IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 8, 2019, accepted October 8, 2019, date of publication October 21, 2019, date of current version November 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948388

SVS-JOIN: Efficient Spatial Visual Similarity Join for Geo-Multimedia

LEI ZHU[1], WEIREN YU[2,3], CHENGYUAN ZHANG[1], ZUPING ZHANG[1], FANG HUANG[1], AND HAO YU[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 author: 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 In the big data era, massive amount of multimedia data with geo-tags has been generated and collected by smart devices equipped with mobile communications module and position sensor module. This trend has put forward higher request on large-scale geo-multimedia retrieval. Spatial similarity join is one of the significant problems in the area of spatial database. Previous works focused on spatial textual document search problem, rather than geo-multimedia retrieval. In this paper, we investigate a novel geo-multimedia retrieval paradigm named spatial visual similarity join (SVS-JOIN for short), which aims to search similar geo-image pairs in both aspects of geo-location and visual content. Firstly, the definition of SVS-JOIN is proposed and then we present the geographical similarity and visual similarity measurement. Inspired by the approach for textual similarity join, we develop an algorithm named SVS-JOIN$_B$ by combining the PPJOIN algorithm and visual similarity. Besides, an extension of it named SVS-JOIN$_G$ is developed, which utilizes spatial grid strategy to improve the search efficiency. To further speed up the search, a novel approach called SVS-JOIN$_Q$ is carefully designed, in which a quadtree and a global inverted index are employed. Comprehensive experiments are conducted on two geo-image datasets and the results demonstrate that our solution can address the SVS-JOIN problem effectively and efficiently.

INDEX TERMS Geo-image, geographical similarity, similarity join, visual similarity.

I. INTRODUCTION

In the big data era, online social networking services, search engine and multimedia sharing services are rapidly growing in popularity, which generate, collect and store large-scale multimedia data [1]–[4], e.g., texts, images, audios and videos. For example, we use online social networking services such as Facebook,[1] Twitter,[2] Linkedin,[3] Weibo,[4] etc. to make friends, sharing hobbies and work information by posting texts, uploading images or short videos. On the other hand, for multimedia data [5] sharing platforms such as Flickr,[5] more than 3.5 million new images posted online every day in March 2013. Every minute there are 100 hours of videos uploaded to YouTube,[6] and more than 2 billion videos totally stored in this platform by the end of 2013. In China, IQIYI[7] is the largest video sharing web site. The total watch time monthly of this online video service exceeded 42 billion minutes. These multimedia online services not only provide great convenience for us, but create possibilities for the generation, collection, storage and sharing of large-scale multimedia data [6], [7]. Moreover, this trend has put forward greater challenges for massive multimedia data retrieval [8], [9].

Smartphones and tablets equipped with communications module (e.g., WiFi and 4G module) and position sensor module (e.g., GPS-Module) collect huge amounts of multimedia

The associate editor coordinating the review of this manuscript and approving it for publication was Pradeep Kumar Gupta.

[1]https://facebook.com/
[2]http://www.twitter.com/
[3]https://www.linkedin.com/
[4]https://weibo.com/
[5]https://www.flickr.com/
[6]https://www.youtube.com/
[7]http://www.iqiyi.com/

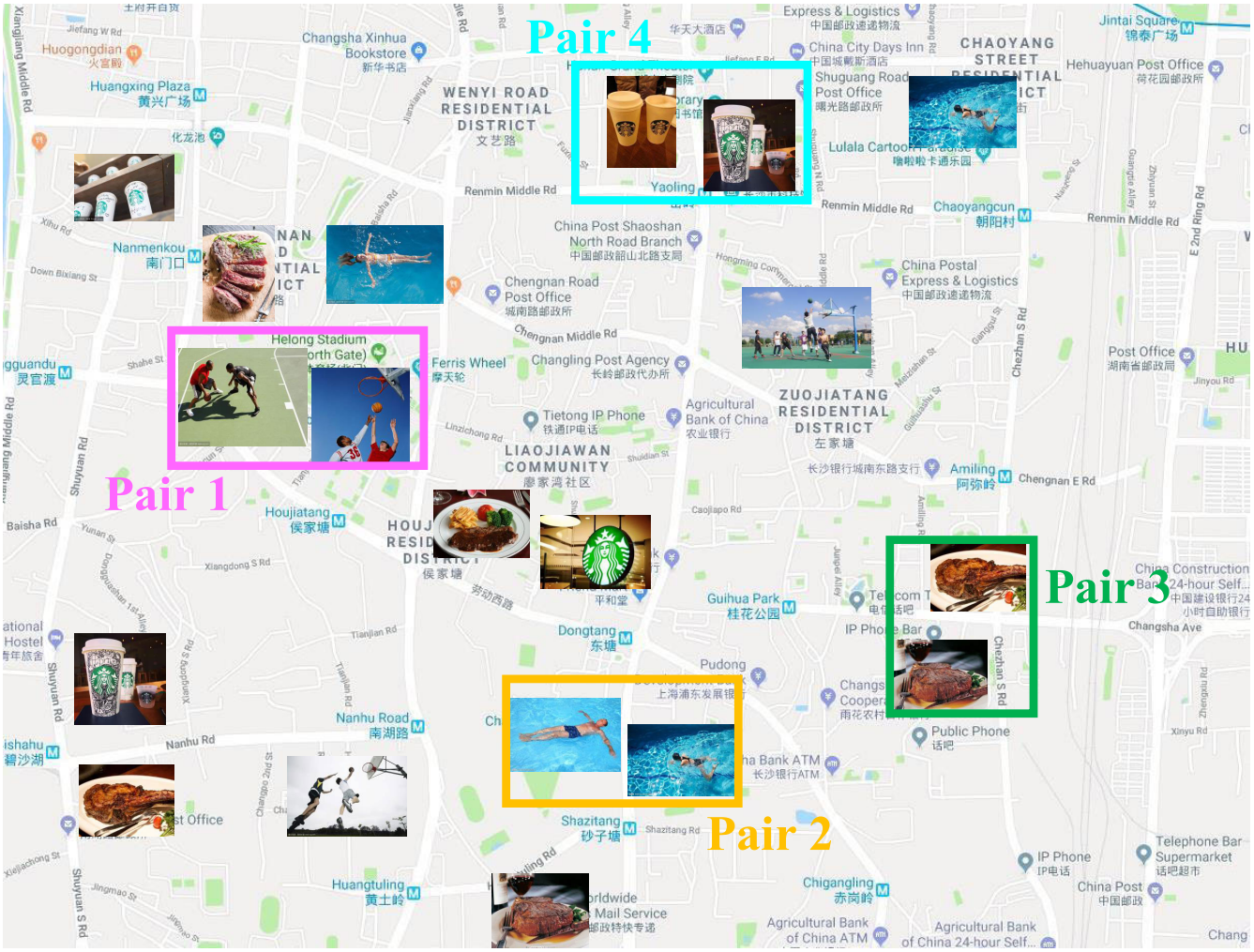


FIGURE 1. An example of spatial visual similarity join.

data [10]–[12] with geo-tags. For example, users can take photos or videos [13], [14] with the geo-location information. Besides, many mobile applications such as WeChat, Twitter and Instagram support posting, storing and sharing geo-multimedia data. Other location-based services such as Google Places, Yahoo!Local, and Dianping provide the convenient query services by taking into account both geographical proximity and multimedia content similarity.

Motivation: Due to the wide application of smart devices and location-based services, spatial textual search problem has become a hot spot in the area of spatial database and information retrieval. Lots of spatial indexing techniques have been proposed to support efficient query, such as R-Tree [15], R^+-Tree [16], R^*-Tree [17], KR^*-Tree [18], IR^2-Tree [19] etc. More recently, Deng *et al.* [20] studied a generic version of closest keywords search called best keyword Cover. Cao *et al.* [21] proposed the problem of collective spatial keyword querying, Fan *et al.* [22] studied the problem of spatio-textual similarity search for regions of interests query. Zhang *et al.* [23] proposed IL-Quadtree to address top-k spatial keyword search problem efficiently. However, these researches just only consider the textual data such as keywords, they do not take into account the content of multimedia data, e.g. images. One of the significant geo-textual search problems, spatial textual similarity join, is to find out the spatial textual object pairs that are similar in both aspects of geo-location and textual content simultaneously. It has attracted wide attention such as [24]–[29]. Nevertheless, there is no work pays attention to geo-multimedia data for this task. In this paper, we aim to investigate a novel paradigm named **S**patial **V**isual **S**imilarity **JOIN** (**SVS-JOIN**) and develop an efficient solution to overcome the challenge of geo-multimedia query. Fig. 1 is a simple but intuitive example to describe this problem.

Example 1: As illustrated in Fig. 1, the spatial visual similarity join can be applied in friends recommendation services on online social networking services. According to the geo-images posted by the users, the social networking

system can find the similar geo-images in both aspects of geo-location and visual content. It is easy to understand that two people may make friend if they have the same hobbies and their position is very close. There are four similar geo-image pairs in Fig. 1 are searched out. For pair 1 shown in magenta rectangle, two users who took the photos about basketball at two close places are very likely to become good friends.

To the best of our knowledge, this paper is the first time to study the SVS-JOIN problem. We introduce the definition of spatial visual similarity join in formal and present the relevant notions. Besides, we discuss how to measure geographical similarity and visual similarity to find the similar geo-image pairs. To measure visual similarity accurately, we employ two types of visual features to generation visual representations of geo-images: (1) the traditional hand-crafted visual features named Scale-Invariant Features Transform (SIFT for short) and (2) deep visual features extracted by convolutional neural networks (CNN for short). The former is named SIFT-BoVW and the latter is called Deep-BoVW. To combat this challenge effectively and efficiently, an algorithm called SVS-JOIN$_B$ inspired by the techniques used in textual similarity join is introduced. Based on it, we develop an extension of SVS-JOIN$_B$ called SVS-JOIN$_G$ which uses spatial grid partition strategy to improve the efficiency. In order to further improve the search performance, a novel method named SVS-JOIN$_Q$ is carefully designed, which is based on quadtree and global inverted index to speed up search.

Contributions: Our main contributions can be summarized as follows:

- To the best of our knowledge, this work is the first to study the problem of spatial visual similarity join. We propose the definition of geo-image, SVS-JOIN and relevant notions. The visual similarity function and geographical similarity function are designed for similar geo-image pair search.
- For the visual representation of geo-images, we propose to employ CNN to extract deep visual features for visual words generation, rather than hand-crafted visual features. We call this method Deep-BoVW that is a combination of deep CNN techniques, Bag-of-Visual-Words and k-means clustering method. As far as we know, there is no existing research that uses the combination of CNN and BoVW to address the image JOIN problem.
- We introduce an algorithm named SVS-JOIN$_B$ inspired by the techniques used for the problem of textual similarity join. An extension of SVS-JOIN$_B$ called SVS-JOIN$_G$ is developed which utilizes spatial grid partition technique to speed up search. To further improve the searching performance, we present a novel method named SVS-JOIN$_Q$ that is based on quadtree partition technique and a global inverted index.
- We have conducted comprehensive experiments on real geo-image dataset. Experimental results demonstrate that our approach has really high performance.

Roadmap: The remainder of this paper is organized as follows: In Section II we introduce the previous researches concerning content-based image retrieval, spatial textual search and set similarity joins, which are related to this work. In Section III, we propose the definition of spatial visual similarity join and relevant concepts. In Section IV, two visual representations named SIFT-BoVW and Deep-BoVW are presented. Besides, we introduce a baseline and an extension named SVS-JOIN$_B$ and SVS-JOIN$_G$ respectively. In addition, a novel algorithm named SVS-JOIN$_Q$ is proposed, in which a combination of a quadtree and a global inverted indexing structure is employed to solve the SVS-JOIN problem more efficiently. In Section V, we present the experiment results. Finally, we conclude the paper in Section VI.

II. RELATED WORK

In this section, we introduce the previous studies of content-based image retrieval, spatial textual search and set similarity joins, which are relevant to this work. To the best of our knowledge, no priori work on this problem.

A. CONTENT-BASED IMAGE RETRIEVAL

1) CBIR VIA SIFT

As one of the most important problems, content-based image retrieval (CBIR for short) [30]–[32] has gained much attention of many researchers in multimedia area [33]–[35]. Scale-Invariant Features Transform (SIFT for short) [36], [37] is one of the conventional methods for visual feature extraction. It transforms an image into a collection of local feature vectors. These features are invariant to translation, scaling, rotation, and partially invariant to illumination changes. In recent years, lots of works have been proposed using SIFT to overcome CBIR challenges. For example, Mortensen *et al.* [39] proposed a feature descriptor which augments SIFT with a global context vector that adds curvilinear shape information from a much larger neighborhood. Ke *et al.* [40] proposed a SIFT and PCA based method to encode the salient aspects of the image gradient in the neighborhood of feature point. Su *et al.* [41] presented horizontal or vertical mirror reflection invariant binary descriptor named MBR-SIFT to solve the problem of image matching. To gain sufficient distinctiveness and robustness for the task of feature matching, Li and Ma [42] designed a novel SIFT based feature descriptor by integrating color and global information. Zhu *et al.* [43] proposed an image registration algorithm called BP-SIFT by using belief propagation, which has significant improvement for the problem of keypoint matching.

2) CBIR VIA BoVW

Originated from text retrieval and mining, BoVW is an important visual representation method in multimedia retrieval and computer vision [5], [11], [44], [45]. BoVW [46] is a conventional image representation model, which is to improve the performance of image feature matching markedly. For image retrieval problem, it generates visual words by utilizing k-means method to cluster SIFT features. Escalante *et al.* [47]

presented an evolutionary algorithm to implement an automatically learning weighting schemes of this model for computer vision tasks. Dimitrovski *et al.* [48] proposed to use predictive clustering trees (PCTs) to improve the BoVW image retrieval in the large-scale image database. Mandal *et al.* [49] proposed a patch-based framework by using SIFT descriptor and BoVW model to improve the performance of handwritten signature detection. Based on S-BoVW paradigm, dos Santos *et al.* [50] proposed a novel method that considers information of texture to generate textual signatures of image blocks. For the task of Medical image retrieval, Zhang *et al.* [51] proposed a BoVW based medical image retrieval approach named PD-LST retrieval to identify discriminative characteristics between different medical images with pruned dictionary. Karakasis *et al.* [52] proposed a BoVW based framework for image retrieval, which uses affine image moment invariants as descriptors of local image areas.

