Multiview Inherent Graph Hashing for Large-Scale Remote Sensing Image Retrieval

Yinghui Sun, Wei Wu, Xiaobo Shen, *Member, IEEE*, and Zhen Cui, *Member, IEEE*

***Abstract*—Remote sensing image retrieval (RSIR) is one of the most challenging tasks in remote sensing (RS) community. With the volume of RS images increases explosively, conventional exhaustive search is often infeasible in real applications. Recently, hashing has attracted increasing attention for RSIR due to significant advantage in terms of computation and storage. Hashing first generates a set of short compact hash codes to encode RS images, and then applies hash codes for effective RSIR. Multiview hashing usually achieves promising RSIR performance by fusing multiples kinds of RS image features. Conventional multiview hashing simply predefines graph Laplacian in each view, which cannot effectively explore underlying similarity structures among RS images. To address this issue, this article proposes a novel multiview inherent graph hashing (MvIGH) for RSIR. MvIGH captures the latent similarities among RS images, and adaptively learns weights of each view to characterize its contribution. In addition, MvIGH further minimizes the quantization errors. We develop an efficient alternating algorithm to solve the formulated optimization problem. The experiments on three public RS image datasets demonstrate the superiority of the proposed method over the existing multiview hashing methods in RSIR tasks.**

***Index Terms*—Hash learning, large-scale remote sensing (RS) image retrieval, multiview remote sensing data.**

I. Introduction

With the rapid development of earth observation (EO), the volume of remote sensing (RS) data increases dramatically [1]–[3]. As a vital application of RS, remote sensing image retrieval (RSIR) [4] has become an open and tough task in the RS community. The goal of RSIR is to find a list of images from the RS dataset that are most similar to the given query image. In early works, this goal was always achieved by exhaustively comparing the query image with each image in the dataset. The search complexity of the algorithm is $O(N)$, and the storage complexity is $O(Nd)$, where N is the number of images in the dataset, and d is the dimensionality of an image feature. In the era of RS Big Data, the number N of images and the dimensions d of feature descriptors of remote sensing images are very high. Exhaustively comparing the high-dimensional feature descriptor of an inquiry image with each image is often prohibitively in computation, both in time and space, which makes it infeasible for practical applications. To address the aforementioned problems, the approximate nearest neighbor (ANN) search which trades the retrieval accuracy for time cost is introduced to RSIR. Among the diverse ANN methods, hashing is a successful solution for many applications due to the significant advantages in terms of computation and storage. Hashing focuses on mapping the images from the original feature space to the Hamming space [5], [6], where the binary codes with similarity preservation are used for the effective similarity search.

The data-independent hashing methods are first studied. They simply generate the hash functions by random projections, which map similar data into similar binary codes with high probabilities. The representative methods in this category include the most well-known locality sensitive hashing (LSH) [7] and its extensions [8]. These methods require relatively long binary codes to achieve good performances. The data-dependent methods [9]–[19], which are also called learning to hash methods, have recently been proposed to address this problem. They apply machine learning techniques to learn the hash functions from the training set to generate more compact binary codes. Compared to the data-independent methods, the data-dependent hashing methods often have comparable or even better methods with shorter binary codes. The representative data-dependent hashing methods include spectral hashing (SH) [9], PCA hashing (PCAH) [11], anchor graph hashing (AGH) [10], iterative quantization (ITQ) [12], discrete graph hashing [13]. To name a few, spectral hashing (SH) [9] is one of the earliest data-dependent hashing methods. SH utilizes the distribution of data and turns to be the eigen-decomposition problem of graph Laplacian matrix. Anchor graph hashing (AGH) [10] is then proposed to adopt anchor graph for hashing learning. However, they work around the single-view image retrieval task. Generally speaking, they cannot be directly handle RS image described as different features.

In reality, remote sensing images represented by multiple kinds of features are continually increasing. For example, each image can be described by different kinds of features, such as sift feature, gist feature, rgb feature. We refer this kind of data as multiview RS data. Each view reflects some specific characteristics of the RS images. Compared to single-view RS images, multiview RS images offers more comprehensive

Manuscript received July 22, 2021; revised September 15, 2021 and October 7, 2021; accepted October 12, 2021. Date of publication October 19, 2021; date of current version November 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 62176126 and 61906091, in part by the Natural Science Foundation of Jiangsu Province, China (Youth Fund Project) under Grant BK20190440, and in part by the Fundamental Research Funds for the Central Universities under Grant 30921011210. *(Corresponding authors: Xiaobo Shen; Zhen Cui.)*

The authors are with the School of Computer and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: sunyh1023@163.com; wwu@njust.edu.cn; njust.shenxiaobo@gmail.com,; zhen.cui@njust.edu.cn,).

Digital Object Identifier 10.1109/JSTARS.2021.3121142

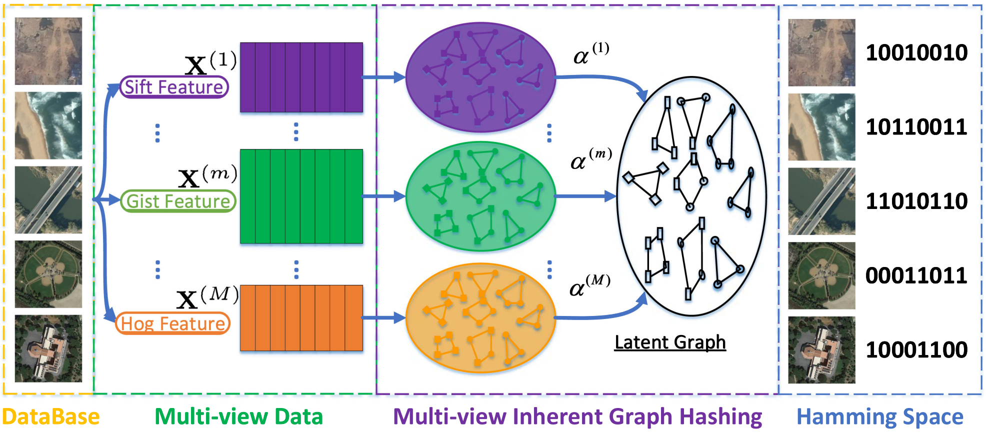


Fig. 1. Overview of the proposed MvIGH.

