$ as one of the top-*k* most relevant geo-images in both aspects of geo-location and visual content. Formally, it is described as follows:

$$RSVQ_k(\mathcal{I}, Q, k) = \{I | Q \in SVQ_k(\mathcal{I}, I, k), I \in \mathcal{I}\} \qquad (1)$$

where $SVQ_k(\mathcal{I}, I, k)$ represents the spatial visual top-$k$ query that aims to return $k$ most similar geo-images by a query $I$ considering geographical proximity and visual similarity



| Geo-Image | Geo-Location | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_1$ | (0,10) | 5.0 | 2.0 | 7.0 | 8.0 | 2.5 | 0.0 | 0.0 | 1.5 | 0.0 | 0.0 |
| $I_2$ | (6,13) | 7.0 | 9.0 | 9.5 | 2.0 | 6.5 | 0.0 | 0.0 | 1.5 | 0.0 | 0.0 |
| $I_3$ | (7,17) | 8.0 | 3.5 | 0.0 | 9.0 | 0.0 | 0.0 | 6.5 | 0.0 | 8.0 | 1.0 |
| $I_4$ | (2,1) | 4.5 | 2.0 | 0.0 | 1.5 | 0.0 | 3.0 | 2.0 | 1.0 | 9.0 | 5.5 |
| $I_5$ | (5,3) | 5.0 | 8.5 | 8.5 | 9.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.5 |
| $I_6$ | (12,17) | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 8.0 | 8.0 | 9.0 | 8.0 |
| $I_7$ | (19,19) | 0.0 | 2.0 | 0.0 | 0.0 | 1.0 | 0.0 | 9.5 | 7.0 | 9.5 | 9.0 |
| $I_8$ | (16,10) | 2.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.5 | 9.0 | 8.0 | 8.0 |
| $I_9$ | (17,8) | 3.0 | 1.5 | 1.0 | 0.0 | 0.0 | 1.0 | 8.5 | 7.5 | 9.0 | 8.5 |
| $I_{10}$ | (19,4) | 4.5 | 3.5 | 0.0 | 0.0 | 8.0 | 8.0 | 5.5 | 0.0 | 0.0 | 0.0 |
| $Q$ | (9,9) | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 9.0 | 7.5 | 8.5 |

**FIGURE 2. An example of reverse spatial visual top-*k* query (RSVQ$_k$). There are ten geo-images $I_1, I_2, \ldots I_{10}$, and a query *Q* containing ten different visual words in this example. The left is the spatial distribution of these ten geo-images. The table on the right demonstrates the details of them: the geo-location descriptor and the visual descriptor.**

simultaneously, formulated as follows:

$$\begin{aligned} &SVQ_k(\mathcal{I}, I, k) \\ &\quad = \left\{\hat{I} | Sim(\hat{I}, I) \geq Sim(I', I), \forall \hat{I}, I' \in \mathcal{I}\right\}, \\ &\qquad |SVQ_k(\mathcal{I}, I, k)| \leq k \end{aligned} \tag{2}$$

where $Sim(\hat{I}, I)$ is the similarity function to measure both geographical proximity and visual similarity between $\hat{I}$ and $I$. Herein we define it in formal as follows:

$$\begin{aligned} Sim(\hat{I}, I) = \mu \times GeoSim(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda}) \\ +(1-\mu) \times VisSim(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v}) \end{aligned} \tag{3}$$

where $\mu \in [0, 1]$ is a parameter to balance the proportion between geographical proximity and visual similarity, i.e., $GeoSim(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda})$ and $VisSim(\hat{I}.\boldsymbol{v}, I\boldsymbol{v})$. If $\mu = 1$ (or $\mu = 0$), the query is just considering the geographical proximity (or visual similarity). In our solution, the users are allowed to set this parameter according to their query preferences.

In the next we formulate the definition of geographic proximity and visual similarity measurement and introduce how to implement $GeoSim(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda})$ and $VisSim(\hat{I}.\boldsymbol{v}, I\boldsymbol{v})$.

*Definition 2 (Geographical Proximity Measurement):* Given a geo-image dataset $\mathcal{I}$, $\forall I, \hat{I} \in \mathcal{I}$ are two geo-images. The geographical proximity between $\hat{I}$ and $I$ is measured by the following function:

$$GeoSim(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda}) = 1 - \frac{EucliDst(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda})}{MaxDst(\mathcal{I})} \tag{4}$$

where $EucliDst(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda})$ is the function to calculate the Euclidean distance between $\hat{I}.\boldsymbol{\lambda}$ and $I.\boldsymbol{\lambda}$, shown as follows:

$$\begin{aligned} &EucliDst(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda}) \\ &\quad = \sqrt{(\hat{I}.\boldsymbol{\lambda}.X - I.\boldsymbol{\lambda}.X)^2 + (\hat{I}.\boldsymbol{\lambda}.Y - I.\boldsymbol{\lambda}.Y)^2} \end{aligned} \tag{5}$$

The function $MaxDst(\mathcal{I})$ in Eq. 4 measures the maximum Euclidean distance between any two geo-locations in the dataset $\mathcal{I}$, which is to normalize the Euclidean distance into [0, 1], i.e.,

$$\begin{aligned} &MaxDst(\mathcal{I}) \\ &\quad = Max\left(\left\{EucliDst(\hat{I}.\boldsymbol{\lambda}, I.\boldsymbol{\lambda}) | \forall \hat{I}, I \in \mathcal{I}\right\}\right) \end{aligned} \tag{6}$$

where $Max(\cdot)$ is to return the largest element from the input collection.

*Definition 3 (Visual Similarity Measurement):* Given a geo-image dataset $\mathcal{I}$, $\forall I, \hat{I} \in \mathcal{I}$ are two geo-images. The visual similarity between these two geo-images is measured by the following function:

$$VisSim(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v}) = \frac{ExJacc(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v})}{MaxVisSim(\mathcal{I})} \tag{7}$$

where $ExJacc(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v})$ is the extended Jaccard distance measurement shown as following:

$$\begin{aligned} &ExJacc(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v}) \\ &= \frac{\sum_{i=1}^{\gamma} W(\hat{v}^{(i)}) \times W(v^{(i)})}{\sum_{i=1}^{\gamma} W(\hat{v}^{(i)})^2 + \sum_{i=1}^{\gamma} W(v^{(i)})^2 - \sum_{i=1}^{\gamma} W(\hat{v}^{(i)}) \times W(v^{(i)})} \end{aligned} \tag{8}$$

to simplify the description, herein we use $\hat{v}^{(i)}$ and $v^{(i)}$ to denote $i$-th visual word of $\hat{I}.\boldsymbol{v}$ and $I.\boldsymbol{v}$, i.e., $\hat{v}^{(i)} \in \hat{I}.\boldsymbol{v}$ and $v^{(i)} \in I.\boldsymbol{v}$. The function $W(\cdot)$ in Eq. 8 is to calculate the weight of visual word by TF-IDF [74]. Similar to the role of $MaxDst(\mathcal{I})$ in Eq. 4, the function $MaxVisSim(\mathcal{I})$ in Eq. 7 is to return the maximum visual similarity, i.e.,

$$\begin{aligned} &MaxVisSim(\mathcal{I}) \\ &\quad = Max\left(\left\{ExJacc(\hat{I}.\boldsymbol{v}, I.\boldsymbol{v}) | \forall \hat{I}, I \in \mathcal{I}\right\}\right) \end{aligned} \tag{9}$$

In the following, we give a simple example to present reverse RSVQ$_k$ problem and how to find the results by comparing to the conventional reverse top-*k* query.

*Example 2:* As shown in Fig. 2, there is an example to describe the reverse spatial visual top-*k* query (RSVQ$_k$) task. Ten geo-images, i.e., $I_1, I_2, \ldots I_{10}$ illustrated by black

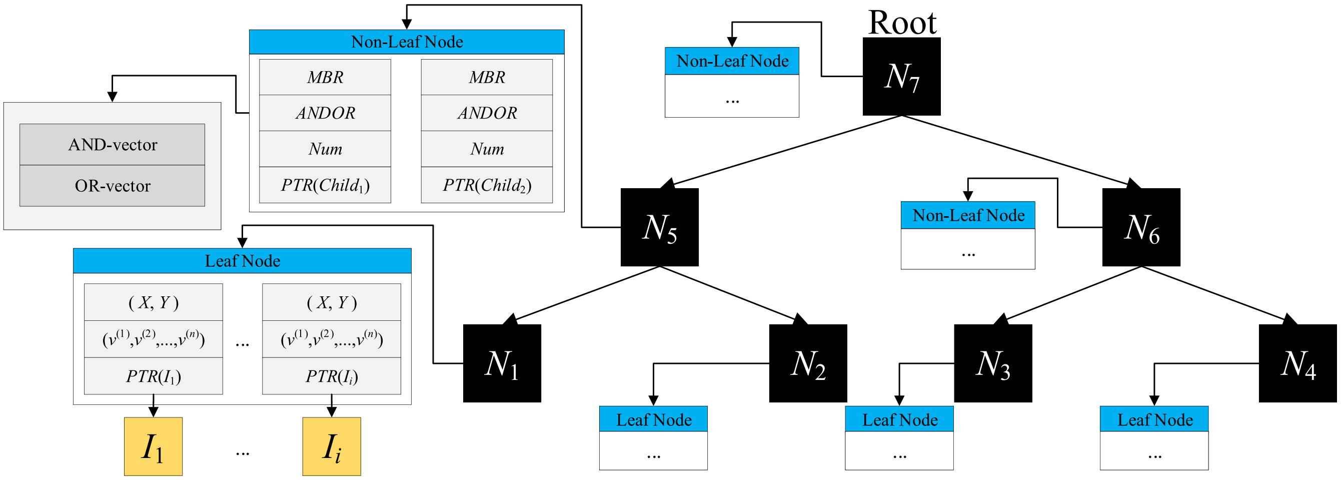


**FIGURE 3.** The structure of VR$^2$-Tree. It is a combination of visual representations and R-Tree. As described above, the visual representation of geo-image is the visual word vector and the geographical partition is implemented by MBR. Each tree node contains both geo-location information and visual content information.

spots, are distributed in a region represented by longitude $X$ and latitude $Y$. $N_1, N_2, \ldots N_7$ is the minimum bounding rectangles (MBRs) that is to describe the approximate location. The visual dictionary is the collection of visual words that are contained by these geo-images. The table on the right shows the geographical information and the weights of each visual words that are contained in each geo-image. Given a query $Q$ (the red spot), and $Q.\boldsymbol{\lambda} = (9, 9)$, $Q.\boldsymbol{v} = (1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 9.0, 9.0, 7.5, 8.5)$. For the conventional reverse top-$k$ query that consider only the geographical proximity (Euclidean distance), and let $k = 2$, the set of results is $\{I_2, I_5, I_8\}$. However, for the RSVQ$_k$, let $\mu = 0.5$, now the set of results is $\{I_8, I_6, I_9\}$ because $Q$ is more similar to $I_6$ and $I_9$ in the aspects of visual content.

### B. BASELINE INTRODUCTION

As far as we know, there is no study that focus on RSVQ$_k$ problem and no baselines have been proposed. Obviously, the existing reverse spatial textual search methods cannot be directly applied to RSVQ$_k$ since the necessity of visual representation and similarity measurement. According to Eq. 3, both geographical proximity and visual similarity should be considered simultaneously during the search. Thus, it is not feasible that perform reverse spatial search and reverse visual search separately and then combine the results of them to answer RSVQ$_k$ query.

In this work, we propose a baseline for RSVQ$_k$, named RSVQ$_k$-R. A pre-computation is processed to calculate the geographical proximity and visual similarity between the query $Q$ and all the geo-images in the dataset $\mathcal{I}$, and the results are stored in two lists. The threshold algorithm [14] is employed to retrieve top-$k$ geo-images which have the highest similarity computing by Eq. 3 on these two lists. In the process of computing, if the similarity between the query $Q$ and a geo-image $I_i$ is larger than the similarity of the $k$-th geo-image $I_k$, then $I_i$ become the new $k$-th geo-image by replacing $I_k$.

For the visual representation of geo-image, we utilize the hand-crafted features, namely SIFT descriptor, and combining with BoVW model to encode the visual content, which is a conventional way used in many image search tasks [46], [75], [76]. Specifically, the visual features are extracted by SIFT technique and clustered by $k$-means method to generate visual dictionary. Each geo-image is represented by a visual word vector in which each element is the weight of the visual word measured by TF-IDF. The spatial index employed in RSVQ$_k$-R is R-Tree.