3) CBIR VIA DEEP LEARNING

As a powerful tool, deep learning [53]–[55] is widely used to solve image retrieval [56]–[58] and computer vision problems. In 2012, AlexNet [59] proposed by Krizhevsky *et al.* significantly improves the accuracy of image retrieval. More recently, lots of deep learning based researches have been proposed for CBIR task. Gordo *et al.* [60] proposed to generate compact global signatures via CNN for image retrieval. Fu *et al.* [61] utilized CNN to generate visual features and employed SVM for classification. Tzelepi *et al.* [62] proposed a novel CNN based approach that exploits the data label information to generate better descriptors for image retrieval. For style based image retrieval task, Matsuo and Yanai [63] proposed to use style vector that is transformed from CNN based style matrix. Zhou *et al.* [64] proposed a CNN-based match kernel to encode CNN feature and SIFT feature to improve the accuracy. Liu *et al.* [65] combined high-level CNN features and low-level DDBTC features to generate two-layer codebook features for performance boosting. Seddati *et al.* [66] proposed to improve Regional Maximal Activation (RMAC) approach by combined multi-scale and multi-layer feature extraction of different RMAC extensions. Yang *et al.* [67] utilized a dynamic match kernel with deep CNN features to search images with different content details but similar semantics. Shimoda and Yanai [68] tested simple, siamese and triplet CNN to generate good visual features for food image retrieval. For some specific application, Nakazawa and Kulkarni [69] presented a CNN based image classification method to solve wafer maps defect pattern recognition issue. Sarraf and Tofighi [70] employed CNN to recognize fMRI image of Alzheimer's brain from normal healthy brain.

It is no doubt that these solutions improve the performance of image retrieval and visual feature matching significantly. However, these works cannot solve the problem of geo-multimedia data retrieval as they have no effective processing for geographical distance measurement.

B. SPATIAL TEXTUAL SEARCH

1) SPATIAL TEXTUAL QUERY

Due to the collection and storage of large scale spatial textual data, there has been increasing interest on spatial textual search problem [71], [72]. Spatial textual search [23], [73], [74] aims to retrieve textual objects or documents with geo-tags by textual similarity and geographical proximity. For top-k spatial keyword queries, Rocha-Junior *et al.* [75] proposed a novel spatial index named Spatial Inverted Index (S2I for short) to enhance the efficiency of search. Li *et al.* [76] proposed an efficient indexing structure named IR-tree, which enables spatial pruning and textual filtering to be performed simultaneously. Zhang *et al.* [77] presented a scalable integrated inverted index called I^3 that uses the Quadtree to hierarchically partition the data space into cells. Zhang *et al.* [78] proposed an efficient index named inverted linear quadtree (IL-Quadtree for short) and designed a novel algorithm to improve the performance of query. Li *et al.* [79] presented BR-tree to solve the problem of keyword-based k-nearest neighbor queries. They utilized R-tree to maintain the spatial information of objects and exploited B-tree to main the terms in the objects. Fan *et al.* [22] proposed grid-based signatures and threshold-aware pruning techniques to address spatio-textual similarity search problem. Zhang *et al.* [80] proposed to model the spatial keyword search problem as a top-k aggregation problem. They developed a rank-aware CA algorithm that works well on inverted lists sorted by textual relevance and spatial curving order. Wang *et al.* [81] proposed an efficient technique named AP-Tree to solve the problem of continuous spatial-keyword queries over streaming data. Zhang *et al.* [82] introduced m-closest keywords (mCK for short) query that aims to search out the spatially closest tuples which match m user-specified keywords. Guo *et al.* [83] proposed another solution to solve the mCK search problem. They devised a novel greedy algorithm named *SKEC* that has an approximation ratio of 2 and in addition, they developed two approximation algorithms called *SKECa* and *SKECa+* respectively to improve the efficiency.

2) SET SIMILARITY JOINS

In recent years, lots of researchers paid attentions on the problem of spatial textual similarity join [24], [84], [85]. A spatial similarity join of two spatial databases aims to search out pairs of objects that are simultaneously similar in both aspects of textual and spatial. Ballesteros *et al.* [25] proposed an algorithm based on MapReduce parallel programming model to solve this problem on large-scale spatial databases. Efstathiades *et al.* [26] propose the problem of Spatio-Textual Point-Set Join query and extended the existing methods to solve the spatial-textual joins problem of point sets. Hu *et al.* [27] introduced a signature-based join framework that prunes large numbers of dissimilar pairs to enhance the search efficiency. To overcome the issue of large number of duplicates, Rong *et al.* [28] introduced a novel duplicate free framework with three filtering methods to

TABLE 1. The summary of notations.

Notation	Definition
$\mathcal{D}_I$	A given database of geo-images
$\lvert\mathcal{D}_I\rvert$	The number of geo-images in $\mathcal{D}_I$
I_i	The i-th geo-tagged images
$I_i.G$	The geographical information component of I_i
$I_i.V$	The visual component of I_i
X	A longitude
Y	A latitude
v_i	A visual word
$\mathcal{R}_I$	A dataset of geo-images
I_i^r	The i-th geo-tagged images in dataset $\mathcal{R}$
$w(v)$	The weight of visual word v
Γ_G	The geographical similarity threshold
Γ_V	The visual similarity threshold
$\mathcal{P}$	A result set
$GeoSim(I_i, I_j)$	The geographical similarity between I_i and I_j
$VisSim(I_I, i_J)$	The visual similarity between I_i and I_j
$EucDis(I_I, i_J)$	The Euclidean distance between I_i and I_j
$MaxDis(\mathcal{R}, \mathcal{S})$	The maximum Euclidean distance between any two geo-iamges from $\mathcal{R}$ and $\mathcal{S}$
$\mathcal{W}$	A global word set
$\boldsymbol{\xi}_n$	The SIFT feature vector of n-th geo-image
$\boldsymbol{\zeta}_n$	The deep convolutional feature vector of n-th geo-image
$\mathcal{G}$	The clustering set
$\eta(\cdot)$	The number of occurrences of a visual word in an image
$\boldsymbol{\chi}$	The mean vector of a cluster
$\boldsymbol{\theta}$	The network parameters of a CNN model
$\mathcal{L}$	A inverted list
$\Omega_o[\hat{o}]$	The number of word overlap of o with $\hat{o}$
Φ	A global word ordering
$Pf(I.V)$	The prefix of $I.V$
C_i	A cell with id i
N_i	A node of quadtree with id i
$\mathcal{I}_G$	A inverted index set

prune dissimilar string pairs without computing their similarity scores. Shang *et al.* [29] presented a knowledge hierarchy based filter-and-verification framework to efficiently identify the similar pairs to address knowledge-aware similarity join problem.

These spatial textual search and similarity joins approaches only consider the textual and spatial information, that means they cannot be directly applied to address geo-image joins problem even if they raise search efficiency substantially. Thus, this paper proposes to combine geographical information and visual representations of geo-images to construct efficient search algorithms for spatial visual similarity joins problem.

III. PRELIMINARIES

In this section, we propose the definition of spatial visual similarity joins (SVS-JOIN) at the first time, then present the geographical and visual similarity measurement. Besides, we briefly introduce the SIFT and CNN techniques respectively, which are the base of our work. Table 1 summarizes the notations frequently used throughout this paper to facilitate the discussion.

A. PROBLEM DEFINITION

Definition 1 (Geo-Image): Let $\mathcal{D}_I = \{I_1, I_2, \ldots, I_{|\mathcal{D}_I|}\}$ be a geo-image dataset, $|\mathcal{D}_I|$ denotes the size of $\mathcal{D}_I$. A geo-image $I_i \in \mathcal{D}_I$ is defined as a tuple $I_i =< I_i.G, I_i.V >$, where $I_i.G$ is the geographical information component that is generated from the geo-tag of this image. More specifically, it consists of longitude X and latitude Y, i.e., $I_i.G = (X, Y)$. Another part, $I_i.V$, is the visual information component that consists of a visual word set $I_i.V = \{v_1, v_2, \ldots, v_n\}$ modeled by BoVW. i is the id of the geo-image.

Consider two geo-image datasets $\mathcal{R}_I = \{I_1^r, I_2^r, \ldots, I_{|\mathcal{R}_I|}^r\}$ and $\mathcal{S}_I = \{I_1^s, I_2^s, \ldots, I_{|\mathcal{S}_I|}^s\}$, similar to spatial textual similarity join, a spatial visual similarity join aims to retrieval all pairs of geo-images from $\mathcal{R}_I$ and $\mathcal{S}_I$ respectively, which are similar enough in both aspects of geo-location and visual content. We introduce two thresholds, i.e., geographical similarity threshold and visual similarity threshold to measure these two similarity. Specifically, for each pair, both of the geographical similarity and visual similarity of these two geo-images are less than geographical similarity threshold and visual similarity threshold. To clarify our work more clearly, we propose the definition of spatial visual similarity join as follows.

Definition 2 (Spatial Visual Similarity Join (SVS-JOIN)): Given two geo-image datasets $\mathcal{R}_I = \{I_1^r, I_2^r, \ldots, I_{|\mathcal{R}_I|}^r\}$ and $\mathcal{S}_I = \{I_1^s, I_2^s, \ldots, I_{|\mathcal{S}_I|}^s\}$, geographical similarity threshold Γ_G and visual similarity threshold Γ_V. A spatial visual similarity join denoted as SVS-JOIN$(\mathcal{R}, \mathcal{S}, \Gamma_G, \Gamma_V)$ returns a set of geo-image pairs $\mathcal{P} \subseteq \mathcal{R} \times \mathcal{S}$, in which each pair contains two highly similar geo-images in both aspect of geo-location and visual content, i.e.,

$$\begin{aligned}\mathcal{P} = \{(I_i^r, I_j^s) | &GeoSim(I_i^r, I_j^s) \leq \Gamma_G, \\ &VisSim(I_i^r, I_j^s) \geq \Gamma_V, \\ &\forall I_i^r \in \mathcal{R}_I, I_j^s \in \mathcal{S}_I\}\end{aligned} \tag{1}$$

where $GeoSim(I_i^r, I_j^s)$ and $VisSim(I_i^r, I_j^s)$ are the geographical similarity function and visual similarity function respectively.

To measure these two similarities quantitatively, we utilize Euclidean distance measurement and Jaccard distance measurement to implement these two functions, shown as follows.

Definition 3 (Geographical Similarity Function): Given two geo-image datasets $\mathcal{R}_I = \{I_1^r, I_2^r, \ldots, I_{|\mathcal{R}_I|}^r\}$ and $\mathcal{S}_I = \{I_1^s, I_2^s, \ldots, I_{|\mathcal{S}_I|}^s\}$, $\forall I_i^r \in \mathcal{R}_I, I_j^s \in \mathcal{S}$, the geographical similarity between I_i^r and I_j^s is measured by the following similarity function:

$$GeoSim(I_i^r, I_j^s) = \frac{EucDis(I_i^r, I_j^s)}{MaxDis(\mathcal{R}, \mathcal{S})} \tag{2}$$

where $EucDst(I_i^r, I_j^s)$ is the Euclidean distance between I_i^r and I_j^s, which is measured by the following function:

$$EucDis(I_i^r, I_j^s) = \sqrt{(I_i^r.G.X - I_j^s.G.X)^2 + (I_i^r.G.Y - I_j^s.G.Y)^2} \tag{3}$$

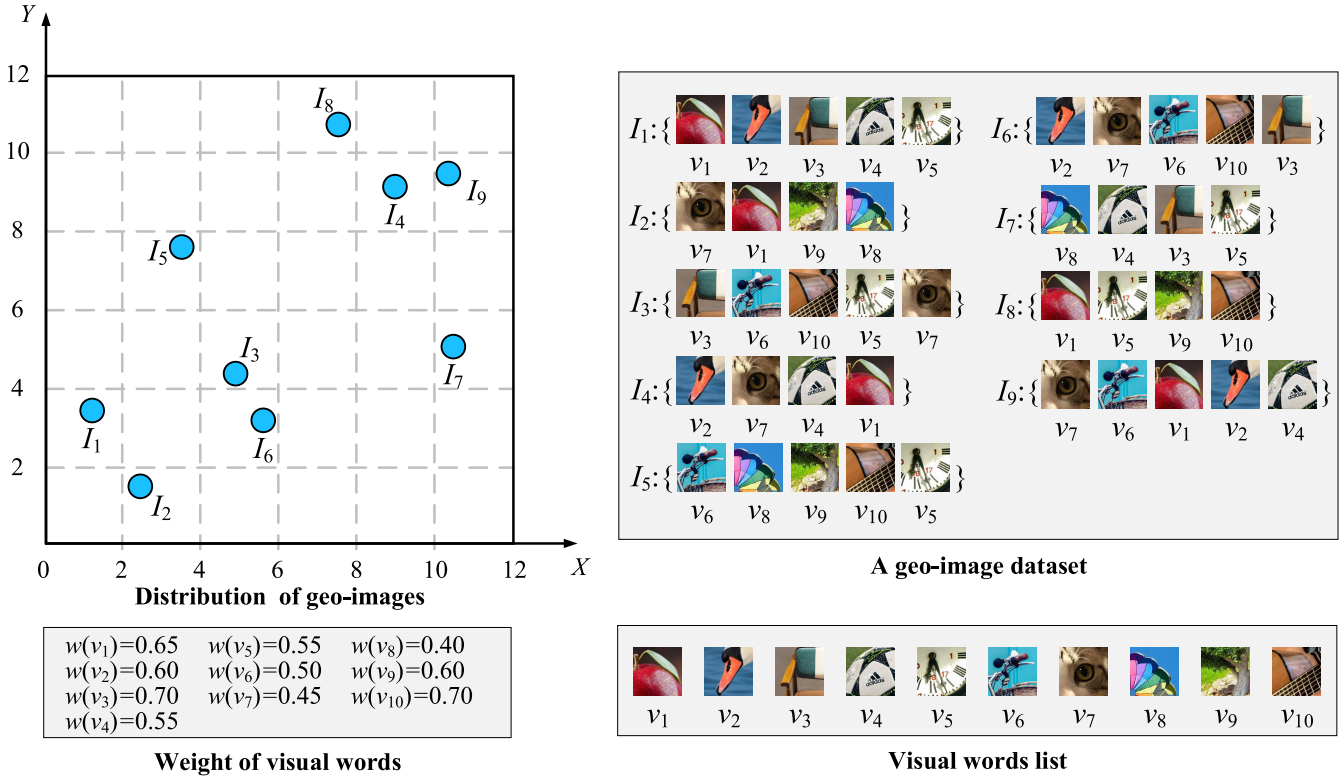


FIGURE 2. An example of geo-images and spatial visual similarity join.

the function $MaxDis(\mathcal{R}, \mathcal{S})$ is to return the maximum Euclidean distance between any two geo-images from $\mathcal{R}$ and $\mathcal{S}$ respectively, which is described in formal as follows:

$$MaxDis(\mathcal{R}, \mathcal{S}) = max(\{EucDis(I_i^r, I_j^s) | I_i^r \in \mathcal{R}, I_j^s \in \mathcal{S}\}) \quad (4)$$

where the function $max(\cdot)$ is to return the maximum element from a set.