information. Existing literature [20] shows that fusing multiple views usually leads to better performance than the single view. Consequently, there emerge more and more demands on multiview RSIR. Although there are only a few works [21], [22] for multiview RSIR, some efforts [23]–[29] have recently been made towards effective hash code learning from multiview data in machine learning community. Composite hashing with multiple information sources (CHMIS) [23] establishes a graph for each view, and then combines them to learn the linear hash function. Multiple feature hashing (MFH) [24] preserves the local structure of each view and globally considers the alignments of all views to learn a group of hash functions. Later on, multiple feature kernel hashing (MFKH) [25] learns hash function by preserving certain similarities with linearly combined multiple kernels corresponding to different features. [26] proposed multiview spectral hashing (SU-MVSH). SU-MVSH computes the α-averaged similarity matrix from all views, and adopts the sequential learning approach to obtain the hash function. However, there are two main drawbacks in these methods. First, they simply use the predefined Gaussian kernel to calculate the similarity between the kNN neighbors in each view. This approach is not robust to noises, and fails to capture the latent similarity structure among the multiple views. Second, they simply discard the discrete constraint to reduce to a relaxed problem. It will suffer from large distortion errors, and thus the obtained binary codes are nonoptimal [13], [30]. Thus, how to generate high-quality hash codes that capture the inherent similarity structure from multiview data is still a challenging research topic.

Due to the effectiveness of hashing-based methods, in this article, we introduced a new multiview hashing method to address the above challenges for RSIR task. This method adaptively learns the weights among the nearest neighbors to fully capture the latent similarity structure among the multiple views. The overview of the proposed method is illustrated in Fig. 1. We summarize the main contributions of this work as follows.

1) We propose a novel multiview inherent graph hashing, i.e., MvIGH, to learn high-quality hash codes for multiview RS images. MvIGH jointly learns the hash codes, hash functions and adaptive similarity weights within one framework to fully preserve the latent similarity structure of multiview RS data. In addition, MvIGH also considers minimizing quantization loss by a new regularizer.
2) We develop an efficient iterative optimization algorithm to solve the proposed MvIGH, where each subproblem can be solved efficiently. Moreover, we theoretically show that the convergence of the optimization algorithm is strictly guaranteed.
3) We perform extensive experiments on three published RS image datasets. The results demonstrate that the proposed MvIGH outperforms the state-of-the-art multiview hashing methods in RSIR.

The rest of this article is organized as follows. Some published literature related to RSIR and hash learning are briefly introduced in Section II. The details of proposed adaptive multigraph hashing are presented in Section III. The experimental evaluations are given in Section IV. Finally, Section V concludes this article.

II. Related Work

A. Remote Sensing Image Retrieval

Existing RSIR methods can be roughly divided into two categories. The first category focuses on extracting effective features of RS images. Datcu *et al.* [31] proposed a knowledge-driven information mining (KIM) system. KIM system first extracts the primitive features from RS images, and reduces the dimension of features by an unsupervised classification. Then Bayesian networks are used to learn the user-specific interests from the semantic level. Yang *et al.* [32] adopted the bag-of-words (BOW) representations for RSIR and some design parameters on BOW features are discussed. Deep convolutional neural networks (DCNN) is introduced to RSIR due to its powerful capacity of feature representation. Tang *et al.* [33] presented a novel unsupervised deep feature learning method which is called deep bag of words (DBOW). DBOW first learns the discriminative descriptor for the RS image using a deep convolutional auto-encoder (DCAE) model in the image patch level, and then the DBOW features are constructed based on the patch features. The other category aims at improving the retrieval scheme. The entropy-balanced statistical (EBS) k-d tree [34] is proposed to develop an efficient indexing mechanism for large-scale RS databases, and the EBS k-d tree is built through top-down decision tree induction. Two-stage reranking (TSR) [35] is proposed to improve the performance of RSIR by using reranking method. First, TSR finds the k-nearest neighbors of a query RS image. Second, an editing scheme is developed to eliminate the negative influence of the dissimilar images. Finally, a multisimilarity fusion reranking (MSFR) method is proposed to rerank the rest RS images. A content-based SAR image retrieval method is proposed in [36]. First, the initial retrieval results is obtained by the region-based fuzzy matching (RFM) measure. Then, a multiple relevance feedback (MRF) scheme is presented to improve the retrieval results.

TABLE I

IMPORTANT NOTATIONS USED IN THIS ARTICLE

Notation	description
$\mathbf{X}^{(m)}$	data matrix of the m-th view
$\mathbf{W}^{(m)}$	projection matrix of the m-th view
$\mathbf{B}$	latent hash code matrix
$\mathbf{S}$	similarity matrix
$\mathbf{I}_r$	identity matrix of size r
$\alpha^{(m)}$	weight of the m-th view
d_m	dimensionality of the m-th view
M	the number of views
N	the number of RS images
k	the number of data sample of anchor graph
c	the number of anchors
r	the length of hash code
r_1	the weight redistribution parameter
r_2	the similarity redistribution parameter

B. Hashing for RSIR

Recently, hashing methods have been introduced to RSIR task [21], [22], [37], [38], [39]. In very early works, data-independent methods are studied. LSH [7] and its extensions are representative works. LSH generates hash functions by random projections without using original data, thus, the length of hash codes should be very long to guarantee the performances. To learn more compact hash codes, many data-dependent hashing methods have been proposed. Data-dependent methods can be roughly divided into unsupervised ones and supervised ones. Unsupervised hashing methods generate hash codes without semantic information. For example, spectral hashing (SH) [9] explores the neighborhood structure inherent in the data and generates balanced and uncorrelated hash codes. Motivated by SH, anchor graphs hashing (AGH) [10] replaces the adjacency graph of RS data by an approximate neighborhood graph by using anchor graphs. Quantization error is an important factor that influences performances of learned hash codes. There are some works focusing on reducing the quantization error. Iterative quantization (ITQ) [12] rotates zero-centered PCA-projected data to minimize the quantization error of mapping. Discrete graph hashing (DGH) [13] is another work dealing with the discrete constraints. Supervised hashing methods exploit semantic similarity between the images to learn more discriminative hash codes. Semi-supervised hashing (SSH) [11] minimize the error on the labeled data while maximizing variance and independence of hash bits over the labeled and unlabeled data. Kernel-based supervised hashing (KSH) [40] minimizes the Hamming distance between similar pairs and maximizes that between dissimilar pairs. The kernel-based hash functions are adopted to handle linearly inseparable data. Demir and Bruzzone [41] adopt the kernel-based nonlinear hashing method for large-scale RSIR task. Extensive experiments provided the effectiveness of hashing methods for large-scale RSIR. The foregoing methods are designed for RS images represented by the single feature. Due to the complex image categories and various texture structures of RS images, the single feature descriptor is often difficult to completely characterize the content of an image. However, there are only a few attempts [21], [22] on multiview hashing for RSIR. Source-invariant deep hashing convolutional neural networks (SIDHCNNs) [21] is the first work exploits the method for multiview RSIR. It contains a series of optimization constraints and is optimized in an end-to-end manner. [22] propose a deep cross-modality hashing network (DCMHN). DCMHN first transforms RGB images into four types of single-channel images, then randomly selects one type of them to construct image pairs with each corresponding SAR or optical images to solve the cross-modality discrepancy caused by imaging mechanisms. Finally, hash function is learned for efficient retrieval.