## IV. THE PROPOSED APPROACH

In this section, we propose an effective approach to overcome the challenge of RSVQ$_k$. Firstly, a novel hybrid index, named VR$^2$-Tree, is introduced in subsection IV-A, which can organize the geo-images efficiently in both aspects of geographical distribution and visual representation. In subsection IV-B we analyze the lower and upper bound of the search in theory. Then we develop a VR$^2$-Tree based algorithm to speed up the search markedly.

### A. HYBRID INDEX

#### 1) VR$^2$-TREE

**The Structure.** To efficiently organize the geo-images, we integrate the visual representation of geo-images and R-Tree to construct a novel hybrid index, named **V**isual **R**epresentation **R-Tree** (**VR$^2$-Tree**). As shown in Fig. 3, VR$^2$-Tree is a balanced tree built on a geo-image database $\mathcal{I}$. Each leaf node contains several tuples in the form of $\mathcal{T} = \langle I.\boldsymbol{\lambda}, I.\boldsymbol{v}, PTR(I)\rangle, I \in \mathcal{I}$. As defined in Section III, $I.\boldsymbol{\lambda} = (X, Y)$ is the geo-location descriptor and $I.\boldsymbol{v} = \left(v^{(1)}, v^{(2)}, \ldots, v^{(n)}\right)$ is the visual descriptor modeled by BoVW technique. $PTR(I)$ is the pointer of a geo-image $I$ in database. Each non-leaf node contains quadruples in the form of $\langle MBR, ANDOR, NUM, PTR(Child)\rangle$, where $MBR$ represents the minimum bounding rectangle of the child node, $ANDOR$ refers to two visual vectors, namely visual word weight AND vector (AND-vector for short) and visual word weight OR vector (OR-vector for short), which are generated by two novel operations for weighted visual word vector. The definitions of them is given thereinafter. $NUM$ is the total number of geo-images in the leaf nodes which belong to the subtree of this non-leaf node. $PTR(Child)$ is the pointer to the child node.

*Definition 4 (Weight AND):* Given two $\gamma$-dimensional visual word vectors $\boldsymbol{v}_1 = (v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(\gamma)})$ and $\boldsymbol{v}_2 = (v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(\gamma)})$, $W(\cdot)$ is the visual word weight function. The weight AND operation on $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, denoted as $\boldsymbol{v}_1 \bigotimes \boldsymbol{v}_2$, is to choose the minimum value of corresponding elements in $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, namely:

$$\begin{aligned}\boldsymbol{v}_1 \bigotimes \boldsymbol{v}_2 = (&Min(W(v_1^{(1)}), W(v_2^{(1)})),\\ &Min(W(v_1^{(2)}), W(v_2^{(2)})), \ldots,\\ &Min(W(v_1^{(\gamma)}), W(v_2^{(\gamma)}))) \end{aligned} \tag{10}$$

where $Min(\cdot, \cdot)$ is to return the minimum of the two inputs.

*Definition 5 (Weight OR):* Given two $\gamma$-dimensional visual word vectors $\boldsymbol{v}_1 = (v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(\gamma)})$ and $\boldsymbol{v}_2 = (v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(\gamma)})$, $W(\cdot)$ is the visual word weight function. The weight OR operation on $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, denoted as $\boldsymbol{v}_1 \bigoplus \boldsymbol{v}_2$, is to choose the maximum value of corresponding elements in $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, namely:

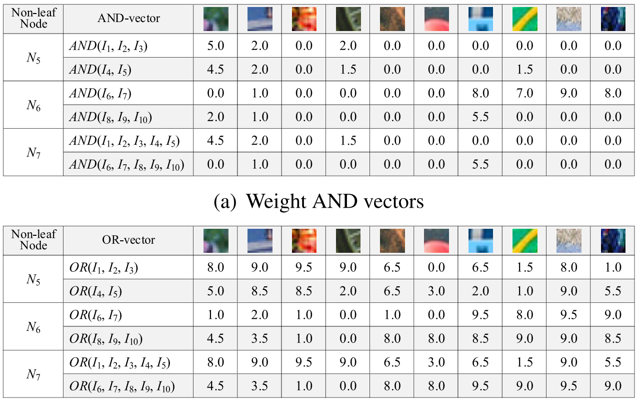
$$\begin{aligned}\boldsymbol{v}_1 \bigoplus \boldsymbol{v}_2 = (&Max(W(v_1^{(1)}), W(v_2^{(1)})),\\ &Max(W(v_1^{(2)}), W(v_2^{(2)})), \ldots,\\ &Max(W(v_1^{(\gamma)}), W(v_2^{(\gamma)}))) \end{aligned} \tag{11}$$

where $Max(\cdot, \cdot)$ is to return the maximum of the two inputs.

For a non-leaf node $N$ of a VR$^2$-Tree, it assumes that the geo-images contained in its subtree are $\{I_1, I_2, \ldots .I_m\}$, the visual word weight AND vector of a quadruple in $N$ is denoted as $AND(I_1, I_2, \ldots .I_m) = I_1.\boldsymbol{v} \bigotimes I_2.\boldsymbol{v} \ldots \bigotimes I_m.\boldsymbol{v}$. Similarly, the visual word weight OR vector is $OR(I_1, I_2, \ldots, I_m) = I_1.\boldsymbol{v} \bigoplus I_2.\boldsymbol{v} \ldots \bigoplus I_m.\boldsymbol{v}$.

According to Definition 4 and 5, we calculate the visual word weight AND vector and visual word weight OR vector of non-leaf nodes, i.e., $N_5, N_6, N_7$ in Example 2, as shown in Fig. 4. For example, $I_1, I_2, I_3$ are contained in the left subtree of $N_5$, and $I_4, I_5$ are contained in the right subtree. Thus, for non-leaf node $N_5$, the weight AND vectors of two quadruples are $AND(I_1, I_2, I_3)$ and $AND(I_4, I_5)$, respectively. Likewise, the weight OR vectors are $OR(I_1, I_2, I_3)$ and $OR(I_4, I_5)$.

**Visual Representation.** Instead of hand-crafted visual features, we propose to utilize deep CNN features to represent each geo-images since CNN features are powerful to represent semantic concept information. Specifically, AlexNet [39] is employed to extract the visual features from each geo-image in $\mathcal{I}$, i.e., $(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})_5 = ALEX(I_i), \forall I_i \in \mathcal{I}$, where $(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})_5$ is the output of 5-th convolutional layer. We sill use BoVW model to generate the visual word vector as the visual representation. Similar to the conventional manner, $k$-means technique is exploited to construct the CNN visual word dictionary containing $\gamma$ different words. Then each geo-image is encoded into the $\gamma$-dimensional visual word vector, i.e., $I_i.\boldsymbol{v} = BOVW((x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})_5)$ in which each word is weighted by TF-IDF method, namely $(W(v_i^{(1)}), W(v_i^{(2)}), \ldots, W(v_i^{(\gamma)})) = TF\text{-}IDF(I_i.\boldsymbol{v})$. In the following discussion, we denote the weighted visual words vector by $I_i.\bar{\boldsymbol{v}}$.



| Non-leaf Node | AND-vector | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $N_5$ | $AND(I_1, I_2, I_3)$ | 5.0 | 2.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | $AND(I_4, I_5)$ | 4.5 | 2.0 | 0.0 | 1.5 | 0.0 | 0.0 | 0.0 | 1.5 | 0.0 | 0.0 |
| $N_6$ | $AND(I_6, I_7)$ | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 7.0 | 9.0 | 8.0 |
| | $AND(I_8, I_9, I_{10})$ | 2.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.5 | 0.0 | 0.0 | 0.0 |
| $N_7$ | $AND(I_1, I_2, I_3, I_4, I_5)$ | 4.5 | 2.0 | 0.0 | 1.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | $AND(I_6, I_7, I_8, I_9, I_{10})$ | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.5 | 0.0 | 0.0 | 0.0 |

(a) Weight AND vectors

| Non-leaf Node | OR-vector | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $N_5$ | $OR(I_1, I_2, I_3)$ | 8.0 | 9.0 | 9.5 | 9.0 | 6.5 | 0.0 | 6.5 | 1.5 | 8.0 | 1.0 |
| | $OR(I_4, I_5)$ | 5.0 | 8.5 | 8.5 | 2.0 | 6.5 | 3.0 | 2.0 | 1.0 | 9.0 | 5.5 |
| $N_6$ | $OR(I_6, I_7)$ | 1.0 | 2.0 | 1.0 | 0.0 | 1.0 | 0.0 | 9.5 | 8.0 | 9.5 | 9.0 |
| | $OR(I_8, I_9, I_{10})$ | 4.5 | 3.5 | 1.0 | 0.0 | 8.0 | 8.0 | 8.5 | 9.0 | 9.0 | 8.5 |
| $N_7$ | $OR(I_1, I_2, I_3, I_4, I_5)$ | 8.0 | 9.0 | 9.5 | 9.0 | 6.5 | 3.0 | 6.5 | 1.5 | 9.0 | 5.5 |
| | $OR(I_6, I_7, I_8, I_9, I_{10})$ | 4.5 | 3.5 | 1.0 | 0.0 | 8.0 | 8.0 | 9.5 | 9.0 | 9.5 | 9.0 |

(b) Weight OR vectors

**FIGURE 4. The visual word weight AND and OR vectors of non-leaf nodes $N_5, N_6, N_7$ in Example 2.**

### 2) THE CONSTRUCTION ALGORITHM

Inspired by the R-Tree [52] insert operation, we develop a similar insertion algorithm based on the heuristics of minimizing the MBR to implement construction of the VR$^2$-Tree, as described in Algorithm 1 detailedly. What is slightly different from the above is that, instead of using the form of $(W(v_j^{(1)}), W(v_j^{(2)}), \ldots, W(v_j^{(\gamma)}))$, we propose to store the visual representation vector in a node $N$ in the new form of $(\langle \hbar_j^{(1)}, W(v_j^{(1)})\rangle, \langle \hbar_j^{(2)}, W(v_j^{(2)})\rangle, \ldots, \langle \hbar_j^{(\alpha)}, W(v_j^{(\alpha)})\rangle)$, where $\hbar_j^{(1)}$ is a code hashed from visual word $v_j^{(1)}$, $\alpha$ is the total number of visual words in $N$. To implement the hashing operation, we employ the technique proposed in [77], namely order preserving minimal perfect hashing.

Specifically, the procedure $OPMP\text{-}HASH(I.\bar{\boldsymbol{v}})$ in Line 5 is to generate the hash codes by order preserving minimal perfect hashing from the original visual words vector and produce the new representation vector, namely $\boldsymbol{V} = (\langle \hbar_j^{(1)}, W(v_j^{(1)})\rangle, \langle \hbar_j^{(2)}, W(v_j^{(2)})\rangle, \ldots, \langle \hbar_j^{(\alpha)}, W(v_j^{(\alpha)})\rangle)$. The procedure $ChooseLeaf(MBR)$ in Line 6 is invoked to choose the leaf node according to the $MBR$, which is similar to the implementation of R-Tree [52]. From Line 7 to 12, the procedures $N.Add(I.\bar{\boldsymbol{v}}, MBR)$, $N.SplitNode()$ and $M.AddNode(O, P)$ are similar to the processes of insertion

**Algorithm 1** **Insert**($I.\bar{\mathbf{v}}$, $MBR$)

```
1:  INPUT an original weighted visual representation vector $I.\bar{\mathbf{v}}$, a $MBR$.
2:  Initializing: A vector $\boldsymbol{V} \leftarrow null$;
3:  Initializing: A node $N \leftarrow null$;
4:  Initializing: A node $M \leftarrow null$;
5:  $\boldsymbol{V} \leftarrow OPMP\text{-}HASH(I.\bar{\mathbf{v}})$;
6:  $N \leftarrow ChooseLeaf(MBR)$;
7:  $N.Add(I.\bar{\mathbf{v}}, MBR)$;
8:  if $N$ needs to be split then
9:      $\{O, P\} \leftarrow N.SplitNode()$;
10:     if $N$ is the root node then
11:         $M.AddNode(O, P)$;
12:         $SetRoot(M)$;
13:     else
14:         $AdjustTree(N.Parent, O, P)$;
15:     end if
16: else if $N$ is not the root node then
17:     $AdjustTree(N.Parent, N, null)$;
18: end if
```

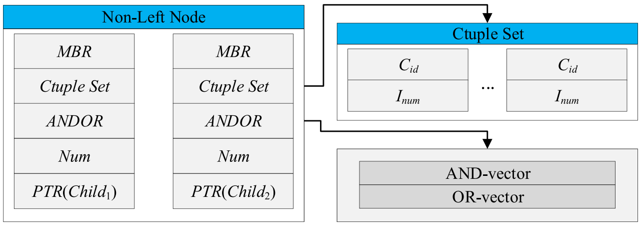


**FIGURE 5.** The non-leaf node structure of CVR$^2$-Tree.

in a R-Tree. Different from the algorithm *AdjustTree* in a R-Tree, the procedure *AdjustTree*$(\cdot)$ invoked in Line 14 and Line 17 is modified for the better compatibility with visual representations.