Definition 4 (Visual Similarity Function): Given two geo-image datasets $\mathcal{R}_I = \{I_1^r, I_2^r, \ldots, I_{|\mathcal{R}_I|}^r\}$ and $\mathcal{S}_I = \{I_1^s, I_2^s, \ldots, I_{|\mathcal{S}_I|}^s\}$, $\forall I_i^r \in \mathcal{R}_I, I_j^s \in \mathcal{S}$, the visual similarity between I_i^r and I_j^s is measured by the following similarity function:

$$VisSim(I_i^r, I_j^s) = \frac{\sum_{v \in I_i^r.V \cap I_j^s.V} w(v)}{\sum_{v \in I_i^r.V \cup I_j^s.V} w(v)} \quad (5)$$

where $w(v)$ represents the weight of the visual word v. In this work, we measure the weight of visual word by term frequency-inverse document frequency *tf-idf* [86].

Assumption: For ease of discussion, here we assume that $\mathcal{R} = \mathcal{S}$. Our approach can be applied well in the case of $\mathcal{R} \neq \mathcal{S}$. Therefore, for a geo-image dataset $\mathcal{R}$, we denote a spatial visual similarity join as SVS-JOIN($\mathcal{R}, \Gamma_G, \Gamma_V$).

Example 2: We give a simple example to describe how to perform a SVS-JOIN search. Consider a geo-image dataset $\mathcal{R} = \{I_1, I_2, \ldots, I_9\}$, shown in Fig. 2. The upper left figure is the geographical distribution of the geo-images in $\mathcal{R}$. The upper right figure shows this geo-image dateset, in which each geo-image is represented by a set of visual words. At the right bottom is the list of these visual words, and their weights are shown at the left bottom. We set $\Gamma_G = 0.3$ and $\Gamma_V = 0.4$, the SVS-JOIN($\mathcal{R}$, 0.3, 0.4) returns the set $\mathcal{P} = \{(I_3, I_6), (I_4, I_9)\}$.

B. SCALE-INVARIANT FEATURES TRANSFORM

Our first visual representation scheme uses SIFT [36], [37]. This conventional technique aims to transform an image into a large set of local feature vectors, which are invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. It has four main phases:

1) SCALE-SPACE EXTREMA DETECTION

The first phase is called scale-space extrema detection. This method searches all the images in scale space, which is to identify potential points of interest that are invariant to scale and orientation by utilizing difference-of-Gaussian (DoG) function.

2) KEYPOINT LOCALIZATION

The second phrase is named keypoint localization, which is to select and localize the keypoints according to their stability. At each candidate location, a fine fitting model is used to determine the location and scale.

3) ORIENTATION ASSIGNMENT

In the orientation assignment phrase, according to the local gradient direction of the image, each keypoint is assigned one or more directions, and all subsequent operations transform

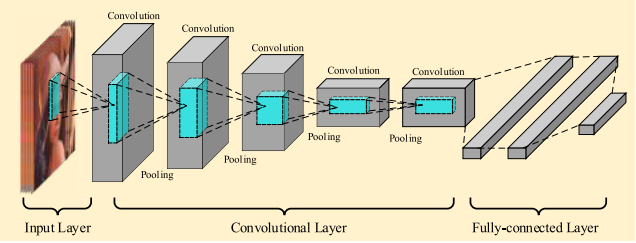


FIGURE 3. An example of convolutional neural network. A typical CNN model has three main parts: Input layer, convolutional and pooling layer, and full-connected layer.

the direction, scale and position of the keypoints to provide invariance of features to these transformations.

4) KEYPOINT DESCRIPTOR

In the last phase, the local gradients of the image are measured around each feature point at selected scales. And these gradients are transformed into a representation which allows for significant local shape distortion and illumination transformation.

C. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Network (CNN for short) was first proposed by Yann Lecun in 1998 [38]. A typical CNN shown in Fig. 3 consists of several convolutional layers, pooling layers and fully-connected layers. The convolutional layer and pooling layer cooperate to form multiple convolution groups, extract visual features from low-level to high-level layer by layer, and finally complete classification through several full-connected layers. CNN simulates feature differentiation by convolutional operation, and reduces the number of model parameters by weight-sharing and pooling. The superiority of the CNN originates from four key ideas [53]: (1) local connections, (2) shared weights, (3) pooling and (4) the use of many layers.

This powerful technique has been applied successfully in many tasks, e.g., image retrieval, visual understanding, pattern classification, etc. In this work, we employ CNN as the second method of visual representation.

IV. METHODOLOGY

In this section, we introduce a novel framework to solve the problem of SVS-JOIN. This framework support two schemes for visual words generation: the one utilizes hand-crafted visual features, namely SIFT in a conventional manner; the other is to produce visual words by generating deep visual representations via CNN, which is a better method to capture high-level semantic concepts from inputs. In addition, inspired by the algorithm of textual similarity joins, we introduce a baseline named **SVS-JOIN**$_B$ and propose a spatial grid partition based algorithm for SVS-JOIN task called **SVS-JOIN**$_G$. As an alternative approach of SVS-JOIN$_G$, a novel quadtree based global index method named **SVS-JOIN**$_Q$ is designed, which can speed up the search significantly.

A. OVERVIEW OF THE FRAMEWORK

Fig. 4 illustrates the proposed framework for SVS-JOIN problem. As discussed above, SVS-JOIN is a geo-image-oriented search problem that means the input of the system is a geo-image database. Therefore, the first priority is to generate the representations of geo-images. Two visual representation schemes are proposed in this framework: the one utilizes hand-crafted visual features that are generated via SIFT descriptor in a conventional manner, and then use BoVW model to produce visual words for each geo-image, this scheme is called **SIFT-BoVW**. The other scheme is to produce visual words by generating deep visual representations via a CNN model, which is a better method to capture high-level semantic concepts from inputs. Similar to the first scheme, we build the deep visual dictionary based on these feature representations generated by CNN and represent all the geo-images by deep visual words. We call this scheme **Deep-BoVW**. Clearly, these two visual representation schemes are all based on BoVW model, which is the basis of our geo-image index technique. In this work, two geo-image index structures are carefully designed: The first is a combination of spatial grid partition and inverted index, and the second one is a quadtree partition based inverted index. Based on these two index and similarity measurement $GeoSim(I_i^r, I_j^s)$ and $VisSim(I_i^r, I_j^s)$, we develop two efficient SVS-JOIN algorithms, namely SVS-JOIN$_G$ and SVS-JOIN$_Q$.

B. VISUAL REPRESENTATION SCHEMES

In this subsection, we introduce the two visual representation schemes in details, namely SIFT-BoVW and Deep-BoVW. Both of them are based on BoVW model. To represent a geo-image as a collection of visual words, we propose to use two different method to generate the visual word representation, namely SIFT and Deep CNN.

1) SIFT-BoVW

In this scheme, Dense-SIFT technique, an extension of SIFT is employed to extract visual features from geo-images. In other words, it maps each geo-image into a 128-dimensions feature vector. After that we utilize k-means clustering method to construct SIFT visual dictionary by converting feature vectors into visual words. Let $\{I_1, I_2, \ldots, I_n\}$ be a set of geo-images, the feature vectors of them are denoted by $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n\} = SIFT(\{I_1, I_2, \ldots, I_n\})$, where $SIFT(\cdot)$ is the SIFT feature extractor. According to the distance between these SIFT feature vectors, k-means method groups these feature vectors into k clusters $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$ which can be formulated by the following objective function:

$$\mathcal{F}_{k-\text{means}} = \underset{\mathcal{G}=\{g_i\}_{i=1}^k}{\arg\min} \sum_{i=1}^{k} \sum_{\boldsymbol{\xi}_j \in g_i} \|\boldsymbol{\xi}_j - \boldsymbol{\chi}_i\|_2^2 \tag{6}$$

where $\boldsymbol{\chi}_i$ is the mean vector of the cluster g_i, namely, $\boldsymbol{\chi}_i = \frac{1}{|g_i|} \sum_{\boldsymbol{\xi} \in g_i} \boldsymbol{\xi}$. L2 norm is used to measure the distance between mean vector and each visual feature

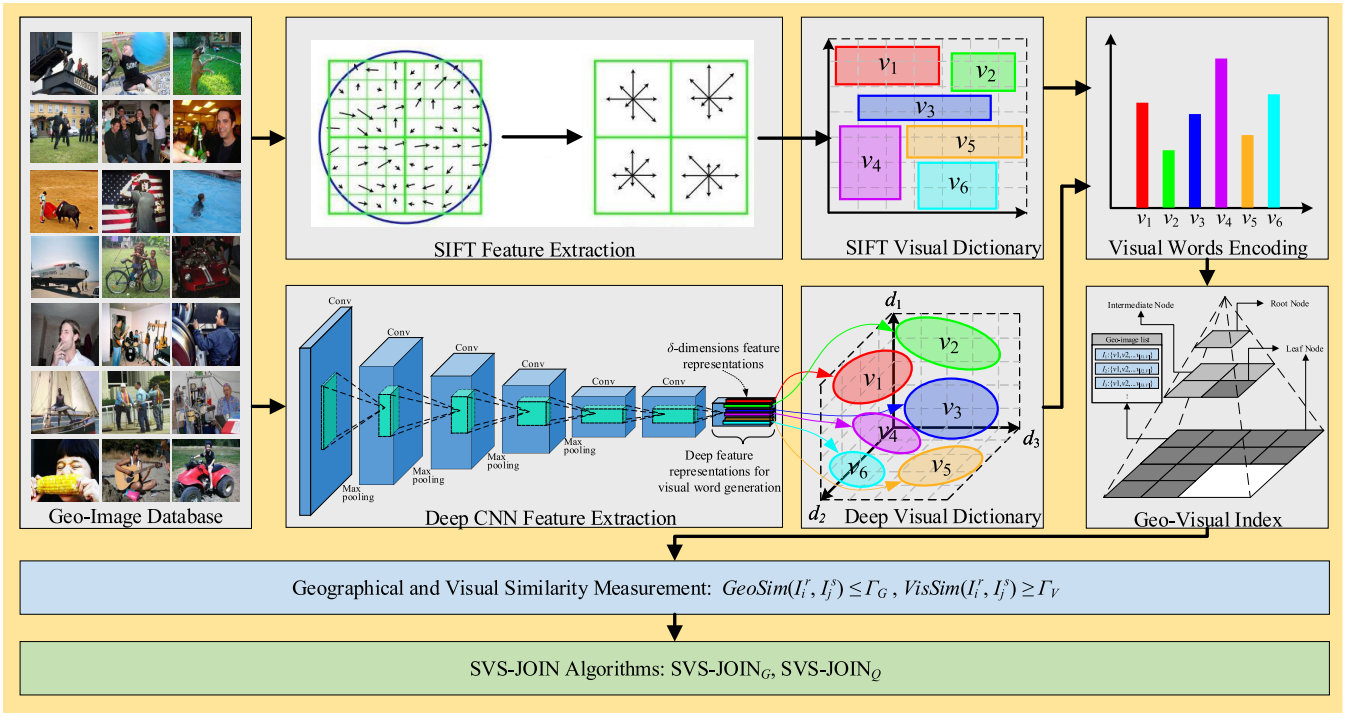


FIGURE 4. The framework to solve the SVS-JOIN problem. Best view in color. This framework supports two schemes for visual words generation: the one utilizes hand-crafted visual features, namely SIFT in a conventional manner; the other is to produce visual words by generating deep visual representations via CNN, which is a better method to capture high-level semantic concepts from inputs. Besides, based on the visual word representations and geographical information, two geo-visual index structures are integrated in this framework to organize geo-images efficiently: The first method is a combination of spatial grid partition and inverted index, and the second one is a quadtree partition based inverted index. Based on these two index techniques and geographical and visual similarity measurement $GeoSim(I_i^r, I_j^s)$ and $VisSim(I_i^r, I_j^s)$, two efficient SVS-JOIN algorithms are developed, namely SVS-JOIN$_G$ and SVS-JOIN$_Q$.

vector. After the clustering, the SIFT visual dictionary with k visual words has been constructed, namely $\{v_1, v_2, \ldots v_k\} = KMEANS(\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n\})$, where $KMEANS(\cdot)$ is the k-means algorithm. The SIFT visual dictionary is used to encode each geo-image by a k-dimensions vector that is the statistics of each visual word. As mentioned in Definition 4, we measure the weight of visual word by *tf-idf*, namely, $\{w(v_1), w(v_2), \ldots w(v_k)\} = TF\text{-}IDF(\{\eta(v_1), \eta(v_2), \ldots \eta(v_k)\})$, where $\eta(\cdot)$ denotes the number of occurrences of a visual word in an image.