III. MULTIVIEW INHERENT GRAPH HASHING

A. Problem Statement

In this article, matrices and vectors are written in boldface. For a matrix $\mathbf{X} = (x_{ij})$, its ith row, jth column are denoted by $\mathbf{X}_{i\cdot}$ and $\mathbf{X}_{\cdot j}$ respectively. The ℓ_2-norm of a vector is defined as $\|\cdot\|_2$. The matrix Frobenius norm is denoted by $\|\cdot\|_F$. $\mathbf{X}^\top$ denotes the transpose of $\mathbf{X}$. $\mathrm{Tr}(\cdot)$ denotes the trace of a square matrix.

Suppose that $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^N$ is a set of RS images, and we are given its corresponding features $\{\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, \ldots, \mathbf{x}_N^{(m)}]^\top \in \mathbb{R}^{N\times d_m}\}_{m=1}^M$, where d_m is the dimension of the mth view, M is the number of views, N is the number of images. We also denote the latent hash code matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]^\top \in \{-1,1\}^{N\times r}$ [1], where $\mathbf{b}_i \in \{-1,1\}^{r\times 1}$ is the hash code associated with $\mathbf{o}_i$, and r is the code length. The aim of MvIGH is to learn $\mathbf{B}$ that can well preserve the latent similarity structure between objects with high probability. The important notations in this article are summarized in Table I.

B. Formulation

In MvIGH, similarity preservation is designed to maintain the neighborhood relationship among the samples in each view after being mapping into the Hamming space. It defines the similarity

[1] We use "-1" bit and "$+1$" bit during training model, and in fact we use "0" bit and "$+1$" bit for storing hash code.

preserving function for mth view

$$\begin{aligned}\min_{\mathbf{b}_i^{(m)}} \quad & \sum_{i,j=1}^{N} S_{ij}^{(m)} \left\| \mathbf{b}_i^{(m)} - \mathbf{b}_j^{(m)} \right\|_2^2 \qquad (1)\\ \text{s.t.} \quad & \mathbf{b}_i^{(m)} \in \{-1,+1\}^r\end{aligned}$$

where $S_{ij}^{(m)}$ denotes the similarity of $\mathbf{x}_i^{(m)}$ and $\mathbf{x}_j^{(m)}$ in the mth view. Equation (1) can be rewritten in a compact matrix form

$$\begin{aligned}\min_{\mathbf{b}^{(m)}} \quad & \operatorname{Tr}\left(\mathbf{B}^{(m)\top}\mathbf{L}^{(m)}\mathbf{B}^{(m)}\right) \qquad (2)\\ \text{s.t.} \quad & \mathbf{B}^{(m)} \in \{-1,+1\}^{N\times r}, \mathbf{B}^{(m)\top}\mathbf{B}^{(m)} = N\mathbf{I}_r\end{aligned}$$

where $\mathbf{L}^{(m)} \in \mathbb{R}^{N\times N}$ is the normalized graph Laplcaian matrix computed by the data local structure using different strategies. The first constraint is a discrete constraint for $\mathbf{B}^{(m)}$, and the second constraint is to require the bits to be uncorrelated. Summing up all the M view, we have the following objective function:

$$\begin{aligned}\min_{\mathbf{B}^{(m)},\boldsymbol{\alpha}} \quad & \sum_{m=1}^{M} \alpha^{(m)} \operatorname{Tr}\left(\mathbf{B}^{(m)\top}\mathbf{L}^{(m)}\mathbf{B}^{(m)}\right) \qquad (3)\\ \text{s.t.} \quad & \mathbf{B}^{(m)} \in \{-1,+1\}^{N\times r}, \mathbf{B}^{(m)\top}\mathbf{B}^{(m)} = N\mathbf{I}_r\\ \text{and} \quad & \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0, m = 1,\ldots,M\end{aligned}$$

where $\alpha^{(m)}$ is a nonnegative variable for weighting the relative importance of the mth view in the learning process. However, the approach has two main drawbacks. 1) Similarity matrices $\mathbf{S}^{(m)}$ in different views are not consistent. Thus, this strategy is not enough to characterize the common structure among different views since it characterizes each view independently. Besides, existing researches [42], [43] have shown that $\mathbf{S}^{(m)}$ is not robust using KNN method that is sensitive to the noise and number of neighbors. 2) conventional similarity matrix is with $N \times N$ size, thus, it is not scalable for large-scale application.

To overcome the above mentioned problems, we first assume that there exists a latent similarity matrix $\mathbf{S}$ to characterize all the views. We then adopt anchor graph [44] for graph construction due to its computation efficiency. Anchor graph uses a small set of samples called anchors to approximate the neighborhood structure. We apply scalable k-means clustering to obtain the anchors. We first generate $c(c \ll N)$ anchor points $\{\boldsymbol{\mu}_j^{(m)}\}_{j=1}^{c}$ by applying scalable k-means clustering [45] on the mth view. Simultaneously, the similarity matrix $\mathbf{S} \in \mathbb{R}^{N\times c}$ characterizes the similarities between the samples and anchors, where S_{ij} denotes the similarity between the ith object and the jth anchor. The formulation with adaptive similarity in the mth view can be rewritten

$$\min_{\mathbf{S},\mathbf{b}_i^{(m)}} \quad \sum_{i=1}^{N}\sum_{j=1}^{c} S_{ij} \left\| \mathbf{b}_i^{(m)} - \mathbf{b}_{\mu_j}^{(m)} \right\|_2^2 \qquad (4)$$

where $\mathbf{b}_{\mu_j}^{(m)}$ is the binary code of $\boldsymbol{\mu}_j^{(m)}$. Summing up all the M views, we have the following objective function:

$$\begin{aligned}\min_{\mathbf{S},\boldsymbol{\alpha},\mathbf{B}^{(m)}} \quad & \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{c} \alpha^{(m)} \left\| \mathbf{b}_i^{(m)} - \mathbf{b}_{\mu_j}^{(m)} \right\|_2^2 S_{ij} \qquad (5)\\ \text{s.t.} \quad & \mathbf{B}^{(m)} \in \{-1,+1\}^{N\times r}\\ \text{and} \quad & \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0, m = 1,\ldots,M.\end{aligned}$$