#### 3) THE EXTENSION OF VR$^2$-TREE

There is a limitation of the VR$^2$-Tree: although the VR$^2$-Tree can organize geo-images according to geographical proximity (by using MBR) as effectively as R-Tree, it ignores the visual similarity during the tree construction. In other words, it could well be that the visual similarity between the geo-images that close to each other in geographical is very small. This phenomenon is easy to find in real environment. For example, on a commercial street, the facilities usually fall into different categories, e.g. restaurant, clothing shop, cafe, cinema, etc. This leads to the low visual similarity between the geo-images collected in these different facilities.

To overcome this limitation, we propose to extend the VR$^2$-Tree by exploiting visual content clustering to modify the structure of the non-leaf node, and we call this extension as Clustering based **VR$^2$-Tree** (**CVR$^2$-Tree**). Specifically, before the construction of the tree, we use $k$-means method to partition the geo-image dataset $\mathcal{I}$ into $k$ clusters according to the visual similarity, i.e., $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\} = KMEANS(\mathcal{I})$. Different from the VR$^2$-Tree, the tuple $\mathcal{T}$ in non-leaf nodes of CVR$^2$-Tree, as shown in Fig. 5, contain a novel entry named *CEntry set* $\mathcal{S}_C = \{\mathcal{E}_C\}$. Each CEntry $\mathcal{E}_C$ corresponding to a cluster is in the following form: $\mathcal{E}_C : \langle C_{id}, I_{num}\rangle$, where $C_{id}$ is the id of the cluster, $I_{num}$ is the total number of geo-images belong to this cluster. For a non-leaf node, its CEntry set is the specific superposition of all the CEntry sets in its child nodes. To describe it clearly, we propose a novel operation, named CEntry set sum to define this calculation formally, as shown in the following.

*Definition 6 (CEntry Set Sum):* Given two CEntry set $\mathcal{S}_{C1}$ and $\mathcal{S}_{C2}$. The sum of these two CEntry sets, i.e., $\mathcal{S}_{C1} \biguplus \mathcal{S}_{C2}$ is defined as follows:

$$\mathcal{S}_{C1} \biguplus \mathcal{S}_{C2} = \mathcal{S}_{C1} \bigcup \mathcal{S}_{C2} \bigcup \mathcal{S}_{+} \setminus \mathcal{S}_{-}, \tag{12}$$

where,

$$\begin{aligned} \mathcal{S}_{+} = \{\mathcal{E}_C \mid &\forall \mathcal{T}_{Ci} \in \mathcal{S}_{C1}, \forall \mathcal{T}_{Cj} \in \mathcal{S}_{C2}, \\ &\text{if } \mathcal{E}_{Ci}.C_{id} = \mathcal{E}_{Cj}.C_{id}, \text{then} \\ &\mathcal{E}_C.I_{num} = \mathcal{E}_{Ci}.I_{num} + \mathcal{E}_{Cj}.I_{num}\}, \end{aligned} \tag{13}$$

and,

$$\begin{aligned} \mathcal{S}_{-} = \{\mathcal{E}_{Ci}, \mathcal{E}_{Cj} \mid \forall \mathcal{E}_{Ci} \in \mathcal{S}_{C1}, \quad &\forall \mathcal{E}_{Cj} \in \mathcal{S}_{C2}, \\ &\mathcal{E}_{Ci}.C_{id} = \mathcal{E}_{Cj}.C_{id}\}, \end{aligned} \tag{14}$$

and the operator $\bigcup$ is the set union operator, $\setminus$ is the set minus operator.

Therefore, for a non-leaf node $N$, its CEntry set $N.\mathcal{S}_C$ is the sum of all the CEntry sets in its child nodes, i.e., $N.\mathcal{S}_C = \biguplus_{i=1}^{L} ChildNode(N)_i.\mathcal{S}_C$, where $ChildNode(N)_i$ represents the $i$-th child node of $N$, $L$ is the total number of children. For example, consider all the geo-images $\{I_1, I_2, \ldots, I_{10}\}$ in Example 2, according to visual similarity we cluster them into 4 clusters: $\mathcal{C}_1 = \{I_1, I_2, I_5\}, \mathcal{C}_2 = \{I_3, I_4\}, \mathcal{C}_3 = \{I_6, I_7, I_8, I_9\}$ and $\mathcal{C}_4 = \{I_{10}\}$. Thus, for the non-leaf node $N_5$, $N_5.\mathcal{S}_{C1} = \{\langle C_1, 2\rangle, \langle C_2, 1\rangle\}$ and $N_5.\mathcal{S}_{C2} = \{\langle C_1, 1\rangle, \langle C_2, 1\rangle\}$; for $N_6$, $N_6.\mathcal{S}_{C1} = \{\langle C_3, 2\rangle\}$ and $N_6.\mathcal{S}_{C2} = \{\langle C_3, 2\rangle, \langle C_4, 1\rangle\}$; and for $N_7$, $N_7.\mathcal{S}_{C1} = N_5.\mathcal{S}_{C1} \biguplus N_5.\mathcal{S}_{C2} = \{\langle C_1, 3\rangle, \langle C_2, 2\rangle\}$ and $N_7.\mathcal{S}_{C2} = N_6.\mathcal{S}_{C1} \biguplus N_6.\mathcal{S}_{C2} = \{\langle C_3, 4\rangle, \langle C_4, 1\rangle\}$.

Like the AND-vector and OR-vector in the node of VR$^2$-Tree, we can calculate the CAND-vector and COR-vector for each cluster. Specifically, the CAND-vector contains the minimal weights of each visual words included in the cluster, and the COR-vector contains the maximum weights of each visual words. For the four clusters $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ mentioned-above, the CAND-vectors and COR-vectors of them are shown in Fig. 6.

### B. RSVQ$_K$ ALGORITHM

Based on the CVR$^2$-Tree, we carefully design a novel algorithm to solve the RSVQ$_k$ problem efficiently. Before introduce the detail of this algorithm in Section IV-B.3, we discuss how to compute the lower bound and upper bound of similarity IV-B.1.

| Cluster | AND-vector | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | $AND(I_1, I_2, I_5)$ | 5.0 | 2.0 | 7.0 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| $C_2$ | $AND(I_3, I_4)$ | 4.5 | 2.0 | 0.0 | 1.5 | 0.0 | 0.0 | 2.0 | 0.0 | 8.0 | 1.0 |
| $C_3$ | $AND(I_6, I_7, I_8, I_9)$ | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 7.0 | 8.0 | 8.0 |
| $C_4$ | $AND(I_{10})$ | 4.5 | 3.5 | 0.0 | 0.0 | 8.0 | 8.0 | 5.5 | 0.0 | 0.0 | 0.0 |

(a) Weight AND vectors

| Cluster | AND-vector | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | $AND(I_1, I_2, I_5)$ | 7.0 | 9.0 | 9.5 | 9.0 | 6.5 | 0.0 | 1.0 | 1.5 | 0.0 | 1.5 |
| $C_2$ | $AND(I_3, I_4)$ | 8.0 | 3.5 | 0.0 | 9.0 | 0.0 | 3.0 | 6.5 | 1.0 | 9.0 | 5.5 |
| $C_3$ | $AND(I_6, I_7, I_8, I_9)$ | 3.0 | 2.0 | 1.0 | 0.0 | 1.0 | 1.0 | 9.5 | 9.0 | 9.5 | 9.0 |
| $C_4$ | $AND(I_{10})$ | 4.5 | 3.5 | 0.0 | 0.0 | 8.0 | 8.0 | 5.5 | 0.0 | 0.0 | 0.0 |

(b) Weight OR vectors

**FIGURE 6. The visual word weight CAND and COR vectors of clusters $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ in Example 2. Similar to the AND and OR operations in VR$^2$-Tree, the CAND-vector contains the minimal weights of each visual words included in the cluster, and the COR-vector contains the maximum weights of each visual words.**

### 1) LOWER BOUND AND UPPER BOUND

To explain the computation of lower bound and upper bound, firstly, we present the notions of minimal similarity and maximal similarity between two tuples in a CVR$^2$-Tree, and then introduce the lower bound and upper bound contribution list.

Given a CVR$^2$-Tree $\mathfrak{T}$, $\forall \mathcal{T} \in \mathfrak{T}$, the lower bound and upper bound of similarity between the tuple $\mathcal{T}$ and its $k$-th most similar geo-image are denoted as $\lfloor \mathcal{T} \rfloor$ and $\lceil \mathcal{T} \rceil$ respectively. The $\gamma$-dimensional visual word weight AND vector and OR vector of $\mathcal{T}$ are denoted as $\mathcal{T}.\boldsymbol{A} = (a^{(1)}, a^{(2)}, \ldots, a^{(\gamma)})$ and $\mathcal{T}.\boldsymbol{O} = (o^{(1)}, o^{(2)}, \ldots, o^{(\gamma)})$ respectively. we define the minimal similarity between two tuples in CVR$^2$-Tree as follows.

*Definition 7 (Minimal Similarity (MinSim)):* Let $\mathcal{T}_1$ and $\mathcal{T}_2 \in \mathfrak{T}$ be two tuples, the minimal similarity between $\mathcal{T}_1$ and $\mathcal{T}_{\in}$ is denoted as $MinSim(\mathcal{T}_1, \mathcal{T}_2)$, which is computed by the following equation:

$$\begin{aligned} &MinSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = Max(\mu \times tMaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\qquad +(1-\mu) \times MinVisSim(\mathcal{T}_1, \mathcal{T}_2),\\ &\qquad \mu \times MaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\qquad +(1-\mu) \times tMinVisSim(\mathcal{T}_1, \mathcal{T}_2)), \end{aligned} \tag{15}$$

$$\begin{aligned} &tMaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = 1 - \frac{MinMaxEucliDst(\mathcal{T}_1, \mathcal{T}_2)}{MaxDst(\mathcal{I})}, \end{aligned} \tag{16}$$

$$\begin{aligned} &MinVisSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = \frac{MinExJacc(\mathcal{T}_1, \mathcal{T}_2)}{MaxVisSim(\mathcal{I})}, \end{aligned} \tag{17}$$

$$\begin{aligned} &MaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = 1 - \frac{MaxEucliDst(\mathcal{T}_1, \mathcal{T}_2)}{MaxDst(\mathcal{I})}, \end{aligned} \tag{18}$$

$$\begin{aligned} &tMinVisSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = \frac{tMinExJacc(\mathcal{T}_1, \mathcal{T}_2)}{MaxVisSim(\mathcal{I})}, \end{aligned} \tag{19}$$

where $tMaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)$ proposed in [78] is a tighter Euclidean distance measurement than $MaxGeoSim(\mathcal{T}_1, \mathcal{T}_2)$ that is the maximal Euclidean distance between $\mathcal{T}_1.MBR$ and $\mathcal{T}_2.MBR$, and

$$\begin{aligned} &MinExJacc(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = \frac{\sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)}}{\sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)2} + \sum_{i=1}^{\gamma} \mathcal{T}_2.W^{(i)2} - \sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)}}, \end{aligned} \tag{20}$$

where, $\mathcal{T}_1.W^{(i)}$ denotes the weight of $i$-th visual word,

$$\begin{cases} \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.o^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.a^{(i)}, \text{ if} \\ \quad \mathcal{T}_1.a^{(i)} \times \mathcal{T}_1.o^{(i)} \geq \mathcal{T}_2.a^{(i)} \times \mathcal{T}_2.o^{(i)} \\ \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.a^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.o^{(i)}, \text{ otherwise} \end{cases} \tag{21}$$

and,

$$\begin{aligned} &tMinExJacc(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad = \max_{1 \leq \iota \leq \gamma} \left( \frac{\mathcal{T}_1.W^{(\iota)} \times \mathcal{T}_2.W^{(\iota)} + \Sigma'}{\mathcal{T}_1.W^{(\iota)2} + \mathcal{T}_2.W^{(\iota)2} - \mathcal{T}_1.W^{(\iota)} \times \mathcal{T}_2.W^{(\iota)} + \Sigma} \right),\\ &\Sigma = \sum_{i=1, i \neq \iota}^{\gamma} \mathcal{T}_1.W^{(i)2} + \mathcal{T}_2.W^{(i)2} - \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)},\\ &\Sigma' = \sum_{i=1, i \neq \iota}^{\gamma} \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)} \end{aligned} \tag{22}$$

where,

$$\begin{cases} \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.o^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.a^{(i)}, \text{ if} \\ \quad \mathcal{T}_1.a^{(i)} \times \mathcal{T}_1.o^{(i)} \geq \mathcal{T}_2.a^{(i)} \times \mathcal{T}_2.o^{(i)} \\ \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.a^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.o^{(i)}, \text{ otherwise} \end{cases} \tag{23}$$

$$\begin{aligned} &\mathcal{T}_2.W^{(\iota)}\\ &\quad = \begin{cases} \mathcal{T}_2.o^{(\iota)}, \text{ if } \mathcal{T}_1.a^{(\iota)} \times \mathcal{T}_1.o^{(\iota)} > \mathcal{T}_2.a^{(\iota)} \times \mathcal{T}_2.o^{(\iota)} \\ \mathcal{T}_2.a^{(\iota)}, \text{ otherwise} \end{cases} \end{aligned} \tag{24}$$

$$\begin{aligned} &\mathcal{T}_1.W^{(\iota)}\\ &\quad = \begin{cases} \mathcal{T}_1.a^{(\iota)}, \text{ if } \sqrt{\mathcal{T}_1.a^{(\iota)} \times \mathcal{T}_1.o^{(\iota)}} < \mathcal{T}_2.W^{(\iota)} \\ \mathcal{T}_1.o^{(\iota)}, \text{ otherwise} \end{cases} \end{aligned} \tag{25}$$