2) Deep-BoVW

As the second and more powerful scheme, we propose to integrate deep CNN and BoVW model to generate the deep visual word representation named Deep-BoVW. Compared with SIFT based method, the feature extraction in a deep convolutional manner can capture the rich high-level semantic concepts, which is more powerful than conventional hand-crafted features with little semantic information. The process is quite similar to SIFT-BoVW: a deep CNN extract visual features from low-level to high-level layer-by-layer, and then a deep visual dictionary is built on these visual features by k-means algorithm, which are used to encode geo-images.

Specifically, we employ a pre-trained deep CNN model, namely AlexNet [59], for the task of visual feature extraction. AlexNet consists of five convolutional layers, some of which are followed by max-pooling layers, three fully-connected layers and a 1000-way softmax layer that is used to classification. In this work, the input images are resized as 227×227 pixels, and we use the fifth convolutional feature representations with the size $13 \times 13 \times 256$ to generate deep visual dictionary. For geo-image set $\{I_1, I_2, \ldots, I_n\}$, the deep visual representation set of them is denoted by $\{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_n\} = CONV(\{I_1, I_2, \ldots, I_n\}; \boldsymbol{\theta})$, wherein $CONV(\cdot; \boldsymbol{\theta})$ is the deep convolutional feature extractor, $\boldsymbol{\theta}$ is the network parameters, and $\forall \boldsymbol{\zeta}_i \in \{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_n\}, \boldsymbol{\zeta}_i = (\zeta_i^{(1)}, \zeta_i^{(2)}, \ldots, \zeta_i^{(256)})^T$ is a deep visual feature representation of an image. Therefore, Like SIFT-BoVW, the deep visual dictionary can be generated by k-means algorithm, namely $\{v_1, v_2, \ldots v_k\} = KMEANS(\{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_n\})$. Likewise, the visual word encoding process is denoted by $\{w(v_1), w(v_2), \ldots w(v_k)\} = TF\text{-}IDF(\{\eta(v_1), \eta(v_2), \ldots \eta(v_k)\})$.

No doubt, AlexNet is definitely not the only choice for feature extraction. And actually in the experiments we also employ two other off-the-shelf CNN models, i.e., VGGNet-16 [87] and GoogLeNet [88] to take this job for performance evaluation. VGGNet is a powerful deep convolution neural network developed by Oxford University Computer Vision group and DeepMind researchers in 2014, and GoogLeNet is another outstanding deep CNN model during that year, which utilizes a novel structure named Inception. Both of them are very powerful for computer vision tasks. To facilitate the discussion, we name these CNN based schemes as

AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW respectively during the comparative experiments.

C. THE BASELINE FOR SPATIAL VISUAL SIMILARITY JOINS

In this section, we propose the baseline for the SVS-JOIN problem. Firstly, we introduce the state-of-the-art algorithm named PPJOIN [90] for textual similarity joins, which is utilized in our baseline. Then we present our baseline named SVS-JOIN$_B$ in detail.

1) THE METHOD FOR TEXTUAL SIMILARITY JOINS

a: INVERTED INDEX BASED METHOD

The traditional way to solve textual similarity joins efficiently is to build an inverted index for the target object dataset $\mathcal{R}$, which associates each word in the global word set $\mathcal{W}$ built beforehand to an objects inverted list $\mathcal{L}$. For each object $o \in \mathcal{R}$, the inverted list $\mathcal{L}(v)$ of each word $w \in o.V$ is traversed, where $o.V$ represents the keywords set of o. Then we account the number of word overlap of o with every object $\hat{o} \in \mathcal{L}(v)$ and save these numbers in a set Ω_o. To facilitate exploration of this method, we denote $\Omega_o[\hat{o}]$ as the number of word overlap of o with $\hat{o}$. Apparently, the candidate pair set can be generated from Ω_o directly. Then for all objects $\hat{o} \in \Omega_o$, if $\Omega_o[\hat{o}] > 0$ and $TxtSim(o, \hat{o}) \geq \theta_t$, then the pair $(o, \hat{o})$ can be one of the results. To describe this process in a formal way, textual similarity joins by this method is to return a result set $\mathcal{P}$,

$$\mathcal{P} = \{(o, \hat{o}) | o \in \mathcal{R}, \hat{o} \in \Omega_o, \Omega_o[\hat{o}] > 0 \ and \ TxtSim(o, \hat{o}) \geq \theta_t\} \tag{7}$$

where $TxtSim(o, \hat{o})$ is the textual similarity function and θ_t is the textual similarity threshold.

b: PREFIX FILTERING PRINCIPLE

When we use the inverted index based method, the inverted list $\mathcal{L}_w$ will be quite long if the word w is very frequent in the dataset. This become a major challenge since a lot of candidate pairs have to be generated in this situation. To reduce the size of the candidate set, an efficient method called prefix filtering principle was devised by [89]. According to this technique, we generate a global word ordering Φ that sorts keywords by word frequency in reverse order, and then for all objects $o \in \mathcal{R}$, order the keywords in $o.V$ by Φ. After the ordering, the prefix of $o.V$ is denoted as $Pf(o.V)$ and the length of it is denoted as $|Pf(o.V)|$, which is measured by the following equation:

$$|Pf(o.V)| = |o.V| - \lceil \theta_t |o.V| \rceil + 1 \tag{8}$$

where $|o.V|$ represents the number of keywords in $o.V$, θ_t is the textual similarity threshold. It is obvious that the length of prefix of an object is determined by the number of keywords contained by this object and the similarity threshold given in advance. According to this principle, we can get the following theorem:

Algorithm 1 PPJOIN Algorithm

1: **INPUT**: $\mathcal{R}$ is an objects dataset sorted by a global ordering Φ, θ_t is a textual similarity threshold.
2: **OUTPUT**: $\mathcal{P}$ is the result pairs set.
3: **for** each $w \in \mathcal{W}$ **do**
4: $\quad \mathcal{L}_w \leftarrow \emptyset$;
5: **end for**
6: **for** each $o \in \mathcal{R}$ **do**
7: $\quad |Pf(o.V)|_S \leftarrow |o.V| - \lceil \theta_t |o.V| \rceil + 1$;
8: $\quad |Pf(o.V)|_I \leftarrow |o.V| - \lceil \frac{2\theta_t}{\theta_t + 1} |o.V| \rceil + 1$;
9: $\quad$ **for** $i = 1$ to $|Pf(o.V)|_S$ **do**
10: $\quad\quad w \leftarrow$ the i-th keyword in $o.V$;
11: $\quad\quad$ **for** $e(\hat{o}, i_{\hat{o}}) \in \mathcal{L}_w$ and $|\hat{o}.V| \geq \theta_t |o.V|$ **do**
12: $\quad\quad\quad$ **if** *QualifyPosFilter*$(o, i_o, \hat{o}, i_{\hat{o}})$ && *QualifySufFilter*$(o, i_o, \hat{o}, i_{\hat{o}})$ **then**
13: $\quad\quad\quad\quad \Omega_o[\hat{o}] \leftarrow \Omega_o[\hat{o}] + 1$;
14: $\quad\quad\quad$ **else**
15: $\quad\quad\quad\quad \Omega_o[\hat{o}] \leftarrow -\infty$;
16: $\quad\quad\quad$ **end if**
17: $\quad\quad$ **end for**
18: $\quad\quad$ **if** $i_o \leq |Pf(o.V)|_I$ **then**
19: $\quad\quad\quad \mathcal{L}_w \leftarrow \mathcal{L}_w \cup \{(o, i_o)\}$;
20: $\quad\quad$ **end if**
21: $\quad$ **end for**
22: $\quad$ *Verify*$(o, \Omega_o, \mathcal{P})$;
23: **end for**
24: **return** $\mathcal{P}$;

Theorem 1: Given two objects $o, \hat{o} \in \mathcal{R}$ and a textual similarity threshold θ_t, if $TxtSim(o, \hat{o}) \geq \theta_t$, then $Pf(o.V) \cap Pf(\hat{o}.V) \neq \emptyset$.

Obviously, the basic idea of Theorem 1 is that if the textual similarity between two objects is larger than a threshold, they should share same keywords. Therefore, this theorem can be used to prune the candidate pair set effectively. Specifically, for each object o, we just only to search the keywords contained in the prefix of o.

c: THE PPJOIN ALGORITHM

PPJOIN is one of the efficient algorithms to solve the textual similarity joins problem, developed by Xiao *et al.* [90]. This algorithm is a combination of positional filtering and prefix filtering-based algorithm.

Algorithm 1 demonstrates the pseudo-code of the PPJOIN algorithm. The input of this algorithm is a textual similarity threshold θ_t and an objects dataset $\mathcal{R}$ that is sorted in ascending order of their size. At first, in Lines 3 to 5, it generates inverted list $\mathcal{L}_w$ for each word in the global words set. Then from Line 6 to line 23, this algorithm traverses every objects in the input dataset $\mathcal{R}$ and find the similar pairs of objects. Specifically, for each object $o \in \mathcal{R}$, probe prefix length $|Pf(o.V)|_S$ and $|Pf(o.V)|_I$ are (Lines 7-8). Then from the first position to the $|Pf(o.V)|_S$-th position, it scans the prefix of o and get the word in the prefix, and generates candidate pair. After that, the filter condition $|\hat{o}.V| \geq \theta_t |o.V|$ is used to filters

the candidate pairs. The positional and suffix filter are operated by calling two procedures *QualifyPosFilter*$(o, i_o, \hat{o}, i_{\hat{o}})$ and *QualifySufFilter*$(o, i_o, \hat{o}, i_{\hat{o}})$ (Lines 11-17). The overlap will be added if the pair can qualify these filters. After that, in Lines 18-20, the inverted list $\mathcal{L}_w$ of each visual word is extended by indexing both geo-objects and geo-locations. At last, this algorithm generates the result set $\mathcal{P}$ by executing the verification procedure *Verify*$(o, \Omega_o, \mathcal{P})$ that aims to whether the actual overlap between o and the current candidates.

2) THE BASELINES FOR SVS-JOIN

In this subsection, we introduce the baseline approach. Inspiring by the prefix filtering principle and the PPJOIN algorithm, we propose a baseline called SVS-JOIN$_B$ for SVS-JOIN problem. Different from the textual similarity joins, our method consider two aspects of information, i.e., geographical information and visual information. We set two thresholds Γ_G and Γ_V to deal with the measurement of geographical similarity and visual similarity. According to the definition of SVS-JOIN, we implement two procedures *GeoSim*$(I, \hat{I})$ and *VisSim*$(I, \hat{I})$ to measure these two similarities.

a: SVS-JOIN$_B$ ALGORITHM

Algorithm 2 demonstrates the computing process in the form of pesudo-code. The input is a geo-image dataset sorted by a global ordering Φ and two thresholds Γ_G and Γ_V. At the beginning of the process, the inverted list $\mathcal{L}_w$ is initialized for each visual word. All the objects in $\mathcal{R}$ are accessed iterately from line 4 to line 21. For each object, probe and index prefix length are calculated (lines 5-6). The position filter procedure and suffix filter procedure are invoked as the same way of PPJOIN. Different from Algorithm 1, in Line 9, except the filter condition $|\hat{o}.V| \geq \Gamma_V|o.V|$, *GeoSim*$(I, \hat{I})$ is called as a geographical similarity filter to prune the geo-image pairs whose spatial distance between two images is not short enough. In Line 22, the procedure *Verify*$(I, \Omega_I, \mathcal{P})$ generates the final results set from the candidate set $\mathcal{P}$ based on Ω_I.

Although SVS-JOIN$_B$ algorithm can effectively deal with the SVS-JOIN problem, we can still improve the efficiency significantly. It is easily to know that we just consider the geo-image $\hat{I}$ that satisfies the filter condition *GeoSim*$(I, \hat{I}) \leq \Gamma_G$. Unfortunately, this is the main limitation. In other words, SVS-JOIN$_B$ algorithm considers all the geo-images $\hat{I} \in \mathcal{L}_w$ for each visual word v which is contained in $Pf(I.V)$. To overcome this challenge, in the following we present a grid based spatial partition strategy and develop a more efficient algorithm named SVS-JOIN$_G$ extending from SVS-JOIN$_B$.

b: SPATIAL GRID

We propose a grid based spatial partition strategy named spatial grid to improve the performance of algorithm. This strategy is to model the two-dimensional spatial area of dataset $\mathcal{R}$ as a grid, denoted as $G(\mathcal{R})$ that contains several cells, which equals to the geographical similarity threshold Γ_G in each dimension. Thus, the area of each cell equals Γ_G^2. Clearly, the spatial grid is determined by a spatial visual similarity join with dataset $\mathcal{R}$ and threshold Γ_G. To put it in another way, for a given dataset $\mathcal{R}$, the grid $G(\mathcal{R})$ do not need to pre-compute.

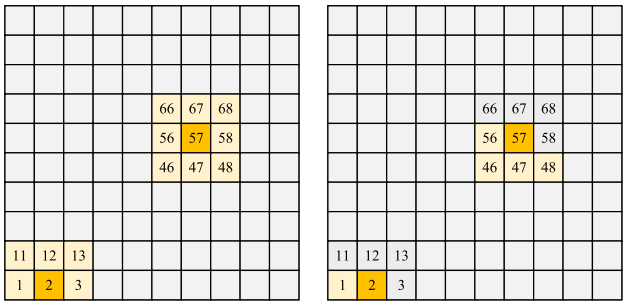
Fig. 5(a) shows how to generate candidate pairs by using spatial grid. The number in a cell is the cell id. We assume that a geo-image is located in the cell 57 colored by yellow, denoted C_{57}. To retrieve the candidate pairs $(I, \hat{I})$, just only the C_{57} and its eight neighbor cells colored by light yellow need to be accessed due to the restriction of geographical similarity threshold. Therefore, for one geo-image, we only need to check total nine cells to find its partner to form a candidate pair. If the current accessed cell is near the edge of the grid, such as C_2, only six cells should be checked for candidates searching. Thus, the search space can be reduced significantly by using this strategy. We utilize the spatial similarity filter to find the result from these cells mentioned above.