In addition, for simplicity, we assume that there exists a linear mapping between the mth view, i.e., $\mathbf{X}^{(m)}$ and Hamming space, i.e., $\mathbf{B}$, where the linear mapping is parameterized with a transformation matrix, i.e., $\mathbf{W}^{(m)}$. To this end, (5) can be further transformed as

$$\begin{aligned}\min_{\mathbf{S},\boldsymbol{\alpha},\mathbf{W}^{(m)}} \quad & \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{c} \alpha^{(m)} \left\| \boldsymbol{\Gamma} \right\|_2^2 S_{ij} \qquad (6)\\ \text{s.t.} \quad & \mathbf{W}^{(m)\top}\mathbf{W}^{(m)} = \mathbf{I}_r\\ \text{and} \quad & \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0, m = 1,\ldots,M\end{aligned}$$

where $\boldsymbol{\Gamma} = \mathbf{x}_i^{(m)\top}\mathbf{W}^{(m)} - \boldsymbol{\mu}_j^{(m)\top}\mathbf{W}^{(m)}$, and the orthogonal constraint is imposed on $\mathbf{W}^{(m)}$ to preserve the metric structure of the ith view of data. In this work, we learn $\mathbf{S}$ from multiview data. Each row of $\mathbf{S}$ corresponds to neighborhood anchors and characterizes the local structure of each data point. The sparse constraint on $\mathbf{S}$ is defined as

$$\|\mathbf{S}_{i.}\|_0 = k \qquad (7)$$

where k is the predefined number of neighborhoods. $\|\cdot\|_0$ denotes the ℓ_0 norm of a vector, i.e., the number of nonzero element of a vector. With the above constraint, (6) can be further transformed into

$$\begin{aligned}\min_{\mathbf{S},\boldsymbol{\alpha},\mathbf{W}^{(m)}} \quad & \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{c} \left(\alpha^{(m)}\right)^{r_1} \left\| \boldsymbol{\Gamma} \right\|_2^2 \left(S_{ij}\right)^{r_2} \qquad (8)\\ \text{s.t.} \quad & \mathbf{W}^{(m)\top}\mathbf{W}^{(m)} = \mathbf{I}_r,\\ \text{and} \quad & \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0, m = 1,\ldots,M\\ \text{and} \quad & \sum_{j=1}^{c} S_{ij} = 1, \|\mathbf{S}_{i.}\|_0 = k, S_{ij} \geq 0, i = 1,\ldots,N\end{aligned}$$

where $r_1, r_2 > 1$ are two parameters that are used to avoid trivial solution of (8). Specifically, r_1 is used to avoid trivial solution of $\boldsymbol{\alpha}$ where $\alpha^{(m)}$ with smallest loss equals to 1 and other entries equal to 0. This solution only selects one view and ignores the other views. In addition, r_2 is used to avoid trivial solution with $\mathbf{S} = \mathbf{0}$.

Due to the difficulty of directly learning discrete binary codes, one conventional way is to bypass the discrete optimization problem by some certain relaxation strategy, which however separates the binary code learning into two mutually independent stages, i.e., learning continuous representations, transforming into binary codes via some binarization methods. Typically, such optimization scheme ignores the correlation of the above two stages, which may severely limit the representative power of the generated binary codes. In this work, we consider the following objective function to characterize the quantization loss:

$$\begin{aligned}&\min_{\mathbf{B},\mathbf{W}^{(m)}} \left\|\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}-\mathbf{B}\right\|_F^2 \\ &\quad \text{s.t.}\quad \mathbf{B}\in\{-1,+1\}^{N\times r}, \mathbf{W}^{(m)\top}\mathbf{W}^{(m)}=\mathbf{I}_r\end{aligned} \tag{9}$$

where $\mathbf{B}$ is the latent hash code matrix. The minimization of the above loss function ensures that the continuous embedding should be close to the target binary codes. By summarizing the adaptive similarity structure preservation and quantization loss minimization of binary codes, we finally formulate the final objective function as follows:

$$\begin{aligned}&\min_{\mathbf{S},\mathbf{W}^{(m)},\boldsymbol{\alpha},\mathbf{B}} \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\alpha^{(m)}\right)^{r_1}\|\boldsymbol{\Gamma}\|_2^2\,(S_{ij})^{r_2} \\ &\qquad + \lambda\left\|\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}-\mathbf{B}\right\|_F^2 \\ &\quad \text{s.t. } \mathbf{B}\in\{-1,+1\}^{N\times r}, \mathbf{W}^{(m)\top}\mathbf{W}^{(m)}=\mathbf{I}_r \\ &\quad \text{and } \sum_{j=1}^{c}S_{ij}=1, \|\mathbf{S}_{i\cdot}\|_0=k, S_{ij}\geq 0, i=1,\ldots,N \\ &\quad \text{and } \sum_{m=1}^{M}\alpha^{(m)}=1, \alpha^{(m)}\geq 0, m=1,\ldots,M\end{aligned} \tag{10}$$

where λ is a nonnegative tradeoff parameter, weighting the relative importance of the similarity preservation term and the quantization loss term.

C. Optimization

It is difficult to solve the above problem directly since 1) the above objective function is not convex over all variables simultaneously; and 2) the constraints are discrete. Therefore, we apply an iterative algorithm to optimize the variables.

1) Update $\mathbf{W}^{(m)}$: We iteratively update $\mathbf{W}^{(m)}$ in the mth view. By dropping some irrelevant terms to $\mathbf{W}^{(m)}$, we have

$$\begin{aligned}&\min_{\mathbf{W}^{(m)}} \sum_{i=1}^{N}\sum_{j=1}^{c}\left\|\mathbf{x}_i^{(m)}\mathbf{W}^{(m)}-\boldsymbol{\mu}_j^{(m)}\mathbf{W}^{(m)}\right\|_2^2(S_{ij})^{r_2} \\ &\qquad + \lambda\left\|\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}-\mathbf{B}\right\|_F^2 \\ &\quad \text{s.t.}\quad \mathbf{W}^{(m)\top}\mathbf{W}^{(m)}=\mathbf{I}_r.\end{aligned} \tag{11}$$