*Property 1:* Given a CVR$^2$-Tree $\mathfrak{T}$, $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}$. $\exists I_2 \in \mathcal{T}_2$ $s.t.$ $\forall I \in \mathcal{T}_1$, $Sim(I_1, I_2) \geq MinSim(\mathcal{T}_1, \mathcal{T}_2)$.

*Definition 8 (Maximal Similarity (MaxSim)):* Let $\mathcal{T}_1$ and $\mathcal{T}_2 \in \mathfrak{T}$ be two tuples, the maximal similarity between $\mathcal{T}_1$ and $\mathcal{T}_{\in}$ is denoted as $MaxSim(\mathcal{T}_1, \mathcal{T}_2)$, which is computed by the following equation:

$$\begin{aligned} MaxSim(\mathcal{T}_1, \mathcal{T}_2) &= \mu \times MinGeoSim(\mathcal{T}_1, \mathcal{T}_2)\\ &\quad +(1-\mu) \times MaxVisSim(\mathcal{T}_1, \mathcal{T}_2), \end{aligned} \tag{26}$$

$$MinGeoSim(\mathcal{T}_1, \mathcal{T}_2) = 1 - \frac{MinEucli(\mathcal{T}_1, \mathcal{T}_2)}{MaxDst(\mathcal{I})}, \tag{27}$$

$$MaxVisSim(\mathcal{T}_1, \mathcal{T}_2) = \frac{MaxExJacc(\mathcal{T}_1, \mathcal{T}_2)}{MaxVisSim(\mathcal{I})} \tag{28}$$

where $MinGeoSim(\mathcal{T}_1, \mathcal{T}_2)$ is the minimal Euclidean distance measurement between two MBRs of $\mathcal{T}_1$ and $\mathcal{T}_2$,

$MaxExJacc(\mathcal{T}_1, \mathcal{T}_2)$ is the maximal visual similarity between $\mathcal{T}_1$ and $\mathcal{T}_2$, which is computed by the following equation:

$$MaxExJacc(\mathcal{T}_1, \mathcal{T}_2) = \frac{\sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)}}{\sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)2} + \sum_{i=1}^{\gamma} \mathcal{T}_2.W^{(i)2} - \sum_{i=1}^{\gamma} \mathcal{T}_1.W^{(i)} \times \mathcal{T}_2.W^{(i)}}, \tag{29}$$

$$\begin{cases} \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.a^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.o^{(i)}, \text{ if} \\ \mathcal{T}_1.a^{(i)} > \mathcal{T}_2.o^{(i)} \\ \mathcal{T}_1.W^{(i)} = \mathcal{T}_1.o^{(i)}, \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.a^{(i)}, \text{ if} \\ \mathcal{T}_1.o^{(i)} < \mathcal{T}_2.a^{(i)} \\ \mathcal{T}_1.W^{(i)} = \mathcal{T}_2.W^{(i)} = \mathcal{T}_1.o^{i}, \text{ if} \\ \mathcal{T}_2.a^{(i)} \leq \mathcal{T}_1.o^{(i)} \leq \mathcal{T}_2.o^{(i)} \\ \mathcal{T}_1.W^{(i)} = \mathcal{T}_2.W^{(i)} = \mathcal{T}_2.O^{(i)}, \text{ otherwise} \end{cases} \tag{30}$$

*Property 2:* Given a CVR$^2$-Tree $\mathfrak{T}$, $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}$. $\forall I_2 \in \mathcal{T}_2, \forall I \in \mathcal{T}_1, Sim(I_1, I_2) \leq MaxSim(\mathcal{T}_1, \mathcal{T}_2)$.

According to the definition of minimal and maximal similarity between two tuples in VR$^2$-Tree or CVR$^2$-Tree, we propose other two notions, namely Lower Bound Determinant Queue and Upper Bound Determinant Queue, which are used to reduce the candidate set effectively.

*Definition 9 (Lower Bound Determinant Queue ($\Psi_L$)):* Given a CVR$^2$-Tree $\mathfrak{T}$, $\mathcal{S}_{\mathcal{T}} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$ is a tuple set in which each tuple is in $\mathfrak{T}$. For a tuple $\mathcal{T} \in \mathcal{S}_{\mathcal{T}}$, a lower bound determinant queue of $\mathcal{T}$, denoted as $\Psi_L(\mathcal{T}) = (\psi_L^{(1)}, \psi_L^{(2)}, \ldots, \psi_L^{(\alpha)})$, is a queue containing $\alpha$ items in the form of $\psi_L^{(i)} = \langle \xi_i, \hat{\mathcal{T}}_i, \vartheta_i \rangle$ that are sorted in descending order of $\xi_i$, wherein $\alpha \in N^+, \alpha \in [1, k]$, $i \in [1, \alpha]$, $\hat{\mathcal{T}}_i$ is another tuple in $\mathcal{S}_{\mathcal{T}}$, namely $\hat{\mathcal{T}}_i \neq \mathcal{T}$, $\xi_i$ is the value of similarity between $\mathcal{T}$ and $\hat{\mathcal{T}}_i$, i.e., $\xi_i = MinSim(\mathcal{T}, \hat{\mathcal{T}}_i)$, $\vartheta_i$ is an integer that is assigned by the following condition:

$$\vartheta_i = \begin{cases} |\hat{\mathcal{T}}_i| - 1, & \text{if } \xi_i = MinSim(\mathcal{T}, \hat{\mathcal{T}}_i) \\ 1, & \textit{otherwise} \end{cases} \tag{31}$$

that minimizes $\alpha$ $s.t.$ $\sum_{i=1}^{\alpha} \vartheta_i \geq k$.

*Property 3:* Given a lower bound determinant queue $\Psi_L(\mathcal{T})$, $\psi_L^{(\alpha)} = \langle \xi_\alpha, \hat{\mathcal{T}}_\alpha, \vartheta_\alpha \rangle$ is the $\alpha$-th item of the queue. If $\psi_L^{(\alpha)}.\xi_\alpha \geq MaxSim(\mathcal{T}, Q)$, the subtree of $\mathcal{T}$ can be pruned safely.

According to the Definition 9 and Property 3, the candidate set can be reduced by pruning the tuples that are not similar enough to the query. Therefore, the lower bound $\lfloor \mathcal{T} \rfloor$ can be assigned by $\psi^{(\alpha)}.\xi_\alpha$.

*Definition 10 (Upper Bound Determinant Queue ($\Psi_U$)):* Given a CVR$^2$-Tree $\mathfrak{T}$, $\mathcal{S}_{\mathcal{T}} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$ is a tuple set in which each tuple is in $\mathfrak{T}$. For a tuple $\mathcal{T} \in \mathcal{S}_{\mathcal{T}}$, a upper bound determinant queue of $\mathcal{T}$, denoted as $\Psi_U(\mathcal{T}) = (\psi_U^{(1)}, \psi_U^{(2)}, \ldots, \psi_U^{(\beta)})$, is a queue containing $\beta$ items in the form of $\psi_U^{(i)} = \langle \xi_i, \hat{\mathcal{T}}_i \rangle$ that are sorted in descending order of $\xi_i$, wherein $\beta \in N^+, \beta \in [1, k]$, $i \in [1, \beta]$, $\hat{\mathcal{T}}_i$ is another tuple in $\mathcal{S}_{\mathcal{T}}$, namely $\hat{\mathcal{T}}_i \neq \mathcal{T}$, $\xi_i = MaxSim(\mathcal{T}, \hat{\mathcal{T}}_i)$, $\beta$ is maximized to satisfy the condition $1 + \sum_{i=1}^{\beta-1} |\hat{\mathcal{T}}_i| \leq k$.

Similar to lower bound determinant queue, upper bound determinant queue has an important property that is formulated as follows.

*Property 4:* Given a upper bound determinant queue $\Psi_U(\mathcal{T})$, $\psi_U^{(\beta)} = \langle \xi_\beta, \hat{\mathcal{T}}_\beta \rangle$ is the $\beta$-th item. If $\xi_\beta < MinSim(\mathcal{T}, Q)$, then $Q$ is one of the $k$ most similar geo-images for all geo-images in $\mathcal{T}$.

It is easy to understand from the Property 4 that the number of geo-images that are similar to any geo-image in the tuple $\mathcal{T}$ (i.e., similarities of them are larger or equal to $MinSim(\mathcal{T}, Q)$) is at most $k - 1$. Therefore, the upper bound $\lceil \mathcal{T} \rceil$ can be assigned by $\psi^{(\beta)}.\xi_\beta$.

### 2) OPTIMIZATION: TIGHTER BOUND VIA CVR$^2$-TREE

To improve the performance of search, we propose a optimization method via CVR$^2$-Tree to obtain a tighter bound. According to cluster id, this method aims to identify the outliers from the tuples in CVR$^2$-Tree, which are picked out from the normal geo-images and severally calculate their bounds. Thus, the bounds of the normal tuples can be tighter.

The outlies can be identified according to the following two situations:

*Situation-1:* For a tuple $\mathcal{T}$, most geo-images in the subtree of $\mathcal{T}$ can be pruned, but there exist a few of geo-images that cannot be pruned, and we treat them as outliers. Obviously, these outliers make the tuple $\mathcal{T}$ and its subtree cannot be pruned. Formally, for a query $Q$ and a tuple $\mathcal{T}$, if $MinSim(\mathcal{T}, Q) < \lfloor \mathcal{T} \rfloor < MaxSim(\mathcal{T}, Q)$, and there exist a subset $Sub_1(\{\mathcal{C}\})$ of $\{\mathcal{C}\}$ of $\mathcal{T}$ $s.t.$ $\sum_{\mathcal{C}_i \in Sub_1(\{\mathcal{C}\})} \mathcal{C}_i.N \geq \epsilon|\mathcal{T}|$, and $\forall \mathcal{C}_i \in Sub_1(\{\mathcal{C}\}) s.t.$

$$\mu\Big(1 - \frac{MinEucliDst(\mathcal{T}_1, \mathcal{T}_2)}{MaxDst(\mathcal{I})}\Big) + (1-\mu)MaxVisSim(\mathcal{C}_i, Q) < \lfloor \mathcal{T} \rfloor$$

where $\epsilon$ is a parameter. The geo-images that are in $\mathcal{T}$ but not in $Sub_1(\{\mathcal{C}\})$ are treated as outliers.