Algorithm 2 SVS-JOIN$_B$ Algorithm

1: **INPUT**: $\mathcal{R}$ is a geo-image dataset sorted by a global ordering Φ, Γ_V is a textual similarity threshold, Γ_G is a geographical similarity threshold.
2: **OUTPUT**: $\mathcal{P}$ is the result pairs set.
3: **for** each $v \in \mathcal{W}$ **do**
4: $\quad\mathcal{L}_w \leftarrow \emptyset$;
5: **end for**
6: **for** each $o \in \mathcal{R}$ **do**
7: $\quad|Pf(I.V)|_S \leftarrow |I.V| - \lceil\Gamma_V|I.V|\rceil + 1$;
8: $\quad|Pf(I.V)|_I \leftarrow |I.V| - \lceil\frac{2\Gamma_V}{\Gamma_V+1}|I.V|\rceil + 1$;
9: $\quad$**for** $i = 1$ to $|Pf(I.V)|_S$ **do**
10: $\quad\quad V \leftarrow$ the i-th visual word in $o.V$;
11: $\quad\quad$**for** $e(\hat{I}, i_{\hat{I}}) \in \mathcal{L}_w$ and *GeoSim*$(I, \hat{I}) \leq \Gamma_G$ and $|\hat{o}.V| \geq \Gamma_V|o.V|$ **do**
12: $\quad\quad\quad$**if** *QualifyPosFilter*$(I, i_I, \hat{I}, i_{\hat{I}})$ && *QualifySufFilter*$(I, i_I, \hat{I}, i_{\hat{I}})$ **then**
13: $\quad\quad\quad\quad\Omega_I[\hat{I}] \leftarrow \Omega_I[\hat{I}] + 1$;
14: $\quad\quad\quad$**else**
15: $\quad\quad\quad\quad\Omega_I[\hat{I}] \leftarrow -\infty$;
16: $\quad\quad\quad$**end if**
17: $\quad\quad$**end for**
18: $\quad\quad$**if** $i_I \leq |Pf(I.V)|_I$ **then**
19: $\quad\quad\quad\mathcal{L}_w \leftarrow \mathcal{L} \cup \{(I, i_I)\}$
20: $\quad\quad$**end if**
21: $\quad$**end for**
22: $\quad$*Verify*$(I, \Omega_I, \mathcal{P})$;
23: **end for**
24: **return** $\mathcal{P}$;

c: SVS-JOIN$_G$ ALGORITHM

Based on spatial grid method, we develop an extension of SVS-JOIN$_B$ called SVS-JOIN$_G$ algorithm. Like SVS-JOIN$_B$, this algorithm utilizes spatial grid strategy. In other words,



(a) The use of spatial grid (b) The use of spatial grid in SVS-JOINN$_G$

FIGURE 5. An example of spatial grid.

Algorithm 3 SVS-JOIN$_G$ Algorithm

1: **INPUT**: $\mathcal{R}$ is a geo-image dataset sorted by a global ordering Φ, Γ_V is the visual similarity threshold, Γ_G is a geographical similarity threshold.
2: **OUTPUT**: $\mathcal{P}$ is the result pairs set.
3: $G(\mathcal{R}) \leftarrow GridConstructor(\mathcal{R}, \Gamma_G)$;
4: **for** each $C_i \in G(\mathcal{R})$ **do**
5: $M[C_i] \leftarrow GetJoinCells(G(\mathcal{R}), C_i)$;
6: **for** each $C_j \in G(\mathcal{R})$ **do**
7: $\mathcal{P} \leftarrow \mathcal{P} \cup$ SVS-JOIN$_B(C_i, C_j, \Gamma_G, \Gamma_V)$;
8: **end for**
9: **end for**
10: **return** $\mathcal{P}$;

a spatial grid is constructed for the input dataset $\mathcal{R}$ as the basic spatial data structure. The geo-images in $\mathcal{R}$ are then accessed in the ascending order of their cell id. For each cell C_i, this algorithm will get a cells set denoted as $M[C_i]$, in which the geo-image will be joined with all of the geo-images in C_i. In $M[C_i]$, the neighbor cells of C_i have smaller id than C_i itself.

There are some differences between SVS-JOIN$_B$ and SVS-JOIN$_G$. For example, SVS-JOIN$_G$ algorithm builds an inverted index for all cells in the grid, rather than a global index. Therefore, for each visual word v in the global visual dictionary, every cell has its inverted index $C_i.\mathcal{L}_w$.

Algorithm 3 demonstrates the process of SVS-JOIN$_G$ algorithm. Similar to SVS-JOIN$_B$ algorithm, the input consists of a geo-image dataset $\mathcal{R}$ sorted by Φ, a visual similarity threshold Γ_V and a geographical similarity threshold Γ_G. The first step is to build a spatial grid $G(\mathcal{R})$ for $\mathcal{R}$, shown in Line 3. The geo-images are ordered according to cell id and $|I.V|$. After this step, it traverses the $G(\mathcal{R})$ to search the join cell by cell. For each cell C_i, the procedure $GetJoinCell(G(\mathcal{R}), C_i)$ is executed to get the cell set $M[C_i]$. For all the cells $C_j \in M[C_j]$, the algorithm executes SVS-JOIN$_B(C_i, C_j, \Gamma_G, \Gamma_V)$ to return the final results set. It is worth noting that the geo-image I located in each cell C_i are checked several times, that means more buffers need to create to store the cells for later processing.

D. THE QUADTREE BASED GLOBAL INDEX METHOD

To further improve the search efficiency, in this section we propose a novel method to solve the problem of SVS-JOIN based on a global inverted index and quadtree partition strategy.

1) QUADTREE PARTITION AND GLOBAL INDEX

a: QUADTREE PARTITION

Quadtree is one of the popular spatial indexing structures used in many applications. It aims to partition a 2-dimensional spatial region into 4 subregions in a recursive manner. Fig. 6(a) illustrates an example of quadtree that partitions the spatial region into L levels. For l-th level, the region is split into 4^l equal subregions. Each node of quadtree corresponds to a subregion. The root node of quadtree locate on the 0-th level, which represents the whole spatial region. Four subnodes in 1-level are partitioned from the root node in 0-th level. And the subnodes in 3-level are split from the nodes in 2-level as the same manner. From the Fig. 6(a) we can find that there are three colors of nodes. In specific, the light gray nodes are root node and intermediate nodes. The dark gray nodes in any level of the quadtree are the leaf nodes according to the split condition. For each leaf node, there is a list of geo-images in it. In general, the whole spatial region is partitioned into several nodes and the geo-images distribute in these nodes.

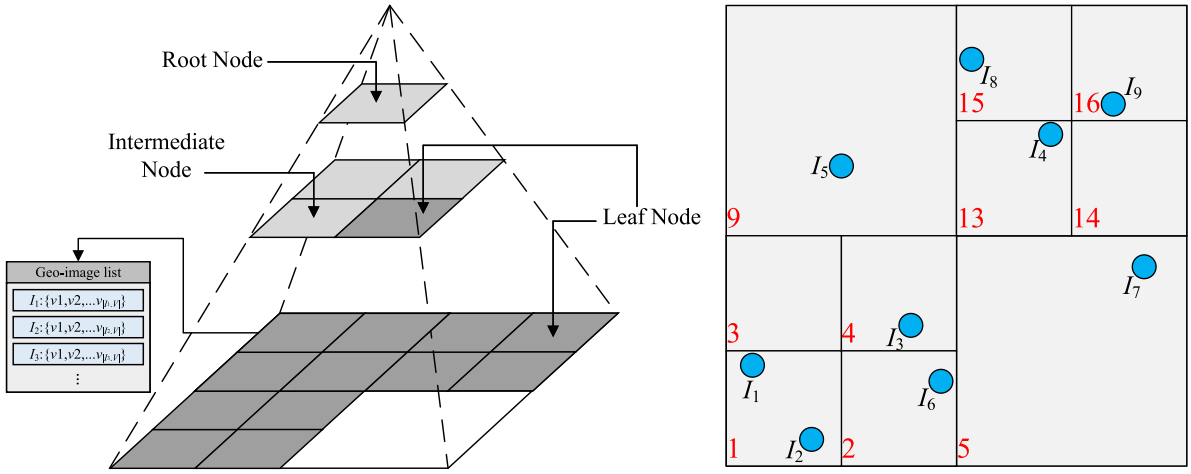
Fig. 6(b) shows the partition of the Example 2 by a quadtree. The red color number in quadtree is the node id. Apparently, these 9 geo-images are distributed in the subregions. For node 1, denoted as N_1, it contains two geo-image I_1 and I_2. As the number of geo-images in $\mathcal{R}$ is really small, the other nodes contain only one geo-image at most.

b: Z-ORDER CURVE

In this paper, we utilize Z-order curve [91] to encode each node of quadtree according to its partition sequence. As a typical space-filling curve technique, Z-order curve can map multi-dimensional data to one dimension while keeping the spatial position of data unchanged. There is a direct relationship between Z-order curve and quadtree. That is, we can utilize the Z-order to sort the data during the quadtree construction. That means the path of the node in quadtree can be represented as Z-order curve. Once sorted, the spatial data can either be stored in a binary search tree and used directly [92]. Fig. 7(a) demonstrates how to generate the Morton code of a subregion based on spatial partition sequence in a region. According to Z-order curve, we denote these 16 subregions from 0 to 15 in decimal, or from 0000 to 1111 in binary. Fig. 7(b) illustrates the Morton code in the quadtree partition of Example 2. It is obvious that the 2-dimensional spatial data are mapped to 1-dimensional space. In our solution, we use the code in binary as the node id.

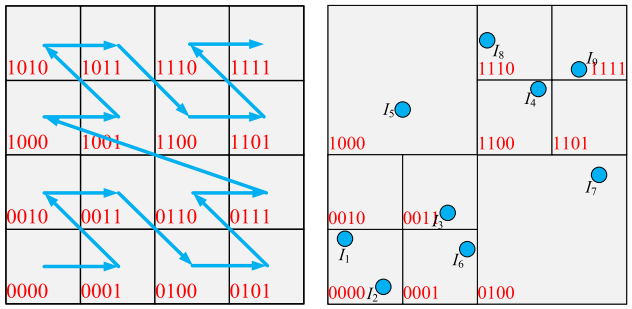
2) SVS-JOIN$_Q$ ALGORITHM

Based on the quadtree partition and the global inverted index, we develop a novel algorithm called SVS-JOIN$_Q$



(a) A quadtree (b) Quadtree partition

FIGURE 6. An example of quadtree partition.



(a) A quadtree (b) Quadtree partition

FIGURE 7. An example of Z-order.

Algorithm 4 SVS-JOIN$_Q$ Algorithm

1: **INPUT**: a geo-image dataset $\mathcal{R}$, a visual similarity threshold Γ_V, a geographical similarity threshold Γ_G.
2: **OUTPUT**: a result pairs set $\mathcal{P}$.
3: $T_{quad} \leftarrow QuadtreeConstructor(\mathcal{R}, \Gamma_G)$;
4: $\hat{\mathcal{R}} \leftarrow AscSortZ(\mathcal{R})$;
5: $\mathcal{I}_G \leftarrow GlobalIndexConstructor(\hat{\mathcal{R}}, \Gamma_V)$;
6: **for** each $I \in \hat{\mathcal{R}}$ **do**
7: $\quad \mathcal{P} \leftarrow \mathcal{P} \cup JoinSearch(I, \mathcal{I}_G, \Gamma_V, \Gamma_G)$;
8: **end for**
9: **return** $\mathcal{P}$;

to solve the spatial visual similarity joins problem efficiently. Algorithm 4 shows the pseudo-code of this algorithm. Algorithm 5 and Algorithm 6 demonstrate two key procedures applied in SVS-JOIN$_Q$. The first step of SVS-JOIN$_Q$ is to construct a quadtree by performing the procedure *QuadtreeConstructor* in Line 3 to partition the whole spatial region of the input dataset $\mathcal{R}$. In Line 4, the sorting function $AscSortZ(\mathcal{R})$ is invoked to sort the data in ascending Z-order. After that, the procedure $GlobalIndexConstructor(\hat{\mathcal{R}}, \Gamma_V)$ is invoked in Line 5 to build the global inverted index for each

Algorithm 5 GlobalIndexConstructor($\hat{\mathcal{R}}$,Γ_V)

1: **INPUT**: a geo-image dataset $\hat{\mathcal{R}}$, a visual similarity threshold Γ_V.
2: **OUTPUT**: a inverted index set $\mathcal{I}_G$.
3: Initializing: sort the visual words in descending order of number of non-zero entries;
4: Initializing: Denote the maximum of $w(I.V[i])$ for all $I \in \hat{\mathcal{R}}$ as maxweight of i-th visual word;
5: Initializing: Denote the maximum of $w(I.V[i])$ from 1 to m as maxweight of $I.V$;
6: Initializing: $\mathcal{P} \leftarrow \emptyset$;
7: Initializing: $\forall \iota_i \in \mathcal{I}_G \leftarrow \emptyset$;
8: Initializing: $S_V \leftarrow \emptyset$;
9: **for** each $I \in \hat{\mathcal{R}}$ **do**
10: $\quad \beta \leftarrow$ set of geo-image in $I.node$ or $I.neighbors$;
11: $\quad$ Denote the maximum of $I.V[i]$ for all $I \in r$ as maxweight$_i(r)$;
12: $\quad$ **for** each $I.V[i] > 0$ in ascending order of i **do**
13: $\quad\quad S_V \leftarrow S_V +$ maxweight$_i(r) * I.V[i]$;
14: $\quad\quad$ **if** $S_V > \Gamma_V$ **then**
15: $\quad\quad\quad InvertedIndexConstructor(\iota_i)$;
16: $\quad\quad$ **end if**
17: $\quad$ **end for**
18: **end for**
19: **for** each $\iota_j \in \mathcal{I}_G$ **do**
20: $\quad$ Record p_{start} and p_{end} of each node in ι_j;
21: $\quad$ Record the p_{I_i} in ι_j
22: **end for**
23: **return** $\mathcal{I}_G$;

visual word according to the visual similarity threshold Γ_V. When building the inverted index lists for geo-image I, only the geo-image $\hat{I}$ in the neighbor nodes or the same node need to be considered. Then for each inverted indexing list, the algorithm recalls the start position and end position of