Algorithm 1: Curvilinear Search Algorithm Based on Cayley Transformation.

```
Input: initial point $\mathbf{W}^{(m)} \in \mathcal{M}_{d_m}^{r}$ [2], matrix $\mathbf{P}^{(m)}$, $\mathbf{K}$, $\mathbf{B}$.
Output: $\mathbf{W}^{(m)}$.
Initialize $k = 0$, $\epsilon > 0$, and $0 < \rho_1 < \rho_2 < 1$.
repeat
    Compute the gradient $\mathbf{G}^{(m)}$ according to (12);
    Generate the skew-symmetric matrix
        $\mathbf{A}^{(m)} = \mathbf{G}^{(m)\top}\mathbf{W}^{(m)} - \mathbf{W}^{(m)\top}\mathbf{G}^{(m)}$;
    Compute the step size $\tau_k$, that satisfies the
        Armijo–Wolfe conditions [46] via the line search
        along the path $\mathbf{H}_k(\tau)$ defined by (13);
    Set $\mathbf{W}^{(m)} = \mathbf{H}(\tau_k)$;
    Set $k = k + 1$;
until convergence
```

We denote $\mathbf{G}^{(m)} = \nabla\mathcal{F}(\mathbf{W}^{(m)})$ as the gradient with respect to $\mathbf{W}^{(m)}$ in (11), which can be computed as follows:

$$\begin{aligned}\mathbf{G}^{(m)} &= 2\mathbf{P}^{(m)\top}\mathbf{K}\mathbf{P}^{(m)}\mathbf{W}^{(m)} \\ &\quad + 2\lambda\mathbf{X}^{(m)\top}\left(\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}-\mathbf{B}\right)\end{aligned} \tag{12}$$

where $\mathbf{P}^{(m)} = [\mathbf{X}^{(m)\top}, \boldsymbol{\mu}^{(m)\top}]^\top$, $\mathbf{K} = [\mathbf{D}_r, -\mathbf{S}; -\mathbf{S}^\top, \mathbf{D}_c]$, $\mathbf{D}_r$ and $\mathbf{D}_c$ are two diagonal matrices with diagonal entry that equals to the sum of each row and each column, respectively. We then further define the skew-symmetric matrix $\mathbf{A}^{(m)} = \mathbf{G}^{(m)\top}\mathbf{W}^{(m)} - \mathbf{W}^{(m)\top}\mathbf{G}^{(m)}$. The new trial point is determined through Crank–Nicolson-like scheme

$$\mathbf{H}(\tau) = \mathbf{W}^{(m)} - \frac{\tau}{2}\mathbf{A}^{(m)\top}\left(\mathbf{W}^{(m)} + \mathbf{H}(\tau)\right) \tag{13}$$

where τ is the step size. $\mathbf{H}(\tau)$ is given in the following closed form:

$$\mathbf{H}(\tau) = \mathbf{W}^{(m)}\mathbf{Q} \tag{14}$$

where $\mathbf{Q} = (\mathbf{I}_r - \frac{\tau}{2}\mathbf{A}^{(m)\top})(\mathbf{I}_r + \frac{\tau}{2}\mathbf{A}^{(m)\top})^{-1}$. (14) is referred as the Cayley transformation. Similar to linear search along a straight line, curvilinear search can be applied to find a proper step size for τ and guarantee the iterations to converge to a stationary point. The details of the curvilinear search algorithm for this subproblem are shown in Algorithm 1.

2)Update $\alpha^{(m)}$: With $\mathbf{W}^{(m)}, \mathbf{B}, \mathbf{S}$ fixed, we optimize the weight of each view $\alpha^{(m)}$. By ignoring some irrelevant terms with respect to α, we have

$$\begin{aligned}&\min_{\alpha^{(m)}} \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\alpha^{(m)}\right)^{r_1}\|\boldsymbol{\Gamma}\|_2^2\,(S_{ij})^{r_2} \\ &\quad \text{s.t.}\quad \sum_{m=1}^{M}\alpha^{(m)}=1, \alpha^{(m)}\geq 0, m=1,\ldots,M\end{aligned} \tag{15}$$

[2] $\mathcal{M}_{d_m}^{r}$ represents a feasible set, which is defined as $\mathcal{M}_{d_m}^{r} = \{\mathbf{W} \in \mathbb{R}^{d_m\times r} | \mathbf{W}^\top\mathbf{W} = \mathbf{I}_r\}$.

Algorithm 2: Multiview Inherent Graph Hashing.

```
Input: Training set $\{\mathbf{X}^{(m)} \in \mathbb{R}^{N\times d_m}\}_{m=1}^{M}$; code length $r$;
  the number of neighborhoods $k$; parameters $r_1, r_2, \lambda$.
Output: projection matrices $\{\mathbf{W}^{(k)}\}_{k=1}^{M}$.
Initialize $\alpha^{(m)} = 1/M$;
Initialize $\mathbf{B}$ as $\{-1,1\}^{N\times r}$ randomly;
Initialize $\mathbf{S}$ by computed the similarity between $\mathbf{X}^{(1)}$
  and $\boldsymbol{\mu}^{(1)}$;
repeat
    Update $\mathbf{W}^{(m)}$ using (14);
    Update $\alpha^{(m)}$ via (20);
    Update $\mathbf{S}$ via (24);
    Update $\mathbf{B}$ using (26);
until convergence
```

For simplicity, by denoting $l_m = \sum_{i=1}^{N}\sum_{j=1}^{c}\|\boldsymbol{\Gamma}\|_2^2(S_{ij})^{r_2}$, (15) becomes

$$\min_{\alpha^{(m)}} \quad \sum_{m=1}^{M} l_m \left(\alpha^{(m)}\right)^{r_1} \tag{16}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0, m = 1, \ldots, M.$$

The Lagrange function with respect to $\alpha^{(m)}$ is

$$\mathcal{L}(\alpha^{(m)}) = \sum_{m=1}^{M} l_m \left(\alpha^{(m)}\right)^{r_1} - \gamma \left(\sum_{m=1}^{M} \alpha^{(m)} - 1\right) \tag{17}$$

where γ is the Lagrange multiplicator. By setting the derivation with respect to $\alpha^{(m)}$ to zero, we have

$$r_1 \left(\alpha^{(m)}\right)^{r_1 - 1} l_m - \gamma = 0. \tag{18}$$

Then we further have

$$\alpha^{(m)} = \left(\frac{\gamma}{r_1 l_m}\right)^{\frac{1}{r_1 - 1}}. \tag{19}$$

With the constraint $\sum_{i=1}^{M}\alpha^{(m)} = 1$, we get

$$\alpha^{(m)} = \frac{\alpha^{(m)}}{\sum_{i=1}^{M}\alpha^{(m)}} = \frac{(l_m)^{\frac{1}{1-r_1}}}{\sum_{i=1}^{M}(l_m)^{\frac{1}{1-r_1}}}. \tag{20}$$