*Situation-2:* For a tuple $\mathcal{T}$, most geo-images in the subtree of $\mathcal{T}$ can be treated as results, but there exist a few of geo-images that cannot be treated as results. Therefore, the tuple $\mathcal{T}$ cannot be treated as a result tuple. Formally, for a query $Q$ and a tuple $\mathcal{T}$, if $MinSim(\mathcal{T}, Q) < \lceil \mathcal{T} \rceil < MaxSim(\mathcal{T}, Q)$, and there exist a subset $Sub_2(\{\mathcal{C}\})$ of $\{\mathcal{C}\}$ of $\mathcal{T}$ $s.t.$ $\sum_{\mathcal{C}_i \in Sub_2(\{\mathcal{C}\})} \mathcal{C}_i.N \geq \epsilon|\mathcal{T}|$, and $\forall \mathcal{C}_i \in Sub_2(\{\mathcal{C}\}) s.t.$

$$\mu\Big(1 - \frac{MaxEucliDst(\mathcal{T}_1, \mathcal{T}_2)}{MaxDst(\mathcal{I})}\Big) + (1-\mu)MinVisSim(\mathcal{C}_i, Q) < \lceil \mathcal{T} \rceil$$

where $\epsilon$ is a parameter. The geo-images that are in $\mathcal{T}$ but not in $Sub_2(\{\mathcal{C}\})$ are treated as outliers.

According to the above two situations, we can identify the tuples whether their subtree can be pruned or treated as results. The implementation of this optimization method is shown in the next part.

**Algorithm 2** RSVQ*k* Algorithm

```
1:  INPUT: the tree root of a CVR$^2$-Tree $\mathfrak{T}.Root$, a reverse spatial visual top-$k$ query $Q$.
2:  OUTPUT: All the geo-images $I$, s.t., $I \in RSVQk(Q, k, \mathcal{I})$.
3:  Initializing: A max-priority queue $\mathcal{P} \leftarrow null$;
4:  Initializing: A candidate geo-image list $\mathcal{L}_C \leftarrow null$;
5:  Initializing: A pruned tuples list $\mathcal{L}_P \leftarrow null$;
6:  Initializing: A results list $\mathcal{L}_R \leftarrow null$;
7:  $EnQueue(\mathcal{P}, \mathfrak{T}.Root)$;
8:  while $IsNotEmpty(\mathcal{P})$ do
9:    $\mathcal{T}_P \leftarrow DeQueue(\mathcal{P})$;
10:   for each child tuple $\mathcal{T}$ of $\mathcal{T}_P$ do
11:     $\Psi_L(\mathcal{T}) \leftarrow \Psi_L(\mathcal{T}_P)$;
12:     $\Psi_U(\mathcal{T}) \leftarrow \Psi_U(\mathcal{T}_P)$;
13:     if $\neg IsResultOrPruned(\mathcal{T}, Q, \mathcal{L}_R)$ then
14:       for each tuple $\hat{\mathcal{T}} \in \mathcal{L}_C \cup \mathcal{L}_R \cup \mathcal{P}$ do
15:         $Update\Psi(\mathcal{T}, \hat{\mathcal{T}})$;
16:         if $IsResultOrPruned(\hat{\mathcal{T}}, Q, \mathcal{L}_R)$ then
17:           $Remove(\hat{\mathcal{T}}, \mathcal{L}_C \cup \mathcal{L}_R \cup \mathcal{P})$;
18:         end if
19:       end for
20:       if $\neg IsResultOrPruned(\mathcal{T}, Q, \mathcal{L}_R)$ then
21:         if $IsIndexNode(\mathcal{T})$ then
22:           if $\mathcal{T}$ is Situation-1 or Situation-2 then
23:             for each $\mathcal{T}' \in Subtree(\mathcal{T})$ do
24:               if $\mathcal{C}_{\mathcal{T}'} \subset Sub_1(\{\mathcal{C}\})$ then
25:                 $Prune(\mathcal{T}')$;
26:               else if $\mathcal{C}_{\mathcal{T}'} \subset Sub_2(\{\mathcal{C}\})$ then
27:                 $\mathcal{L}_R.Add(\mathcal{T}')$;
28:               else if $IsIndexNode(\mathcal{T}')$ then
29:                 $EnQueue(\mathcal{P}, \mathcal{T}')$;
30:               else
31:                 $\mathcal{L}_C.Add(\mathcal{T}')$;
32:               end if
33:             end for
34:           end if
35:           $EnQueue(\mathcal{P}, \mathcal{T})$;
36:         else
37:           $\mathcal{L}_C.Add(\mathcal{T})$;
38:         end if
39:       end if
40:     end if
41:   end for
42: end while
43: $Verify(\mathcal{L}_C, \mathcal{L}_P, \mathcal{L}_R, Q)$;
```

### 3) TOP-*k* SEARCH ALGORITHM

Based on the CVR$^2$-Tree and the notion of the lower bound and upper bound, we carefully develop an efficient search algorithm for the task of RSVQ*k*, which is shown in Algorithm 2.

Specifically, the inputs of RSVQ*k* algorithm are a tree root of a CVR$^2$-Tree and a query $Q$. This algorithm accesses the CVR$^2$-Tree $\mathfrak{T}$ from top to bottom and computes the lower bound $\lfloor\mathcal{T}\rfloor$ and $\lceil\mathcal{T}\rceil$ step-by-step for each $\mathcal{T} \in \mathfrak{T}$. Then, according to $\lfloor\mathcal{T}\rfloor$ and $\lceil\mathcal{T}\rceil$, the algorithm to determine a tuple $\mathcal{T}$ should be pruned or the geo-images in it are the results. At the beginning of it, a max-priority queue $\mathcal{P}$ and three lists are initialized, i.e, a candidate geo-image list $\mathcal{L}_C$ in which the geo-image need to be checked, a pruned tuples list $\mathcal{L}_P$ in which the tuples will not be results and a results list $\mathcal{L}_R$. The first step is to put the tree root into the queue $\mathcal{P}$ by invoking the procedure $EnQueue(\mathcal{P}, \mathfrak{T}.Root)$ (in Line 7). Then If the queue $\mathcal{P}$ is not empty, the tuple with the highest priority, denoted by $\mathcal{T}_P$ is dequeued from $\mathcal{P}$ (Lines 8-9). After that, for each child $\mathcal{T}$ of $\mathcal{T}_P$, it inherits the lower bound determinant list and upper bound determinant list from $\mathcal{T}_P$ (Lines 11-12). Based on $\Psi_L(\mathcal{T})$ and $\Psi_U(\mathcal{T})$, the procedure $IsResultOrPruned(\mathcal{T}, Q, \mathcal{L}_R)$ (Algorithm 3) is invoked to determine whether $\mathcal{T}$ is a result or need to be pruned (Line 13). As shown in Algorithm 3, if $\lfloor\mathcal{T}\rfloor \geq MaxSim(\mathcal{T}, Q)$, that means $\mathcal{T}$ can be pruned, we put $\mathcal{T}$ into list $\mathcal{L}_P$; if $\lceil\mathcal{T}\rceil < MinSim(\mathcal{T}, Q)$ and $\mathcal{T}$ is the rightest child, that means $\mathcal{T}$ can be treated as a result, we put it into results list $\mathcal{L}_R$; if $\mathcal{T}$ does not belongs to above situations, we tighten the lower bound and upper bound by

**Algorithm 3** IsResultOrPruned($\mathcal{T}$, $Q$, $\mathcal{L}_R$)

```
if $\lfloor\mathcal{T}\rfloor \geq MaxSim(\mathcal{T}, Q)$ then
  $\mathcal{L}_P.Add(\mathcal{T})$;
  return true;
else if $\lceil\mathcal{T}\rceil < MinSim(\mathcal{T}, Q)$ and $IsRightest(\mathcal{T})$ then
  $\mathcal{L}_R.Add(\mathcal{T}.Subtree)$
  return true;
else
  return false;
end if
```

**Algorithm 4** Update$\Psi(\mathcal{T}, \hat{\mathcal{T}})$

```
for each item $\psi_L^{(i)} \in \Psi_L(\mathcal{T})$ do
  if $\psi_L^{(i)}.\hat{\mathcal{T}}_i = \mathcal{T} \,||\, \psi_L^{(i)}.\hat{\mathcal{T}}_i = Parent(\mathcal{T})$ then
    $Remove(\psi_L^{(i)}, \Psi_L(\mathcal{T}))$;
  end if
end for
if $\lceil\mathcal{T}\rceil < MaxSim(\mathcal{T}, \hat{\mathcal{T}})$ then
  $\Psi_U(\mathcal{T}) \leftarrow \{\psi_U\}_t \subset \Psi_U(\mathcal{T})$ by $MaxSim(\mathcal{T}, \hat{\mathcal{T}})$,
  s.t. $\sum_{i=1}^{t} \Psi_U(\mathcal{T}).\vartheta_i \geq k$;
end if
if $\lfloor\mathcal{T}\rfloor < tMiNSim(\mathcal{T}, \hat{\mathcal{T}})$ then
  $\Psi_L(\mathcal{T}) \leftarrow \{\psi_L\}_t \subset \Psi_L(\mathcal{T})$ by $tMiNSim(\mathcal{T}, \hat{\mathcal{T}})$,
  s.t. $\sum_{i=1}^{t} \Psi_L(\mathcal{T}).\vartheta_i \geq k$;
end if
if $\lfloor\mathcal{T}\rfloor < MinSim(\mathcal{T}, \hat{\mathcal{T}})$ then
  $\Psi_L(\mathcal{T}) \leftarrow \{\psi_L\}_t \subset \Psi_L(\mathcal{T})$ by $MinSim(\mathcal{T}, \hat{\mathcal{T}})$
  s.t. $\sum_{i=1}^{t} \Psi_L(\mathcal{T}).\vartheta_i \geq k$;
end if
```

**Algorithm 5** Verify($\mathcal{L}_C, \mathcal{L}_P, \mathcal{L}_R, Q$)

```
1:  while IsNotEmpty($\mathcal{L}_C$) do
2:      Initialize $\mathcal{T} \in \mathcal{L}_P$ with the lowest level;
3:      $\mathcal{L}_P = \mathcal{L}_P - \{\mathcal{T}\}$;
4:      for each geo-image $I \in \mathcal{L}_C$ do
5:          $Update\Psi(I, \mathcal{T})$;
6:          if $IsResultOrPruned(I, Q)$ then
7:              $\mathcal{L}_C = \mathcal{L}_C - \{I\}$;
8:          end if
9:      end for
10:     for each child tuple $\hat{\mathcal{T}}$ of $\mathcal{T}$ do
11:         $\mathcal{L}_P = \mathcal{L}_P \cup \{\hat{\mathcal{T}}\}$;
12:     end for
13: end while
```

invoking procedure $Update\Psi(\mathcal{T}, \hat{\mathcal{T}})$ using $\hat{\mathcal{T}} \in \mathcal{L}_C \cup \mathcal{L}_R \cup \mathcal{P}$ (Lines 14-15). In Line 16 and 17, the algorithm invokes procedure *IsResultOrPruned* again to determine whether $\hat{\mathcal{T}}$ is pruned or treated as a result. If yes, then the algorithm removes $\hat{\mathcal{T}}$ from $\mathcal{P}$ or $\mathcal{L}_C$. In Lines 20-35, if $\mathcal{T}$ is not a result or pruned, and meanwhile it is an index node (Lines 20-21), then we identify whether the tuple $\mathcal{T}$ belongs to situation-1 or situation-2. If yes, the algorithm checks whether the tuples in subtree of $\mathcal{T}$ are results or not based on the relation between $\mathcal{C}_{\mathcal{T}'}$ and the cluster set $Sub_1(\{\mathcal{C}\})$ and $Sub_2(\{\mathcal{C}\})$. If not, the algorithm puts $\mathcal{T}$ into queue $\mathcal{P}$. Finally, in Line 43, the procedure *Verify* is invoked to decide whether the geo-images in list $\mathcal{L}_C$ are results.

The pseudo-code of procedure *Verify* is shown in Algorithm 5, which aims to check the effect of the tuples in $\mathcal{T}_{\mathcal{P}}$ on each tuples in $\mathcal{L}_{\mathcal{C}}$. First, in Lines 1-2, this procedure chooses a tuple from the list $\mathcal{L}_{\mathcal{P}}$ with the lowest level in the CVR$^2$-Tree. The reason of this process is that the tuples in the lower level generally have tighter bounds. That means they are more likely to identify the tuples that are results or not. In Lines 4-7, the tuple $\mathcal{T}$ is used to update the determinant queue of each geo-image that is contained in $\mathcal{L}_C$, then the geo-images are checked whether they can be dropped from the $\mathcal{L}_C$. In Line 10-11, this algorithm adds child tuple of $\mathcal{T}$ into list $\mathcal{L}_{\mathcal{P}}$, due to the effect on the candidates in $\mathcal{L}_C$.