Algorithm 6 JoinSearch(I,$\mathcal{I}_G$,Γ_G,Γ_V)

1: **INPUT** a geo-image I, a global index set $\mathcal{I}_G$, a visual similarity threshold Γ_V, a geographical similarity threshold Γ_G.
2: **OUTPUT** a result pairs set $\mathcal{P}$.
3: Initializing: $S_T \leftarrow \emptyset$;
4: Initializing: $\mathcal{P} \leftarrow \emptyset$;
5: Initializing: $score \leftarrow \sum_i I.V[i]*\text{maxweight}_i(\mathcal{R})$;
6: **for** each i s.t. $I.V[i] > 0$ **do**
7: **for** each node $\mathcal{N} \in I.N \cup I.neighbors$ **do**
8: $S_V \leftarrow S_V + \text{maxweight}_i(r) * I.V[i]$;
9: **if** $\mathcal{N} \in I.neighbors$ **then**
10: $p_{start} = \mathcal{N}.p_{start}$;
11: $p_{end} = \mathcal{N}.p_{end}$;
12: **else**
13: $p_{start} = GetPosition(\iota_i, I)$;
14: $p_{end} = \mathcal{N}.p_{end}$;
15: **end if**
16: **for** each $\hat{I} \in \iota_i[p_{start}, p_{end}]$ **do**
17: **if** I equals $\hat{I}$ **then**
18: Continue;
19: **end if**
20: **if** $Sim[\hat{I}] \neq 0$ || $score \leq \Gamma_G$ **then**
21: $Sim[\hat{I}] \leftarrow Sim[\hat{I}] + I.V[i] * \hat{I}.V[i]$;
22: **end if**
23: $score \leftarrow score - I.V[i]*\text{maxweight}_i(\mathcal{R})$;
24: **end for**
25: **end for**
26: **end for**
27: $Verify(I, \hat{I}, S_V, \mathcal{P})$;
28: **return** $\mathcal{P}$;

each node and the exact position of geo-images for searching. $JoinSearch(I, \mathcal{I}_G, \Gamma_G, \Gamma_V)$ in Line 7 is invoked to measure the geographical similarity and visual similarity and then retrieve all the similar geo-image pairs to generate the results $\mathcal{P}$.

V. EXPERIMENTS

In this section, we present results of a comprehensive performance evaluation on real and synthetic geo-image datasets to evaluate the accuracy, efficiency and scalability of the proposed approaches. Firstly, we introduce the details of dataset and workload in subsection V-A. Then in subsection V-B we discuss the results of experiments on two different datasets.

A. DATASET AND WORKLOAD

1) DATASETS

Performance of the proposed methods is evaluated on both real and synthetic spatial and image datasets. The following two datasets are deployed in our experiments.

- **Flickr.** Real image dataset Flickr is obtained by crawling millions image from the popular photo-sharing platform Flickr(http://www.flickr.com/). To evaluate the scalability of our proposed algorithm, The dataset size varies from 100K to 500K. The geo-location information can be obtained from the geo-tag of each image.
- **ImageNet.** Synthetic dataset ImageNet is obtained from the largest image dataset ImageNet, which is widely used in image processing and computer vision. it includes 14,197,122 images and 1.2 million images with SIFT features. We generate ImageNet dataset with varying size from 100K to 500K. The geographical information of the images are randomly generated from spatial datasets Rtree-Portal (http://www.rtreeportal.org).

Fig. 8 shows some example images of these two datasets. Some images selected from Flickr are shown in Fig. 8(a), such as the photos of outdoor sports, guitar playing, dogs, etc. The image from ImageNet dataset are shown in Fig. 8(b), which belong to many different categories, e.g., fast food, fish, dog, car, snake, flower, etc.

2) WORKLOAD

The geo-image dataset size increases from 100K to 500K; the number of the visual words contained in a geo-image grows from 20 to 100; the geographical similarity threshold Γ_G and visual similarity threshold Γ_V varies from 0.02 to 0.10 and from 0.5 to 0.9 respectively. By default, The image dataset size, the number of the visual words, the geographical similarity threshold, visual similarity threshold set to **300K**, **60**, **0.006**, **0.7** respectively. The default visual representation scheme is AlexNet-BoVW.

All the Experiments are run on a PC with Intel(R) Xeon 2.60GHz dual CPU, 16GB memory and NVIDIA GeForce GTX 1080 GPU running the Ubuntu 16.04 LTS Operation System. The visual feature extraction models (SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW) are implemented in Python and all the SVS-JOIN search algorithms (SVS-JOIN$_B$, SVS-JOIN$_G$ and SVS-JOIN$_Q$) in the experiments are implemented in Java. Note that the quadtree of SVS-JOIN$_Q$ method is maintained in memory.

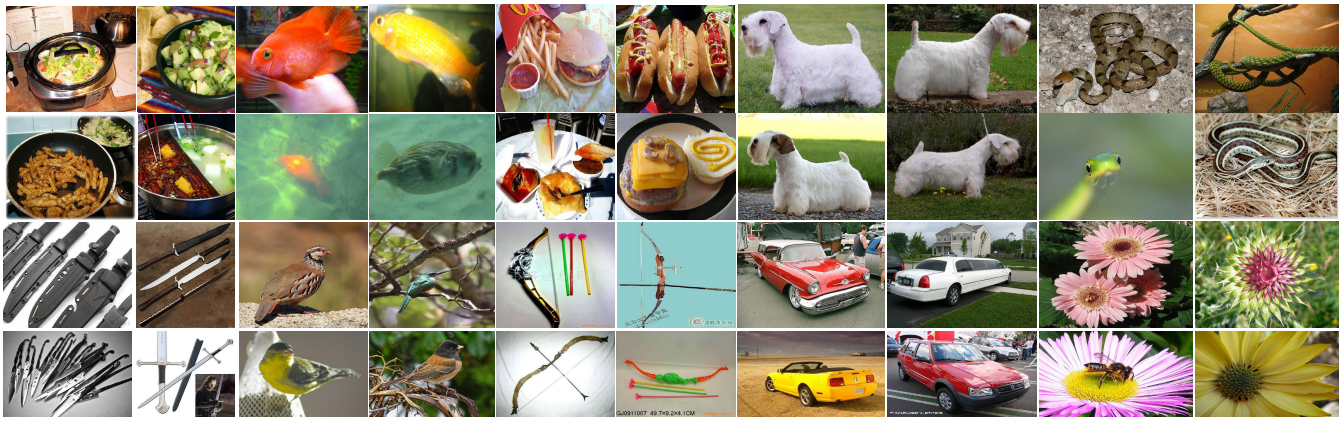
B. PERFORMANCE EVALUATION

1) COMPARISON BETWEEN VISUAL REPRESENTATION SCHEMES

We compare the search accuracy of four proposed visual representation schemes: SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW. As this is the first work to solve the SVS-JOIN problem, we just compare our methods on Flickr and ImageNet datasets.

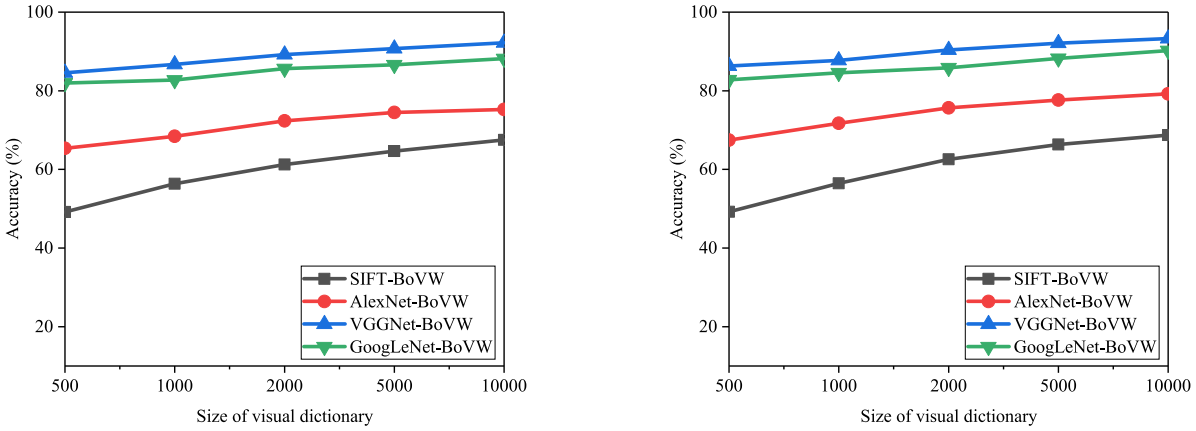
To evaluate how the dictionary size affect the search accuracy, we set the size of SIFT/deep visual dictionarys of these four schemes to 500, 1000, 2000, 5000, 10000. The conventional feature representation method, SIFT-BoVW is treated as a baseline, in which the patch size of geo-image is set to 16 × 16 pixels. For the AlexNet-BoVW method, as mentioned above we use features of the fifth convolutional layer with size of 13 × 13 × 256 to generate the visual dic-

(a) Flickr



(b) ImageNet

FIGURE 8. Some example images of experimental datasets used in our experiment.

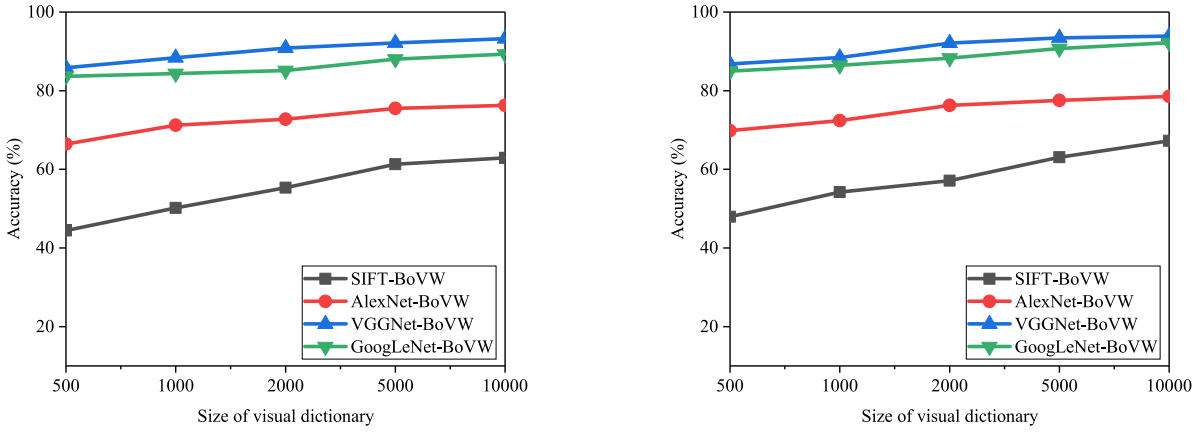


(a) The training ratio is 80%

(b) The training ratio is 90%

FIGURE 9. Accuracy of SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW on Flickr Dataset.

tionary. For other two deep feature representation schemes, VGGNet-BoVW and GoogLeNet-BoVW, we utilize features of Conv5_3 layer with size of $14 \times 14 \times 512$ and features of inception 4(e) layer with size of $14 \times 14 \times 832$ to construct dictionary respectively. Besides, we choose two training-testing settings for these evaluations, namely (1) training ratio is 80%: the dataset is split into 80% for training and 20% for testing; (2) training ratio is 90%: the dataset is split into 90%



(a) The training ratio is 80%

(b) The training ratio is 90%

FIGURE 10. Accuracy of SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW on ImageNet Dataset.

for training and 10% for testing. The details of experiments are shown as follows.

a: EVALUATION ON FLICKR DATASET

Fig. 9 illustrates the comparisons of SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW on Flickr dataset under the training ratio of 80% and 90% respectively. We can see from the Fig. 9(a) that the accuracy of all of the method creep up with the increase of the size of visual dictionary. Because the larger the visual dictionary, the more details can be represented by visual word model. This directly improves the search accuracy. As the superiority of the VGGNet-16 and GoogLeNet in the image recognition, the perfomances of VGGNet-BoVW and GoogLeNet-BoVW are higher than AlexNet-BoVW and SIFT-BoVW. Since the more semantic concepts information can be captured, these CNN-based approaches can easily combat the conventional opponent for the task of SVS-JOIN search. The accuracy of VGGNet-BoVW is a little higher than GoogLeNet-BoVW, near to 92% at the dictionary size is 10000. When the training ratio is increased to 90%, the performances of these four methods are little better than before. Because enlarging the size of training set can improve the performance of feature representation. However, what hasn't changed is the best performance of VGGNet-BoVW, which rises gradually with the growth of dictionary size. Similar to the, the accuracy of GoogLeNet-BoVW and AlexNet-BoVW ranked second and third respectively, which are obviously higher than the traditional approach SIFT-BoVW.