3) Update $\mathbf{S}$: By denoting $g_{ij} = \sum_{m=1}^{M}(\alpha^{(m)})^{r_1}\|\boldsymbol{\Gamma}\|_2^2$, the optimization problem with respect to $\mathbf{S}$ in (10) can be reduced to

$$\min_{\mathbf{S}} \quad \sum_{i=1}^{N}\sum_{j=1}^{c} g_{ij}\left(S_{ij}\right)^{r_2} \tag{21}$$

$$\text{s.t.} \quad \sum_{j=1}^{c} S_{ij} = 1, \|\mathbf{S}_{i.}\|_0 = k, i = 1, \ldots, N.$$

We update $\mathbf{S}$ row by row respectively. The objective function with respect to the ith row in (21) becomes

$$\min_{\mathbf{S}_{i.}} \quad \sum_{j=1}^{c} g_{ij}\left(S_{ij}\right)^{r_2} \tag{22}$$

$$\text{s.t.} \quad \sum_{j=1}^{c} S_{ij} = 1, \|\mathbf{S}_{i.}\|_0 = k, S_{ij} \geq 0, i = 1, \ldots, N.$$

Without the constraint $\|\mathbf{S}_{i.}\|_0 = k$, the optimal solution to the problem in (22) is as follows:

$$S_{ij} = \frac{(g_{ij})^{\frac{1}{1-r_2}}}{\sum_{l=1}^{c}(g_{il})^{\frac{1}{1-r_2}}}. \tag{23}$$

Since $\|\mathbf{S}_{i.}\|_0 = k$, only k entries in $\mathbf{S}_{i.}$ are nonzero, which means we only need to optimize the k nonzeros elements. The larger g_{ij} leads to larger objective function value in (22). Thus, we only need to update the k elements of $\mathbf{S}_{i.}$ that corresponds to the k smallest g_{ij}, and set the rest $N - k$ elements to zero. Assume that the smallest k elements of g_{ij} are $g_{ij_1}, g_{ij_2}, \ldots, g_{ij_k}$, the optimal $\mathbf{S}_i.$ with the sparse constraint in (21) is as follows:

$$S_{i,j_l} = \begin{cases} \frac{(g_{ij_l})^{\frac{1}{1-r_2}}}{\sum_{q=1}^{k}(g_{i,j_q})^{\frac{1}{1-r_2}}}, & \text{for} \quad l = 1, 2, \ldots, k \\ 0, & \text{Otherwise.} \end{cases} \tag{24}$$

4) Update $\mathbf{B}$: The objective function in (10) with respect to $\mathbf{B}$ can be reduced to

$$\min_{\mathbf{B}} \quad \left\|\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)} - \mathbf{B}\right\|_F^2 \tag{25}$$

$$\text{s.t.} \quad \mathbf{B} \in \{-1, +1\}^{N\times r}.$$

Since $\mathbf{W}^{(m)}$ is fixed, (25) can be further reduced to the following problem:

$$\min_{\mathbf{B}} \quad -2\left(\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}\right)^{\top}\mathbf{B} \tag{26}$$

$$\text{s.t.} \quad \mathbf{B} \in \{-1, +1\}^{N\times r}.$$

Obviously, we can easily have the solution of $\mathbf{B}$ as follows:

$$\mathbf{B} = \text{sign}\left(\sum_{m=1}^{M}\mathbf{X}^{(m)}\mathbf{W}^{(m)}\right) \tag{27}$$

where sign$(\cdot)$ is the sign function.

We next have the following convergence theorem of MvIGH.

Theorem 1: The alternate updating rules in Algorithm 2 monotonically decrease the objective function value of MvIGH, i.e., (10) in each iteration, and Algorithm 2 will converge to a local minimum of MvIGH.

Proof: The proof is easy to obtain. Since each subproblem is solved exactly, the value of objective function is monotonically decreasing. Besides, the objective function is lower bounded by zero. Thus Algorithm 2 is guaranteed to converge to a local minimum of MvIGH. ■

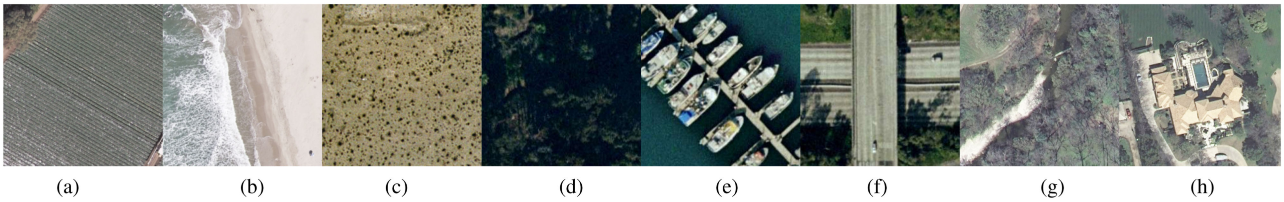


Fig. 2. Some examples of the UCM dataset.

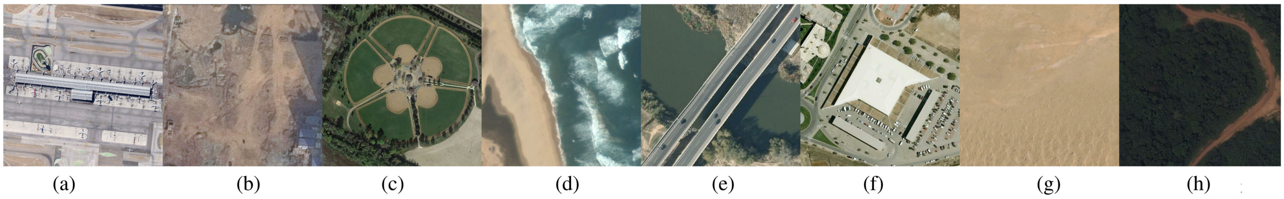


Fig. 3. Some examples of the AID dataset.

TABLE II
STATISTICS OF THREE DATASETS

Datasets	# Dataset	# Training	# Database	# Query	# Dim
UCM	2,100	1,680	1,680	420	512/500
AID	10,000	8,000	8,000	2,000	384/300
NWPU	31,500	25,200	25,200	6,300	228/150

IV. EXPERIMENTS

In this section, we evaluate the performance of the proposed MvIGH for RSIR tasks, and compare it with several state-of-the-art hashing methods.