## V. EXPERIMENTS

In this section, the comprehensive experiments on four datasets are presented, which evaluate the performance of the proposed approach. Firstly, the datasets and workload of the experiments are introduced in section V-A, then discuss the evaluations in section V-B.

### A. DATASETS AND WORKLOAD

#### 1) DATASETS

In our experiments, four synthetic geo-image datasets are used to evaluate the performance of various approaches. Two common used image datasets, i.e., Flickr and ImageNet, are used as the source of the synthetic geo-image datasets. The following four datasets are deployed in the experiments:

- **Flickr-RP**. The synthetic dataset Flickr-RP is produced by obtaining geographical locations from corresponding spatial datasets from Rtree-Portal[6] and randomly geo-tagging the images in Flickr,[7] the most popular photo-sharing platform. That means we do not use the original geo-tags of these images. To evaluate the scalability of the proposed approach, The dataset size varies from 200K to 1000K.
- **Flickr-US**. The synthetic dataset Flickr-US is produced by obtaining geographical locations from the US Board on Geographic Names.[8] Like the dataset Flickr-RP, we use these geographical location information to generate new geo-tags for the images in Flickr.
- **ImageNet-RP**. The synthetic dataset ImageNet-RP is generated by obtaining geographical locations from the US Board on Geographic Names[9] and randomly geo-tagging the images obtaining from the largest image dataset ImageNet.[10] ImageNet is widely used in image processing and computer vision, which includes 14,197,122 images and 1.2 million images with SIFT features. Like the Flickr dataset, We generate ImageNet dataset with varying size from 200K to 1000K.
- **ImageNet-US**. The synthetic dataset ImageNet-US is generated by obtaining geographical locations from the US Board on Geographic Names[11] and randomly geo-tagging the images in ImageNet.

Some samples of Flickr and ImageNet dataset are shown in Fig. 7.

#### 2) WORKLOAD

A workload for reverse spatial visual top-*k* query experiments includes 100 input queries. The query locations are randomly selected from the locations of the underlying geo-objects. By default, the number of final (top-*k*) results $k = 3$; the image dataset size is 600K, which grows from 200K to 1000K; the parameter $\mu$ is set to 0.7; The number of query visual words is set to 100, which changes from 25 to 150. We report the average response time of 100 queries. The details of these parameters are presented in Table 2. All the experiments are run on a workstation with Intel(R) CPU Xeon 2.60GHz, 16GB memory and NVIDIA GeForce GTX 1080 GPU running Ubuntu 16.04 LTS Operation System. All query algorithms in the experiments are implemented in Java.

To the best of our knowledge, this work is the first time to investigate the problem of reverse spatial visual top-*k* query. In other words, there exists no method for this challenge. we compare the performance of the following approaches:

[6] http://www.rtreeportal.org

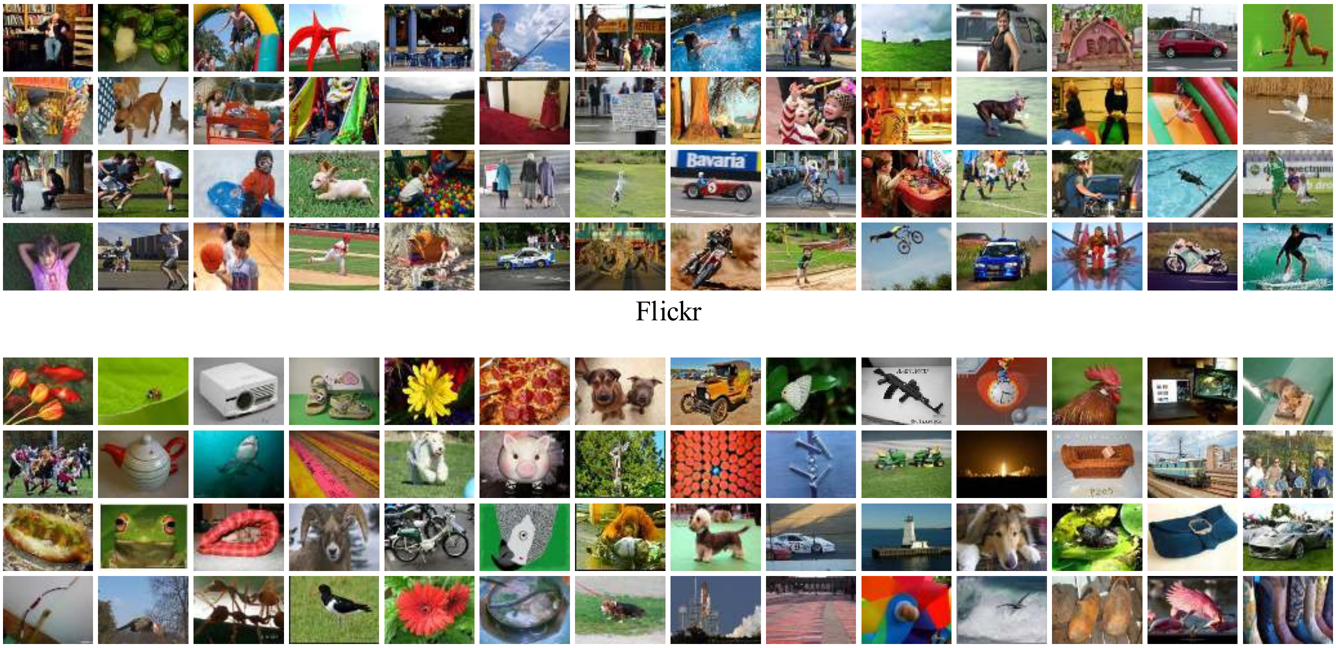
[7] http://www.flickr.com/

[8] http://geonames.usgs.gov

[9] http://geonames.usgs.gov

[10] http://image-net.org/index

[11] http://geonames.usgs.gov


Flickr

ImageNet

**FIGURE 7.** Some samples of Flickr and ImageNet dataset used in our experiments.

**TABLE 2.** The parameters evaluated in the experiments. The default values are shown in bold.

| Parameters | Values |
|---|---|
| Dataset size | 200K, 400K, **600K**, 800K, 1000K |
| Top-$k$ | 1, **3**, 5, 7, 9 |
| $\mu$ | 0, 0.1, 0.3, 0.5, **0.7**, 0.9, 1 |
| Number of query visual words | 25, 50, 75, **100**, 125, 150 |

- **RSVQ$_k$-R**. RSVQ$_k$-R is the baseline introduced in Section III-B, which employs R-Tree as the spatial index.
- **RSVQ$_k$-VR$^2$**. RSVQ$_k$-VR$^2$ is the proposed method introduced in Section IV-A.1, which employs VR$^2$-Tree as the spatial index.
- **RSVQ$_k$-CVR$^2$**. RSVQ$_k$-CVR$^2$ is the proposed method introduced in Section IV-A.3, which employs the extension of VR$^2$-Tree, i.e., CVR$^2$-Tree.
- **RSVQ$_k$-OptCVR$^2$**. RSVQ$_k$-OptCVR$^2$ is the proposed method which uses CVR$^2$-Tree with the optimization method introduced in Section IV-B.2.

As discussed above, the techniques of visual word generation used in the baseline is SIFT+BoVW. We utilize SIFT technique to extract local visual features of samples in the geo-image datasets, and then encode them into visual words vectors with a pre-learned vocabulary tree. The number of local visual features of each sample is from 1 to 300. For the proposed approaches, the pre-trained CNN model, i.e., AlexNet is used to learn the visual features. We fine-tune the AlexNet on the two geo-image datasets by stochastic gradient descent (SGD) algorithm. The momentum is set to 0.9 and weight decay is set to 0.0005. To prevent over-fitting, each layer is followed by a drop-out operation with a drop-out ratio of 0.5. After fine-tuning, the outputs of the first two fully-connected layers as the deep visual features, which are used to generate deep visual words vectors.

### *B. PERFORMANCE EVALUATIONS*

In this section, we evaluate the reverse search performance of the proposed approaches, i.e., RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$, and compare them with the baseline RSVQ$_k$-R on different size of geo-image datasets. Some search results of the proposed approaches are shown in Fig. 8. The images in green rectangle are the correct results and the failed cases are in the red rectangle.

#### 1) EVALUATION ON THE SIZE OF DATASETS

We evaluate the effect of varying the size of geo-image dataset on Flickr-RP, Flickr-US, ImageNet-RP and ImageNet-US, shown in Fig. 9 using log-scale. Obviously, the proposed algorithms outperform the baseline on these four datasets. Particularly, with the increasing of the dataset size, the efficiency of RSVQ$_k$-R declines dramatically because all the geo-images have to be considered for spatial visual top-$k$ search. By comparison, the performances of RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$ drop relatively slowly due to the efficiently spatial index and search algorithm.

To clearly demonstrates the trends of these proposed approaches, we draw the experimental data of RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$ via linear scale, shown in Fig. 10. For these four datasets, the performance of

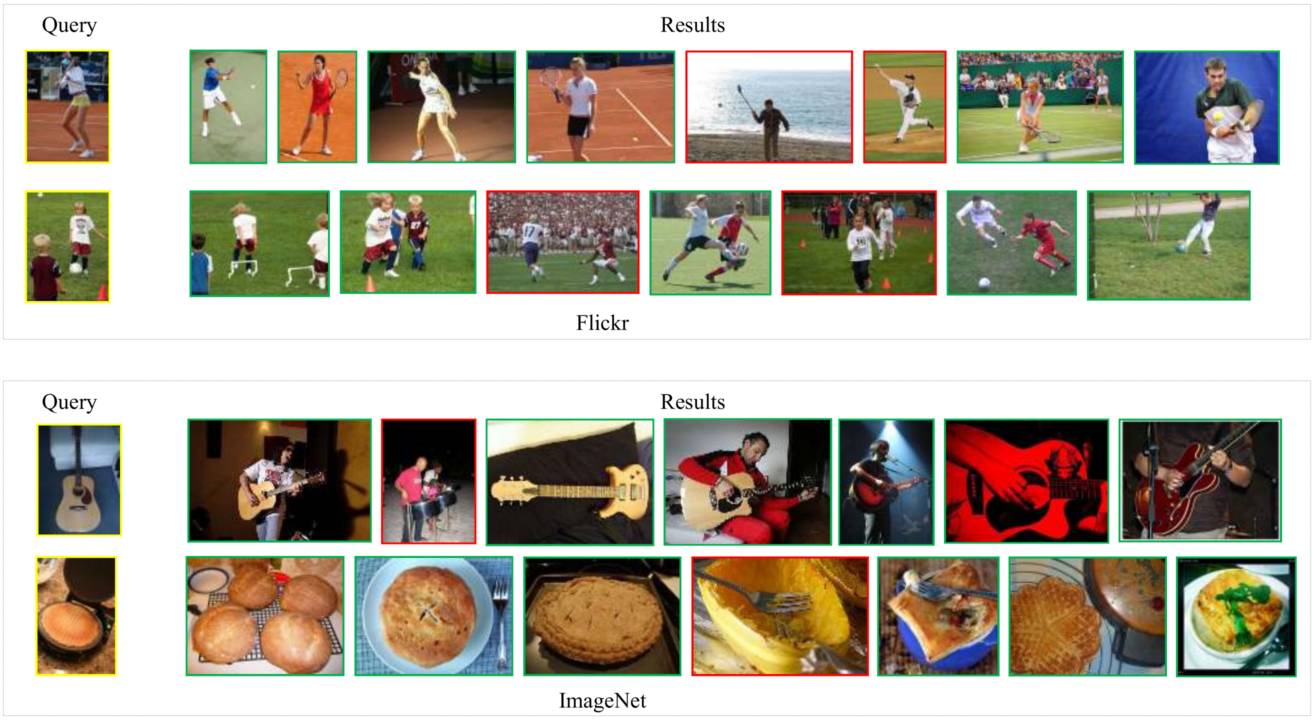
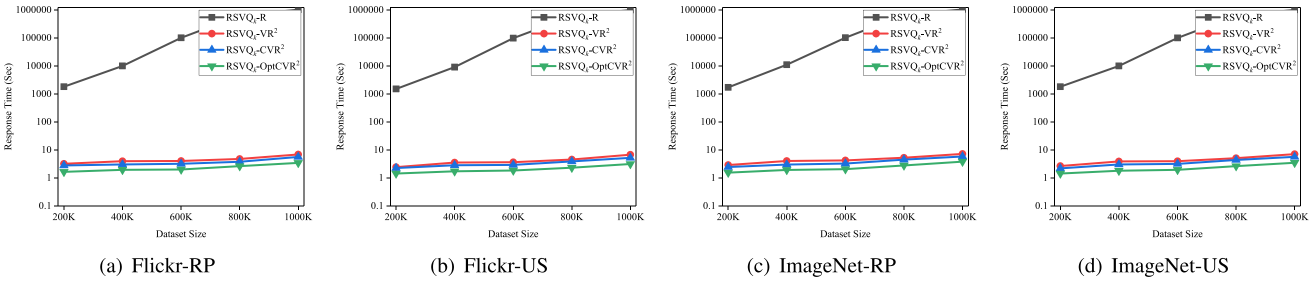
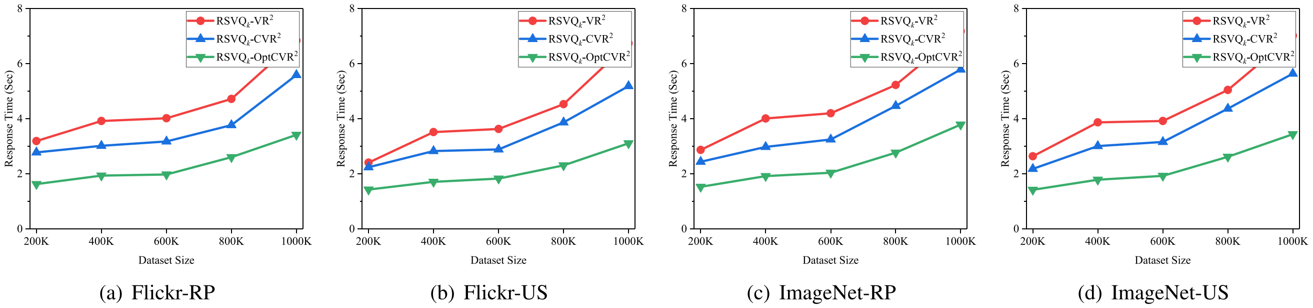