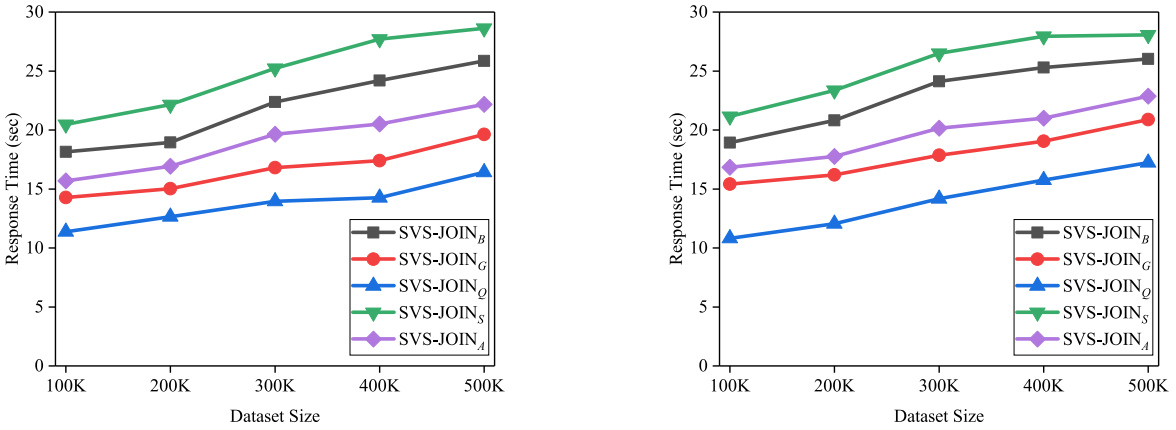
b: EVALUATION ON ImageNet DATASET

Fig. 10 demonstrates the results of the experiment between SIFT-BoVW, AlexNet-BoVW, VGGNet-BoVW and GoogLeNet-BoVW on Flickr dataset under the training ratio of 80% and 90% respectively. Under the training ratio of 80%, shown as Fig. 10 (a), all of the methods show a fluctuating growth as the size of visual dictionary grows. Like the situations on Flickr dataset, VGGNet-BoVW is superior to the opponents, which slowly rises in the internal of [500, 5000]. The performance of GoogLeNet-BoVW is the second, which is a bit lower than the former and much higher than AlexNet-BoVW and SIFT-BoVW. The Fig. 10 (b) shows beyond doubt that the deep CNN based methods are clearly defeat the traditional opponent, SIFT-BoVW, which is exactly the same as before. The accuracy of GoogLeNet-BoVW is very close to VGGNet-BoVW, and the performance of AlexNet-BoVW seems very hard to surpass them. It gradually increases to about 76% at the dictionary size is 10000.

2) COMPARISON BETWEEN DIFFERENT SVS-JOIN ALGORITHMS

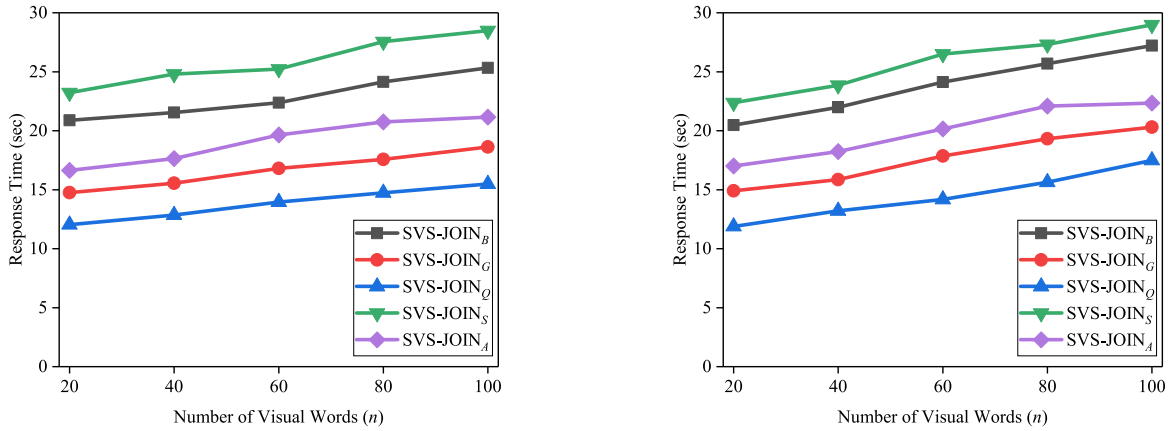
In the following we evaluate the search efficiency of SVS-JOIN algorithms on Flickr and ImageNet dataset and discuss how the dataset size, number of visual words, geographical and visual similarity threshold affect the system performance. As this work is the first time to evaluate the SVS-JOIN algorithms, we compare the performance of the following methods:

- **SVS-JOIN**$_B$. SVS-JOIN$_B$ is the technique introduced in subsection IV-C.
- **SVS-JOIN**$_G$. SVS-JOIN$_G$ is the technique introduced in subsection IV-C.
- **SVS-JOIN**$_Q$. SVS-JOIN$_Q$ is the technique introduced in subsection IV-D.
- **SVS-JOIN**$_S$. SVS-JOIN$_S$ is the technique extended from the signature-based algorithm in [93]. We modify this existing algorithm by replacing the textual Jaccard measurement with our visual similarity measurement. Besides, we use visual word representation to generate signature.
- **SVS-JOIN**$_A$. SVS-JOIN$_A$ is a combination of All-Pairs algorithm proposed in [94] and grid partition technique



(a) Evaluation on Flickr (b) Evaluation on ImageNet

FIGURE 11. Evaluation on various dataset size on Flickr and ImageNet.



(a) Evaluation on Flickr (b) Evaluation on ImageNet

FIGURE 12. Evaluation on the number of visual words on Flickr and ImageNet.

over the dataset. Likewise, we replace the textual similarity measurement with the proposed visual similarity function.

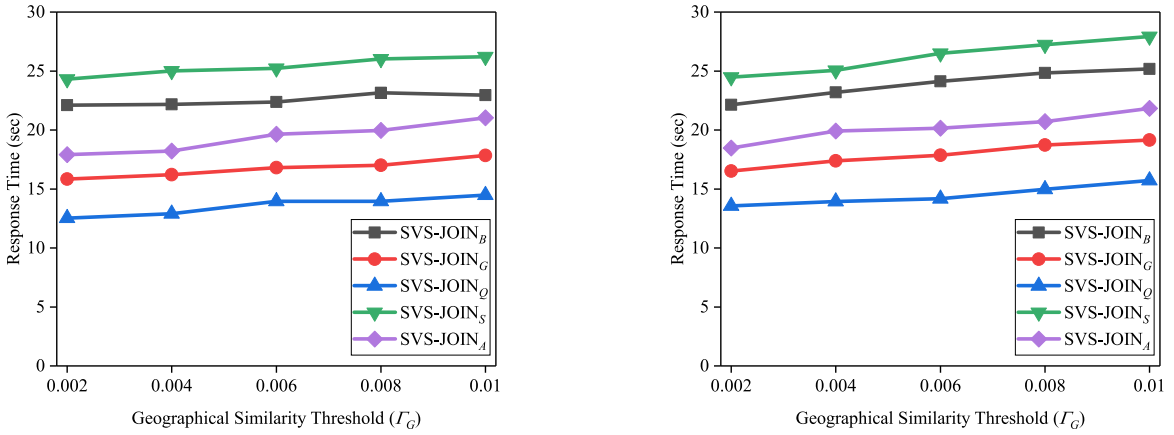
As mentioned above, the default visual representation scheme used in all these approaches is Deep-BoVW (AlexNet-BoVW) in this experiment.

a: EVALUATION ON THE SIZE OF DATASET

We evaluate the effect of the variation of dataset size on Flickr and ImageNet shown in Fig. 11. It is obvious that the response time of SVS-JOIN$_B$, SVS-JOIN$_G$, SVS-JOIN$_Q$, SVS-JOIN$_S$ and SVS-JOIN$_A$ increase gradually in Fig. 11(a). Specifically, the performance of SVS-JOIN$_S$ is the worse than SVS-JOIN$_B$ because the search algorithm used in SVS-JOIN$_B$ is more efficient than SVS-JOIN$_S$, which is nearly 30 seconds when the dataset size is enlarged to 500K. The time cost of SVS-JOIN$_G$ fluctuate from about 14 second to 23 second, which is higher than SVS-JOIN$_Q$ because the quadtree and global inverted index based solution is more efficient. However, it is more efficient than SVS-JOIN$_A$ due to the use of PPJOIN algorithm. Fig. 11(b) illustrates that the evaluation on ImageNet dataset. Similar to the situation on Flickr dataset, the performance of SVS-JOIN$_Q$ is the best due to the high efficiency of quadtree partition strategy. However, with the rising of the dataset size from 100K to 500K, the speed of increment of time cost of SVS-JOIN$_Q$ is a bit higher than the speed on Flickr, which might be due to the variety of images. On the other hand, the performance of SVS-JOIN$_B$ is still worse than SVS-JOIN$_G$ and SVS-JOIN$_Q$ since it has no better spatial index than the others. But SVS-JOIN$_B$ defeats SVS-JOIBN$_S$ again, not surprisingly.

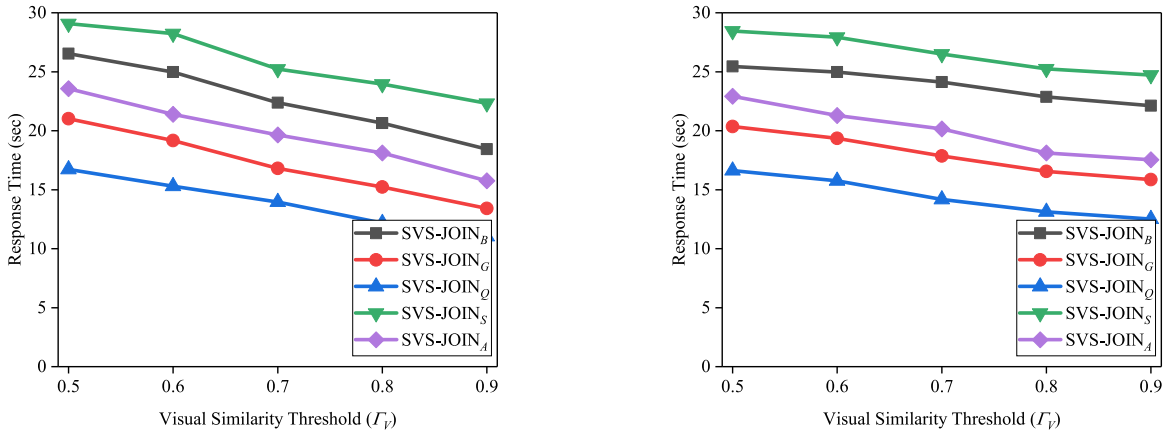
b: EVALUATION ON THE NUMBER OF VISUAL WORDS

We evaluate the effect of the number of visual words on Flickr and ImageNet dataset shown in Fig. 12. We can see from Fig. 12(a) that the response time of all these five methods



(a) Evaluation on Flickr (b) Evaluation on ImageNet

FIGURE 13. Evaluation on the geographical similarity threshold on Flickr and ImageNet.



(a) Evaluation on Flickr (b) Evaluation on ImageNet

FIGURE 14. Evaluation on the visual similarity threshold on Flickr and ImageNet.

grow step by step with the increment of number of visual words. Similar to the situation above, the lowest efficient approach is SVS-JOIN$_S$ that cannot defeat any opponent. For SVS-JOIN$_B$, when the number of visual words is larger than 40, the growth speed of it is a bit faster. Apparently, the response time of it is high, which is just lower than SVS-JOIN$_S$. SVS-JOIN$_Q$ is the most efficient algorithm among them on this dataset due to the benefit of quadtree index. As the same visual word representation utilized in these five methods, the impacts of increasing the number of visual words on them are the same, which is reflected in the similar trend. The evaluation on ImageNet dataset is shown in Fig. 12(b). Once again, without spatial partition technique and advanced search strategy, the performance of SVS-JOIN$_S$ is the worst. In the interval [60, 100], the growth speed of SVS-JOIN$_B$ and SVS-JOIN$_A$ are bit faster. However, this situation does not appear in SVS-JOIN$_G$ and SVS-JOIN$_Q$. There is no doubt the performance of SVS-JOIN$_Q$ is the best, just like the evaluations mentioned above. Thus, once again, the results confirm that the proposed quadtree partition strategy is better than the grid partition for SVS-JOIN problem.

c: EVALUATION ON THE GEOGRAPHICAL SIMILARITY THRESHOLD

We evaluate the effect of the spatial similarity threshold on Flickr and ImageNet dataset shown in Fig. 13. In Fig. 13(a), with the increasing of geographical similarity threshold, the growth rate of response time of all these five algorithms are relatively small. It is as expected that SVS-JOIN$_S$ approach has the lowest search efficiency from beginning to end. For SVS-JOIN$_B$, it shows slight fluctuations of response time, which is higher than SVS-JOIN$_A$, SVS-JOIN$_G$ and SVS-JOIN$_Q$ all along because there is no advanced spatial index technique used in it to boost the efficiency. As explained above, it just considers the filter condition $GeoSim(I, \hat{I}) \leq \Gamma_G$ during the search. For SVS-JOIN$_Q$

algorithm, the range of its fluctuation is very small, and this method has the lowest response time. This mainly benefit from our spatial search strategy, i.e., the grid-based and quadtree-based search algorithms with global index are not very sensitive to the change of geographical similarity threshold. On the other hand, the trend of SVS-JOIN$_G$ is similar to SVS-JOIN$_Q$, although its efficiency is lower than the latter. But it can defeat SVS-JOIN$_A$. For the comparisons on ImageNet dataset, we can find from Fig. 13(b) that the trends of SVS-JOIN$_S$ and SVS-JOIN$_B$ are slightly different from the situations on Flickr. The growth of the time cost seems to be a bit faster. In specific, when threshold Γ_G increases to 0.008, SVS-JOIN$_S$ increase from 24.5 to 28, and the proposed baseline SVS-JOIN$_B$ has a rise from 22 to 25. However, other two algorithms, SVS-JOIN$_G$ and SVS-JOIN$_Q$ seem to be not much affected by the increasing of Γ_G. Besides, the latter performs much better than the former, which is benefit from the usage of efficient spatial index, namely quadtree with Z-order and global inverted index.

d: EVALUATION ON THE VISUAL SIMILARITY THRESHOLD

We evaluate the effect of the visual similarity threshold on Flickr and ImageNet dataset shown in Fig. 14. We can see from the Fig. 14(a) that with the rising of visual similarity threshold, the search efficiency of these five algorithms are improved gradually. It is mainly because more geo-images are considered to be dissimilar when the threshold is very large. That means more candidates are pruned with enlarging the visual similarity threshold. Like the comparison above, the response time of SVS-JOIN$_S$ is the highest since no more efficient search strategy is used. With the grid partition technique, SVS-JOIN$_A$ and SVS-JOIN$_G$ can defeat the two that do not utilize spatial partition strategy. The efficiency of SVS-JOIN$_Q$ is higher than SVS-JOIN$_A$ and SVS-JOIN$_G$ because the employment of quadtree and global index technique can boost the spatial search obviously. In Fig. 14(b). The response time of them decline gradually but the speed of decrement is a bit slower than the speed on Flickr dataset. Same as the situation above, the SVS-JOIN$_Q$ is obviously superior to other approaches.