A. Dataset

Three published RS image datasets are selected for evaluation. To construct multiview data, we adopt two visual features to represent RS images: GIST features and SIFT-based BOW features. The statistics of the three datasets are summarized in Table II. The detailed descriptions of three datasets are as follows.

1) The first one is proposed by the University of California Merced [47], and we name it UCM. There are 2100 RS images divided into 21 semantic categories in UCM. Each category includes 100 images with 256 × 256 pixels. In addition, the spatial resolution is 0.3 m per pixel in the RGB color space. Some examples are showed in Fig. 2. We extract the 512-D Gist vector and 500-D vector quantized from dense Sift features, respectively.
2) The second dataset is introduced by Wuhan University and HuaZhong University of Science and Technology [48], and we refer to it as AID in this article. AID consists 10 000 RS images with fixed size of 600 × 600 which are divided into 30 semantic categories. The volume of different scenes are ranged from 200 to 420, and the spatial resolution changes from 0.5 to 8 m per pixel. Some examples of different scenes are exhibited in Fig. 3. We extract the 384-D Gist vector and 300-D vector quantized from dense Sift features, respectively.
3) The last dataset is constructed by Northwestern Polytechnical University [49], and we name it NWPU for short. NWPU is made up of 45 scene categories, and each category includes 700 images with a size of 256 × 256 in the RGB space. The spatial resolution varies from about 0.2 to 30 m per pixel. Some examples are displayed in Fig. 4. We extract the 228-D Gist vector and 150-D vector quantized from dense Sift features, respectively.

For each dataset, we randomly sample 20% samples as the query set, and the rest samples as the training set and database.

B. Experimental Setting

To prove the effectiveness of the proposed MvIGH, we compare it with one single view hashing method SH [9] and five multiview hashing methods, including CHMIS [23], MFH [24], MFKH [25], SU-MVSH [26], and CMFH. The source codes of the comparison methods are kindly provided by the authors. In MvIGH, the parameter λ ranges from $[10^{-3}, 10^{-1}, 10^{-0}, 10^{1}, 10^{3}]$, r_1 and r_2 range from [2, 4, 6, 8, 10], and the number of neighbors k ranges from $[3, 4, 8, 10, 20]$. These parameters are finally chosen by cross-validation on the training set. The parameters of the comparison methods are carefully tuned according to the corresponding literatures, and their best performances are reported.

In this article, large-scale remote sensing image retrieval performance is quantitatively evaluated using the following two widely adopted metrics: the mean average precision (MAP) and the precision-recall curve. mAP is the mean of all the queries' average precision (AP) in the database. For a query q, AP is defined as

$$AP(q) = \frac{1}{L_q} \sum_{s=1}^{R} P_q(s)\delta_q(s) \tag{28}$$

where R is the number of retrieved list, L_q is the number of the ground truth neighbors in the list, $P_q(s)$ is the precision of the top s retrieved results, $\delta_q(s) = 1$ if the sth result is the true neighbor and 0 otherwise. In our experiment, we set R as the number of images in the database. A precision-recall (PR) curve is a graph with precision values on the *y*-axis and recall values on the *x*-axis.

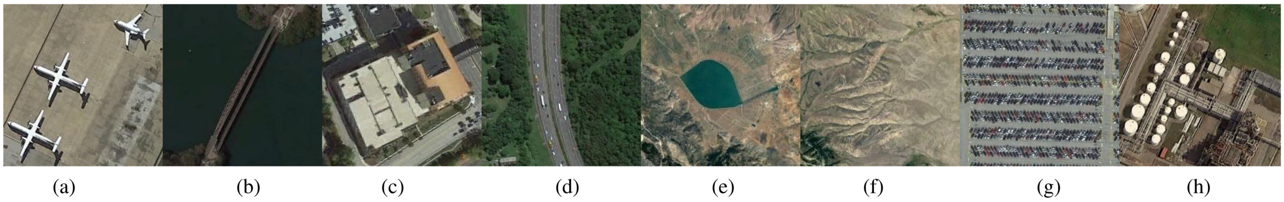


Fig. 4. Some examples of the NWPU dataset.

TABLE III
MAP Comparison With Respect to Different Number of Bits on UCM Dataset

Method	16	32	64	128
SH	0.1076	0.1417	0.1621	0.1596
CHMIS	0.1612	0.1677	0.1499	0.1262
MFH	0.1476	0.1427	0.1361	0.1251
MFKH	0.1510	0.1618	0.1748	0.1482
SU-MVSH	0.1623	0.1901	0.2013	0.1711
CMFH	0.1472	0.1639	0.1761	0.1881
MvIGH	**0.1832**	**0.1954**	**0.2110**	**0.2117**

Bold numbers denote best performances.

TABLE IV
MAP Comparison With Respect to Different Number of Bits on AID Dataset

Method	16	32	64	128
SH	0.0572	0.0556	0.0562	0.0660
CHMIS	0.0832	0.0771	0.0662	0.0588
MFH	0.0665	0.0637	0.0633	0.0589
MFKH	0.0801	0.0879	0.0847	0.0793
SU-MVSH	0.0716	0.0893	0.1008	0.0959
CMFH	0.0721	0.0810	0.0937	0.1004
MvIGH	**0.0867**	**0.0957**	**0.1017**	**0.1080**

Bold numbers denote best performances.

TABLE V
MAP Comparison With Respect to Different Number of Bits on NWPU Dataset

Method	16	32	64	128
SH	0.0307	0.0309	0.0330	0.0373
CHMIS	0.0444	0.0431	0.0378	0.0331
MFH	0.0399	0.0399	0.0369	0.0330
MFKH	0.0449	0.0497	0.0484	0.0453
SU-MVSH	0.0412	0.0496	0.0515	0.0401
CMFH	0.0406	0.0428	0.0485	0.0513
MvIGH	**0.0459**	**0.0531**	**0.0543**	**0.0571**

Bold numbers denote best performances.