**FIGURE 8.** Some search results of the proposed approaches on Flickr and ImageNet. The images in green rectangle are the correct results and the failed cases are in the red rectangle.



(a) Flickr-RP (b) Flickr-US (c) ImageNet-RP (d) ImageNet-US

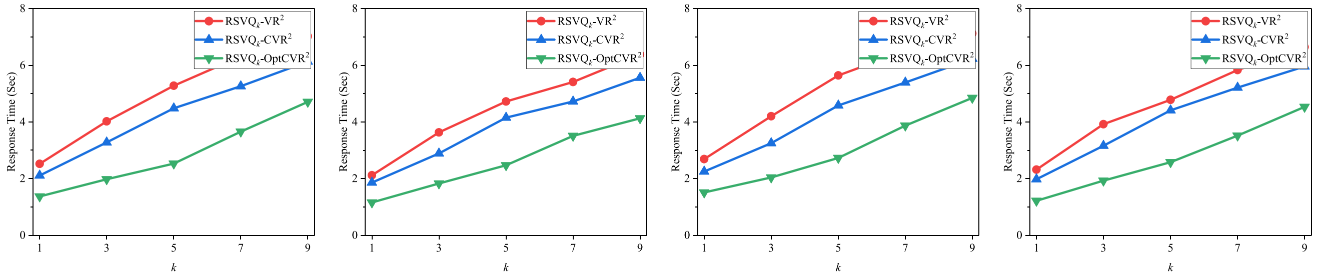
**FIGURE 9.** Evaluation on the size of geo-image datasets (log-scale).



(a) Flickr-RP (b) Flickr-US (c) ImageNet-RP (d) ImageNet-US

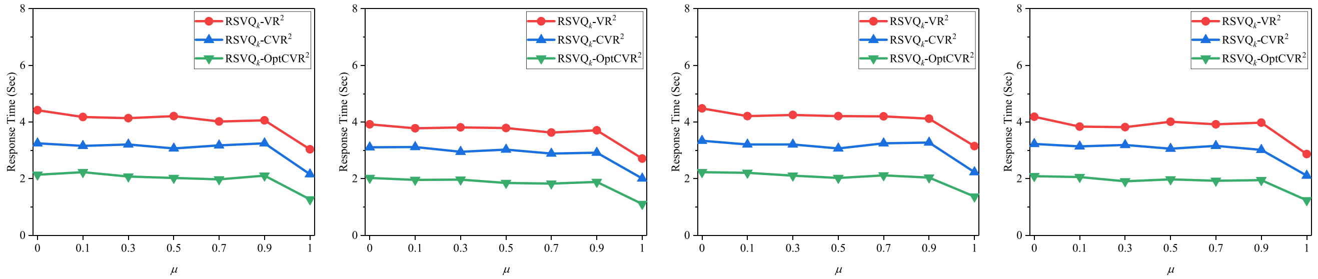
**FIGURE 10.** Evaluation on the size of geo-image datasets (linear-scale).

RSVQ$_k$-VR$^2$ is the lowest. Specifically, its response time is fluctuating upward in interval [200K, 800K], and after that it grows rapidly. By using the more efficient index, i.e., CVR$^2$-Tree, the algorithm RSVQ$_k$-CVR$^2$ can defeat the former. Similarly, the response time rises markedly when the dataset size is larger than 800K. Benefit from the optimization



(a) Flickr-RP (b) Flickr-US (c) ImageNet-RP (d) ImageNet-US

**FIGURE 11.** Evaluation on the number of results.



(a) Flickr-RP (b) Flickr-US (c) ImageNet-RP (d) ImageNet-US

**FIGURE 12.** Evaluation on the balance parameter $\mu$ in the similarity measurement.

technique, RSVQ$_k$-OptCVR$^2$ is the most efficient algorithm, whose growth rate of response time is the lowest as well. On Flickr-RP, it increases from 1.8 at 200K to nearly 3.2, which is similar to the situations on the other three datasets.

### 2) EVALUATION ON THE NUMBER OF RESULTS $k$

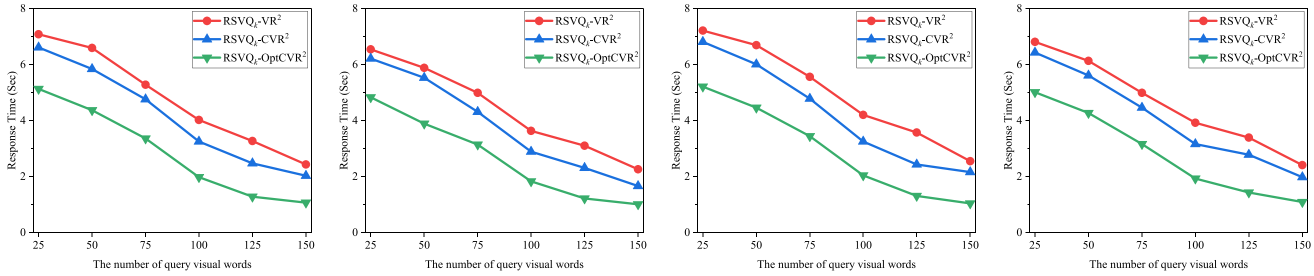
We evaluate the effect of varying the number of results $k$ on Flickr-RP, Flickr-US, ImageNet-RP and ImageNet-US, shown in Fig. 11. As the huge performance gap between the baseline and the three proposed algorithms, we do not plot the experimental data of RSVQ$_k$-R. Instead, we just show the differences of RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$. Beyond all doubt, the response time of all these algorithms increase gradually with the rise of $k$. Due to the optimization method, RSVQ$_k$-OptCVR$^2$ overcomes the others on all the four datasets. By comparison, without the optimization, the performance of RSVQ$_k$-CVR$^2$ is worse than the former, which shows an upward trend with fluctuation. Apparently, the response time of RSVQ$_k$-VR$^2$ is the highest since the promotion of efficiency by the VR$^2$-Tree is not larger than CVR$^2$-Tree, especially the applying of optimization technique.

### 3) EVALUATION ON THE BALANCE PARAMETER $\mu$

We evaluate the effect of varying the value of balance parameter $\mu$ in the similarity measurement on the four datasets. Like above experiments, we do not plot the data of RSVQ$_k$-R due to the enormous efficiency gap. On Flickr-RP dataset shown in Fig. 11(a), we can see clearly that the efficiency of RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$ are not obviously affected by changing $\mu$ in interval [0, 0.9]. Specifically, they move up and down slightly. However, when $\mu = 1$, the time cost of these algorithms drop down obviously because the visual similarity is ignored totally. As expected, RSVQ$_k$-OptCVR$^2$ wins this comparison by applying optimization via CVR$^2$-Tree. On Flickr-US, the runtime of these algorithms are slightly lower than the values on Flickr-RP, but the trends of them is very similar. They decline rapidly at $\mu = 1$. As expected, the situations on ImageNet-RP (Fig. 11(c)) and ImageNet-US (Fig. 11(d)) are very similar to the former two.

### 4) EVALUATION ON THE NUMBER OF QUERY VISUAL WORDS

In the last set of experiments, we evaluate the effect of varying the number of query visual words on these four datasets. The experimental results are illustrated in Fig. 13. By the same token, we do not consider the results of baseline and just show the differences between RSVQ$_k$-VR$^2$, RSVQ$_k$-CVR$^2$ and RSVQ$_k$-OptCVR$^2$. It is evident that the runtime of these algorithms decrease gradually as the number of query visual words increases. In particularly, the change rates of them in interval [25, 75] is a bit larger than the value in [100, 150]. The reason is that more visual words may enhance the pruning by diminishing the average visual similarity between query and geo-images. Same as those of the above sets of experiments, RSVQ$_k$-OptCVR$^2$ has the highest efficiency on all these datasets.



(a) Flickr-RP (b) Flickr-US (c) ImageNet-RP (d) ImageNet-US

**FIGURE 13. Evaluation on the number of query visual words.**

In summary, these experimental results demonstrate that the proposed spatial index VR$^2$-Tree, especially CVR$^2$-Tree with the optimization method can substantially improve the performance of reverse spatial visual search. The proposed search algorithm shows obvious superiority with the comparison to the baseline.

## VI. CONCLUSION

This paper investigates a novel search problem named RSVQ$_k$ query, which aims to retrieve a set of geo-image objects that have the query image as one of the most relevant images in both aspects of geographical proximity and visual similarity. To improve the search efficiency, a new hybrid index named VR$^2$-Tree and its extension is presented, which is a combination of visual representation of geo-image and R-Tree. Besides, the optimization method to tighter the bound via CVR$^2$-Tree is introduced. In addition, an efficient CVR$^2$-Tree based algorithm, named RSVQ$_k$ algorithm is careful developed, which can speed up the reverse search significantly. Comprehensive experiments are conducted on four geo-image datasets, and the results demonstrate that the proposed approach can address the RSVQ$_k$ problem effectively and efficiently.

## REFERENCES

[1] Y. Wang, X. Lin, L. Wu, and W. Zhang, "Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1393–1404, Jan. 2017.

[2] C. Zhang, Y. Lin, L. Zhu, A. Liu, Z. Zhang, and F. Huang, "CNN-VWII: An efficient approach for large-scale video retrieval by image queries," *Pattern Recognit. Lett.*, vol. 123, pp. 82–88, May 2019.

[3] C. Zhang, L. Zhu, J. Long, S. Lin, Z. Yang, and W. Huang, "A hybrid index model for efficient spatio-temporal search in HBase," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, Jun. 2018, pp. 108–120.

[4] P. Zhang, H. Lin, B. Yao, and D. Lu, "Level-aware collective spatial keyword queries," *Inf. Sci.*, vol. 378, pp. 194–214, Feb. 2017.

[5] X. Jin, S. Shin, E. Jo, and K.-H. Lee, "Collective keyword query on a spatial knowledge base," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 11, pp. 2051–2062, Nov. 2019.

[6] T. Guo, X. Cao, and G. Cong, "Efficient algorithms for answering the m-closest keywords query," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2015, pp. 405–418.

[7] K. Deng, X. Li, J. Lu, and X. Zhou, "Best keyword cover search," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 61–73, Jan. 2015.

[8] K. Yao, J. Li, G. Li, and C. Luo, "Efficient group top-k spatial keyword query processing," in *Proc. Asia–Pacific Web Conf.* Cham, Switzerland: Springer, Sep. 2016, pp. 153–165.

[9] X. Lin, J. Xu, and H. Hu, "Reverse keyword search for spatio-textual top-*k* queries in location-based services," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3056–3069, May 2015.

[10] Y. Gao, X. Qin, B. Zheng, and G. Chen, "Efficient reverse top-k Boolean spatial keyword queries on road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1205–1218, May 2015.