VI. CONCLUSION

In this paper, we study a novel geo-image retrieval paradigm named SVS-JOIN problem. Given a set of geo-images that contains geographical information and visual content information, SVS-JOIN aims to search out all the geo-image pairs from the dataset, which are similar to each other in both aspects of geographical similarity and visual similarity. We define SVS-JOIN problem in formal at first time and then propose the geographical and visual similarity function. An algorithm named SVS-JOIN$_B$ is developed, which is inspired by the approaches applied on spatial similarity joins. To improve the efficiency of search, we extend this algorithm to a novel algorithm called SVS-JOIN$_G$ that utilizes spatial grid strategy to enhance the performance of spatial retrieval. Besides, we introduce an alternative algorithm named SVS-JOIN$_Q$ that employs quadtree technique and a global inverted indexing structure, which can further speed up the search. The experimental evaluation on real and synthetic geo-multimedia datasets shows that our methods has a really outstanding performance.

REFERENCES

[1] Y. Wang, X. Lin, and Q. Zhang, "Towards metric fusion on multi-view data: A cross-view based graph random walk approach," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2013, pp. 805–810.
[2] 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.*, pp. 1–25, Oct. 2018.
[3] Y. Wang, X. Lin, L. Wu, W. Zhang, and Q. Zhang, "Exploiting correlation consensus: Towards subspace clustering for multi-modal data," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 981–984.
[4] 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, Mar. 2017.
[5] L. Wu, Y. Wang, and L. Shao, "Cycle-consistent deep generative hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1602–1612, Apr. 2018.
[6] Y. Wang, X. Lin, L. Wu, W. Zhang, and Q. Zhang, "LBMCH: Learning bridging mapping for cross-modal hashing," In *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 999–1002.
[7] C. Zhang, Y. Lin, L. Zhu, X. Yuan, J. Long, and F. Huang, "Hierarchical one permutation hashing: Efficient multimedia near duplicate detection," *Multimedia Tools Appl.*, pp. 1–24, Jun. 2018.
[8] J. Long, L. Zhu, C. Zhang, Z. Yang, Y. Lin, and R. Chen, "Efficient interactive search for geo-tagged multimedia data," *Multimedia Tools Appl.*, pp. 1–30, Aug. 2018.
[9] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Comput. Vis. Image Understand.*, vol. 167, pp. 63–73, Feb. 2018.
[10] L. Wu, Y. Wang, J. Gao, and X. Li, "Deep adaptive feature embedding with local sample distributions for person re-identification," *Pattern Recognit.*, vol. 73, pp. 275–288, Jan. 2018.
[11] 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.
[12] 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.*, Jul. 2018, pp. 1–23.
[13] 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.
[14] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4833–4843, Oct. 2018.
[15] A. Guttman, "R-trees: A dynamic index structure for spatial searching," *SIGMOD Rec.*, vol. 14, no. 2, pp. 47–57, Jun. 1984.
[16] T. K. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree: A dynamic index for multi-dimensional objects," in *Proc. 13th Int. Conf. Very Large Data Bases*, 1987, pp. 507–518.
[17] 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, Jun. 1990.
[18] 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, p. 16.
[19] 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.
[20] 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.
[21] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2011, pp. 373–384.
[22] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu, "Seal: Spatio-textual similarity search," *Proc. VLDB Endowment*, vol. 5, no. 9, pp. 824–835, 2012.

[23] 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.

[24] D. Deng, Y. Tao, and G. Li, "Overlap set similarity joins with theoretical guarantees," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 905–920.

[25] J. Ballesteros, A. Cary, and N. Rishe, "SpSJoin: Parallel spatial similarity joins," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2011, pp. 481–484.

[26] C. Efstathiades, A. Belesiotis, D. Skoutas, and D. Pfoser, "Similarity search on spatio-textual point sets," in *Proc. EDBT*, 2016, pp. 329–340.

[27] H. Hu, G. Li, Z. Bao, J. Feng, Y. Wu, Z. Gong, and Y. Xu, "Top-k spatio-textual similarity join," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 551–565, Feb. 2016.

[28] C. Rong, C. Lin, Y. N. Silva, J. Wang, W. Lu, and X. Du, "Fast and scalable distributed set similarity joins for big data analytics," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1059–1070.

[29] Z. Shang, Y. Liu, G. Li, and J. Feng, "K-Join: Knowledge-aware similarity join," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3293–3308, Dec. 2016.

[30] 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.

[31] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 81–93, Jan./Feb. 1999.

[32] N. Singhai and S. K. Shandilya, "A survey on: Content based image retrieval systems," *Int. J. Comput. Appl.*, vol. 4, no. 2, pp. 22–26, 2010.

[33] L. Wu, Y. Wang, X. Li, and J. Gao, "Deep attention-based spatially recursive networks for fine-grained visual recognition," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1791–1802, May 2019.

[34] L. Wu, Y. Wang, X. Li, and J. Gao, "What-and-where to match: Deep spatially multiplicative integration networks for person re-identification," *Pattern Recognit.*, vol. 76, no. 3, pp. 727–738, 2018.

[35] L. Wu and Y. Wang, "Robust hashing for multi-view data: Jointly learning low-rank kernelized similarity consensus and hash functions," *Image Vis. Comput.*, vol. 57, pp. 58–66, Jan. 2017.

[36] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, Sep. 1999, pp. 1150–1157.

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

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

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

[40] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR*, Jun./Jul. 2004, vol. 2, no. 4, pp. 506–513.

[41] M. Su, Y. Ma, X. Zhang, Y. Wang, and Y. Zhang, "MBR-SIFT: A mirror reflected invariant feature descriptor using a binary representation for image matching," *PLoS One*, vol. 12, no. 5, 2017, Art. no. e0178090.

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

[43] Y. Zhu, S. Cheng, V. Stanković, and L. Stanković, "Image registration using BP-SIFT," *J. Vis. Commun. Image Represent.*, vol. 24, no. 4, pp. 448–457, 2013.

[44] L. Wu, Y. Wang, J. Gao, and X. Li, "Where-and-when to look: Deep siamese attention networks for video-based person re-identification," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1412–1424, Jun. 2019.

[45] L. Wu, Y. Wang, and J. Shepherd, "Efficient image and tag co-ranking: A bregman divergence optimization method," in *Proc. 21st ACM Int. Conf. Multimedia*, Oct. 2013, pp. 593–596.

[46] 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, pp. 1470–1477.

[47] H. J. Escalante, V. Ponce-López, S. Escalera, X. Baró, A. Morales-Reyes, and J. Martínez-Carranza, "Evolving weighting schemes for the bag of visual words," *Neural Comput. Appl.*, vol. 28, no. 5, pp. 925–939, 2017.

[48] 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.

[49] R. Mandal, P. P. Roy, U. Pal, and M. Blumenstein, "Bag-of-visual-words for signature-based multi-script document retrieval," *Neural Comput. Appl.*, vol. 31, pp. 1–25, Mar. 2018.

[50] J. M. dos Santos, E. S. de Moura, A. S. da Silva, and R. da Silva Torres, "Color and texture applied to a signature-based bag of visual words method for image retrieval," *Multimedia Tools Appl.*, vol. 76, no. 15, pp. 16855–16872, 2017.

[51] F. Zhang, Y. Song, W. Cai, A. G. Hauptmann, S. Liu, S. Pujol, R. Kikinis, M. J. Fulham, D. D. Feng, and M. Chen, "Dictionary pruning with visual word significance for medical image retrieval," *Neurocomputing*, vol. 177, pp. 75–88, Feb. 2016.

[52] E. G. Karakasis, A. Amanatiadis, A. Gasteratos, and S. A. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model," *Pattern Recognit. Lett.*, vol. 55, pp. 22–27, Apr. 2015.

[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

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

[56] 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. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 157–166.

[57] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.

[58] M. Tzelepi and A. Tefas, "Deep convolutional learning for content based image retrieval," *Neurocomputing*, vol. 275, pp. 2467–2478, Jan. 2018.

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[60] 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.

[61] R. Fu, B. Li, Y. Gao, and P. Wang, "Content-based image retrieval based on CNN and SVM," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 638–642.

[62] M. Tzelepi and A. Tefas, "Exploiting supervised learning for finetuning deep CNNs in content based image retrieval," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2918–2923.

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

[64] D. Zhou, X. Li, and Y.-J. Zhang, "A novel CNN-based match kernel for image retrieval," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2445–2449.

[65] 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.

[66] 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.*, Oct. 2017, pp. 1246–1255.

[67] 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.

[68] W. Shimoda and K. Yanai, "Learning food image similarity for food image retrieval," in *Proc. IEEE 3rd Int. Conf. Multimedia Big Data (BigMM)*, Apr. 2017, pp. 165–168.

[69] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.

[70] S. Sarraf and G. Tofighi, "Classification of Alzheimer's disease using fMRI data and deep learning convolutional neural networks," 2016, *arXiv:1603.08631*. [Online]. Available: https://arxiv.org/abs/1603.08631

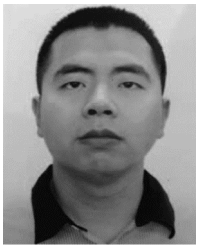
[71] F. M. Choudhury, J. S. Culpepper, Z. Bao, and T. Sellis, "Batch processing of top-*k* spatial-textual queries," *ACM Trans. Spatial Algorithms Syst.*, vol. 3, no. 4, 2018, Art. no. 13.

[72] H. K.-H. Chan, C. Long, and R. C.-W. Wong, "On generalizing collective spatial keyword queries," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1712–1726, Sep. 2018.

[73] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu, "Spatial keyword querying," in *Proc. Int. Conf. Conceptual Modeling*. Berlin, Germany: Springer, Oct. 2012, pp. 16–29.
[74] L. Chen, S. Shang, C. Yang, and J. Li, "Spatial keyword search: A survey," *GeoInformatica*, pp. 1–22, Jul. 2019.
[75] J. B. Rocha-Junior, 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.
[76] Z. Li, K. C. K. 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.
[77] D. Zhang, K.-L. Tan, and A. K. H. Tung, "Scalable top-k spatial keyword search," in *Proc. 16th Int. Conf. Extending Database Technol.*, Mar. 2013, pp. 359–370.
[78] C. Zhang, Y. Zhang, W. Zhang, and X. Lin, "Inverted linear quadtree: Efficient top k spatial keyword search," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 901–912.
[79] G. L. Li, J. Xu, and J. H. Feng, "Keyword-based k-nearest neighbor search in spatial databases," in *Proc. 21st ACM Conf. Inf. Knowl. Manage.*, Nov. 2012, pp. 2144–2148.
[80] 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.*, Jul. 2014, pp. 355–364.
[81] X. Wang, Y. Zhang, W. Zhang, X. Lin, and W. Wang, "AP-Tree: Efficiently support continuous spatial-keyword queries over stream," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 1107–1118.
[82] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa, "Keyword search in spatial databases: Towards searching by document," in *Proc. IEEE 25th Int. Conf. Data Eng.*, Mar./Apr. 2009, pp. 688–699.
[83] T. Guo, X. Cao, and G. Cong, "Efficient algorithms for answering the m-closest keywords query," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May/Jun. 2015, pp. 405–418.
[84] A. Belesiotis, D. Skoutas, C. Efstathiades, V. Kaffes, and D. Pfoser, "Spatio-textual user matching and clustering based on set similarity joins," *Int. J. Very Large Data Bases*, vol. 27, no. 3, pp. 297–320, Jun. 2018.
[85] M. Perdacher, C. Plant, and C. Böhm, "Cache-oblivious high-performance similarity join," in *Proc. Int. Conf. Manage. Data*, Jun./Jul. 2019, pp. 87–104.
[86] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[88] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[89] S. Chaudhuri, V. Ganti, and R. Kaushik, "A primitive operator for similarity joins in data cleaning," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Apr. 2006, p. 5.
[90] C. Xiao, W. Wang, X. Lin, J. X. Yu, and G. Wang, "Efficient similarity joins for near-duplicate detection," *ACM Trans. Database Syst.*, vol. 36, no. 3, Aug. 2011, Art. no. 15.
[91] G. M. Morton, *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*. Ottawa, ON, Canada, IBM Ltd, 1966.
[92] [Online]. Available: https://en.wikipedia.org/wiki/Z-order_curve
[93] S. Liu, G. Li, and J. Feng, "Star-join: Spatio-textual similarity join," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, Oct./Nov. 2012, pp. 2194–2198.
[94] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 131–140.

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

WEIREN YU received the Ph.D. degree from the School of Computer Science and Engineering, University of New South Wales. He is currently an Assistant Professor of Computer Science with the University of Warwick, and also an Honorary Fellow with Imperial College. Before joining Warwick, he was a postdoctoral position with Imperial College. He has published more than 30 articles in DB and IR. His research interests include web search and information retrieval, graph data management, and streams data mining. He received three Best Paper Awards, two CiSRA Best Paper Awards, a One of the Best Papers of ICDE 2013, and the Best Student Paper Award. He has served on various editorial boards, and as a PC and an active Reviewer for journals, such as the IEEE TKDE, *The VLDB Journal*, IEEE TIFS, ACM TKDD, WWWJ, *Sensors* and conferences, such as SIGIR, SIGMOD, VLDB, ICDE, EDBT, 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 and query processing on spatial data and multimedia data.

ZUPING ZHANG received the B.S. degree from the Foundation of Mathematics, 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, Changsha, China, in 2005. He is currently a Professor with the School of Information Science and Engineering, Central South University. His current research interests include information fusion and information systems, big data technology and application, parameter computing, and biology computing.

FANG HUANG was born in Changsha, China. She received the Ph.D. degree in traffic information engineering and control from Central South University, China, in 2007. She is currently a Professor with the School of Information Science and Engineering, Central South University. Her main research interests include social network mining and analysis, data mining and knowledge discovery, and big data analysis.

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 include image retrieval, machine learning, computer vision, and crowdsourcing learning.