[11] S. Yang, M. A. Cheema, X. Lin, Y. Zhang, and W. Zhang, "Reverse k nearest neighbors queries and spatial reverse top-k queries," *VLDB J. Int. J. Very Large Data Bases*, vol. 26, no. 2, pp. 151–176, Apr. 2017.

[12] F. M. Choudhury, J. S. Culpepper, T. Sellis, and X. Cao, "Maximizing bichromatic reverse spatial and textual k nearest neighbor queries," *Proc. VLDB Endowment*, vol. 9, no. 6, pp. 456–467, Jan. 2016.

[13] P. Zhao, H. Fang, V. S. Sheng, Z. Li, J. Xu, J. Wu, and Z. Cui, "Monochromatic and bichromatic ranked reverse Boolean spatial keyword nearest neighbors search," *World Wide Web*, vol. 20, no. 1, pp. 39–59, Jan. 2017.

[14] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," *J. Comput. Syst. Sci.*, vol. 66, no. 4, pp. 614–656, Jun. 2003.

[15] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," 2017, *arXiv:1706.06064*. [Online]. Available: https://arxiv.org/abs/1706.06064

[16] A. S. Tarawneh, A. Hassanat, C. Celik, D. Chetverikov, M. S. Rahman, and C. Verma, "Deep face image retrieval: A comparative study with dictionary learning," 2018, *arXiv:1812.05490*. [Online]. Available: https://arxiv.org/abs/1812.05490

[17] P. Liu, J.-M. Guo, C.-Y. Wu, and D. Cai, "Fusion of deep learning and compressed domain features for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5706–5717, Dec. 2017.

[18] Y. Wang, X. Lin, L. Wu, and W. Zhang, "Effective multi-query expansions: Robust landmark retrieval," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 79–88.

[19] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[20] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 2, no. 1, pp. 1–19, 2006.

[21] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3939–3949, Nov. 2015.

[22] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," 2016, *arXiv:1608.05560*. [Online]. Available: https://arxiv.org/abs/1608.05560

[23] J. Long, L. Zhu, C. Zhang, Z. Yang, Y. Lin, and R. Chen, "Efficient interactive search for geo-tagged multimedia data," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30677–30706, Nov. 2019.

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[25] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, no. 2, Sep. 1999, pp. 1150–1157.

[26] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, p. 1470.

[27] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 4, Nov. 2004, pp. 506–513.

[28] E. Mortensen, H. Deng, and L. Shapiro, "A SIFT descriptor with global context," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2005, pp. 184–190.

[29] C. Li and L. Ma, "A new framework for feature descriptor based on SIFT," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 544–557, Apr. 2009.

[30] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Improving bag-of-visual-words image retrieval with predictive clustering trees," *Inf. Sci.*, vol. 329, pp. 851–865, Feb. 2016.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015, 2015.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[33] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[34] Y. Lecun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 253–256.

[35] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," 2016, *arXiv:1601.06759*. [Online]. Available: https://arxiv.org/abs/1601.06759

[36] C. Wang, H. Yang, C. Bartz, and C. Meinel, "Image captioning with deep bidirectional LSTMs," in *Proc. ACM Multimedia Conf. (MM)*, 2016, pp. 988–997.

[37] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 157–166.

[38] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, "Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 57–70, Jan. 2017.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[40] S. Matsuo and K. Yanai, "CNN-based style vector for style image retrieval," in *Proc. ACM Int. Conf. Multimedia Retr. (ICMR)*, 2016, pp. 309–312.

[41] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 241–257.

[42] M. Tan, S. Yuan, and Y. Su, "Content-based similar document image retrieval using fusion of CNN features," in *Proc. Int. Conf. Internet Multimedia Comput. Service*. Singapore: Springer, Aug. 2017, pp. 260–270.

[43] O. Seddati, S. Dupont, S. Mahmoudi, and M. Parian, "Towards good practices for image retrieval based on CNN features," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1246–1255.

[44] J. Yang, J. Liang, H. Shen, K. Wang, P. L. Rosin, and M.-H. Yang, "Dynamic match kernel with deep convolutional features for image retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5288–5302, Nov. 2018.

[45] C. Zhang, R. Chen, L. Zhu, A. Liu, Y. Lin, and F. Huang, "Hierarchical information quadtree: Efficient spatial temporal image search for multimedia stream," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30561–30583, Nov. 2019.

[46] C. Zhang, Y. Lin, L. Zhu, Z. Zhang, Y. Tang, and F. Huang, "Efficient region of visual interests search for geo-multimedia data," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30839–30863, 2019.

[47] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2011, pp. 373–384.

[48] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collective spatial keyword queries: A distance owner-driven approach," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 689–700.

[49] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, and M. L. Yiu, "Spatial keyword querying," in *Proc. Int. Conf. Conceptual Modeling*. Berlin, Germany: Springer, Oct. 2012, pp. 16–29.

[50] G. Cong and C. S. Jensen, "Querying geo-textual data: Spatial keyword queries and beyond," in *Proc. Int. Conf. Manage. Data*, Jun. 2016, pp. 2207–2212.

[51] C. Zhang, Y. Zhang, W. Zhang, and X. Lin, "Inverted linear quadtree: Efficient top k spatial keyword search," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 901–912.

[52] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, vol. 14, no. 2, 1984, pp. 47–57.

[53] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," *ACM SIGMOD Rec.*, vol. 19, no. 2, pp. 322–331, May 1990.

[54] Z. Li, K. C. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang, "IR-Tree: An efficient index for geographic document search," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 4, pp. 585–599, Apr. 2011.

[55] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems," in *Proc. 19th Int. Conf. Sci. Stat. Database Manage. (SSDBM)*, Jul. 2007, pp. 1–16.

[56] C. Zhang, Y. Zhang, W. Zhang, and X. Lin, "Inverted linear quadtree: Efficient top k spatial keyword search," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1706–1721, Jul. 2016.

[57] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 656–665.

[58] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial Web objects," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 337–348, Aug. 2009.

[59] J. B. Rocha, Jr., O. Gkorgkas, S. Jonassen, and K. Nørvåg, "Efficient processing of top-k spatial keyword queries," in *Proc. Int. Symp. Spatial Temporal Databases*. Berlin, Germany: Springer, Aug. 2011, pp. 205–222.

[60] D. Zhang, K.-L. Tan, and A. K. H. Tung, "Scalable top-k spatial keyword search," in *Proc. 16th Int. Conf. Extending Database Technol. (EDBT)*, 2013, pp. 359–370.

[61] D. Zhang, C.-Y. Chan, and K.-L. Tan, "Processing spatial keyword query as a top-k aggregation query," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 355–364.

[62] F. Korn and S. Muthukrishnan, "Influence sets based on reverse nearest neighbor queries," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 201–212, Jun. 2000.

[63] X. Xie, X. Lin, J. Xu, and C. S. Jensen, "Reverse keyword-based location search," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 375–386.

[64] A. Vlachou, C. Doulkeridis, Y. Kotidis, and K. Nørvåg, "Reverse top-k queries," in *Proc. IEEE 26th Int. Conf. Data Eng. (ICDE)*, Mar. 2010, pp. 365–376.

[65] M. Safar, D. Ibrahimi, and D. Taniar, "Voronoi-based reverse nearest neighbor query processing on spatial networks," *Multimedia Syst.*, vol. 15, no. 5, pp. 295–308, Oct. 2009.

[66] J. Lu, Y. Lu, and G. Cong, "Reverse spatial and textual k nearest neighbor search," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2011, pp. 349–360.

[67] M. A. Cheema, W. Zhang, X. Lin, Y. Zhang, and X. Li, "Continuous reverse k nearest neighbors queries in Euclidean space and in spatial networks," *VLDB J.*, vol. 21, no. 1, pp. 69–95, Feb. 2012.

[68] A. Vlachou, C. Doulkeridis, and K. Nøvåg, "Monitoring reverse top-k queries over mobile devices," in *Proc. 10th ACM Int. Workshop Data Eng. Wireless Mobile Access (MobiDE)*, 2011, pp. 17–24.

[69] M. A. Cheema, X. Lin, W. Zhang, and Y. Zhang, "Influence zone: Efficiently processing reverse k nearest neighbors queries," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 577–588.

[70] A. W. Yu, N. Mamoulis, and H. Su, "Reverse top-k search using random walk with restart," *Proc. VLDB Endowment*, vol. 7, no. 5, pp. 401–412, Jan. 2014.

[71] S. Wang, M. A. Cheema, and X. Lin, "Efficiently monitoring reverse k-nearest neighbors in spatial networks," *Comput. J.*, vol. 58, no. 1, pp. 40–56, Jan. 2015.

[72] S. Yang, M. A. Cheema, X. Lin, and Y. Zhang, "SLICE: Reviving regions-based pruning for reverse k nearest neighbors queries," in *Proc. IEEE 30th Int. Conf. Data Eng.*, Mar. 2014, pp. 760–771.

[73] C. Luo, L. Junlin, G. Li, W. Wei, Y. Li, and J. Li, "Efficient reverse spatial and textual k nearest neighbor queries on road networks," *Knowl.-Based Syst.*, vol. 93, pp. 121–134, Feb. 2016.

[74] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Jan. 1988.

[75] P. Zhao, X. Kuang, V. S. Sheng, J. Xu, J. Wu, and Z. Cui, "Scalable top-*k* spatial image search on road networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, Apr. 2015, pp. 379–396.

[76] C. Zhang, K. Cheng, L. Zhu, R. Chen, Z. Zhang, and F. Huang, "Efficient continuous top-k geo-image search on road network," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30809–30838, Nov. 2019.

[77] E. A. Fox, Q. F. Chen, A. M. Daoud, and L. S. Heath, "Order-preserving minimal perfect hash functions and information retrieval," *ACM Trans. Inf. Syst.*, vol. 9, no. 3, pp. 281–308, Jul. 1991.

[78] E. Achtert, H.-P. Kriegel, P. Kröger, M. Renz, and A. Züfle, "Reverse k-nearest neighbor search in dynamic and general metric databases," in *Proc. 12th Int. Conf. Extending Database Technol. Adv. Database Technol. (EDBT)*, 2009, pp. 886–897.

**LEI ZHU** was born in Changsha, China, in June 1988. He received the M.Sc. degree from Central South University, China, in 2014. He is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering, Central South University. His research interests are in the field of machine learning, deep learning, computer vision, and spatio-temporal data retrieval.

**JIAYU SONG** was born in Inner Mongolia Autonomous Region, China. He is currently pursuing the bachelor's degree in data science and big data technology with Central South University, in 2017. He main research interests include deep learning and machine learning.

**WEIREN YU** received the Ph.D. degree from the School of Computer Science and Engineering, University of New South Wales. He held a postdoctoral position with Imperial College. He is currently a Lecturer of computer science with the University of Warwick, and an Honorary Fellow at Imperial College. He has published more than 30 articles in DB and IR, and received three Best Paper Awards, two CiSRA Best Paper Awards, a One of the Best Papers of ICDE 2013, and a Best Student Paper Award. His research spans web search and information retrieval, graph data management, and streams data mining. He has served on various editorial boards, and as PC and an Active Reviewer of journals (e.g., IEEE TKDE, VLDB J, IEEE TIFS, ACM TKDD, WWWJ, and *Sensors*) and conferences (e.g., SIGIR, SIGMOD, VLDB, ICDE, EDBT, and CIKM).

**CHENGYUAN ZHANG** was born in Hunan, China. He received the B.S. degree from Sun-Yat sen University, in 2008, and the master's and Ph.D. degrees in computer science from the University of New South Wales, in 2011 and 2015, respectively. He is currently a Lecturer with the School of Computer Science and Engineering, Central South University, China. His main research interests include information retrieval, query processing on spatial data, and multimedia data.

**HAO YU** was born in Shangrao, China, in December 1994. He received the M.Sc. degree from Guangxi Normal University, China, in 2018. He is currently pursuing the Ph.D. degree in computer science and technology with Central South University. His research interests are in the field of image retrieval, machine learning, computer vision, and crowdsourcing learning.

**ZUPING ZHANG** received the B.S. degree in foundation of mathematics from Hunan Normal University, in 1989, the M.S. degree from the Foundation of Mathematics, Jilin University, in 1992, and the Ph.D. degree in computer application technology from Central South University, in 2005. He is currently a Professor with the School of Information Science and Engineering, Central South University, Changsha, China. His current research interests include information fusion and information systems, bigdata technology and application, parameter computing, and biology computing.