TABLE VI
MAP Comparison With Respect to Different Number of Anchors on UCM Dataset

Strategy	10	20	50	100	200
Random	0.2046	0.2069	0.2027	0.2094	0.2055
k-means	0.2078	**0.2113**	0.2077	0.2088	0.2017

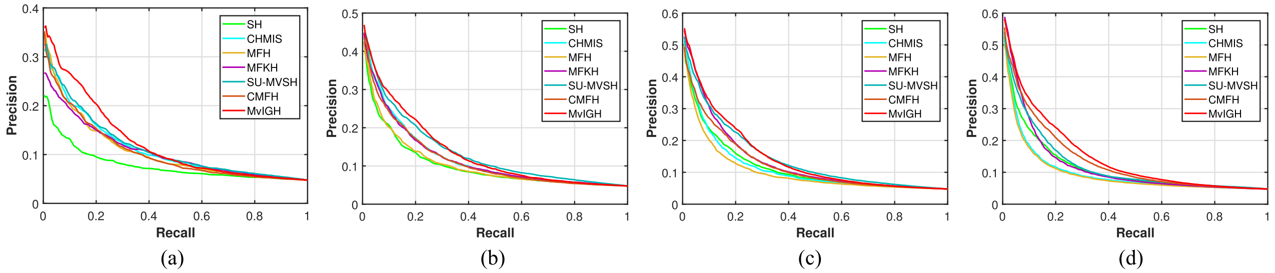
C. Performance Evaluation

This section evaluates the performance of MvIGH by comparing it with six state-of-the-art methods on the three datasets. The mAP results of all the methods on UCM, AID, NWPU are reported in Tables III–V, respectively. In addition, the PR curves with different code lengths, i.e., 16, 32, 64, 128 b are also shown in Fig. 5–7. We empirically compare k-means and random strategies in the proposed MvIGH and vary c from [10, 20, 50, 100, 200] to evaluate the number of anchors on MvIGH. The performances of the proposed MvIGH with respect to the two strategies on UCM dataset with 64-b code are shown in Table VI. The observations can be found from these results.

1) The proposed MvIGH outperforms all the comparisons on all the RS datasets. MvIGH has the best mAP results in all the 12 cases from Tables III–V. In Figs. 5–7, we see that the PR curves of MvIGH are above the others. These results obviously validate the superiority of MvIGH over the comparison methods on the large-scale visual retrieval tasks.
2) From Tables III–V, we can see that multiview methods outperform single-view method in most cases. The performance of the proposed MvIGH increases with the length of hash codes increasing, and that of some comparison methods decreases with the length of hash codes increasing. For example, the performance of SU-MVSH increases first when code length varies from [16,32,64], then decreases when code length is 128. One possible reason lies in low variances of the latter bits in these methods, which reduces the quality of the entire hash codes.
3) From Table VI, we can see k-means and random strategies achieve similar performance, and the reason is that similarity graph is optimized in the proposed MvIGH, thus select of anchors has limited impact on performance. However, the anchors generated by k-means is enough to cover the whole dataset when c is close to the number of categories, while random strategy may need to generate more anchors. As large c leads to high computational complexity, it is suggested to apply k-means than random strategy in MvIGH.

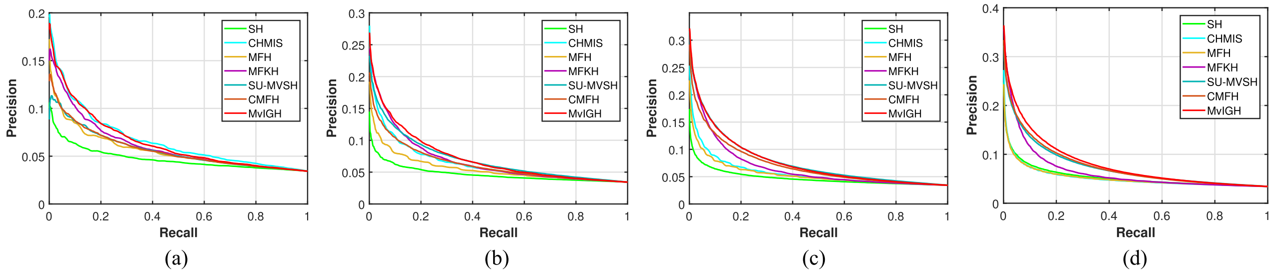
D. Convergence Analysis

In this section, we empirically analyze the convergence of MvIGH. We adopt the AID dataset for this experiment, and Fig. 8 shows the convergence curves of MvIGH. From Fig. 8, we can clearly see that MvIGH converges very quickly.



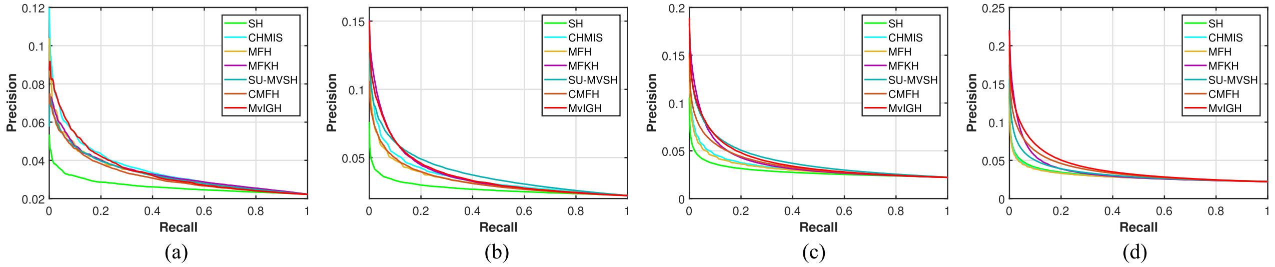
(a) (b) (c) (d)

Fig. 5. Precision-recall curves on UCM dataset with respect to different number of bits. (a) 16 b. (b) 32 b. (c) 64 b. (d) 128 b.



(a) (b) (c) (d)

Fig. 6. Precision-recall curves on AID dataset with respect to different number of bits. (a) 16 b. (b) 32 b. (c) 64 b. (d) 128 b.



(a) (b) (c) (d)

Fig. 7. Precision-recall curves on NWPU dataset with respect to different number of bits. (a) 16 b. (b) 32 b. (c) 64 b. (d) 128 b.

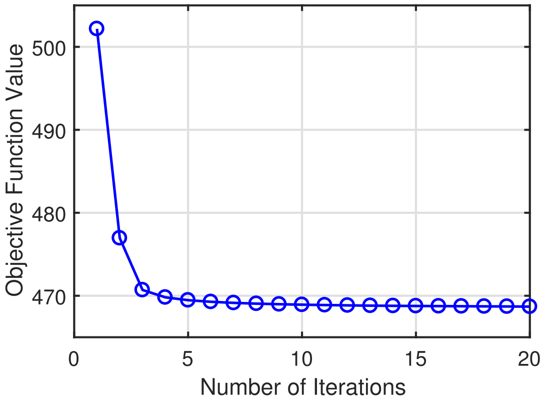


Fig. 8. Convergence analysis of the proposed MvIGH on the AID dataset.

E. Time Complexity Analysis

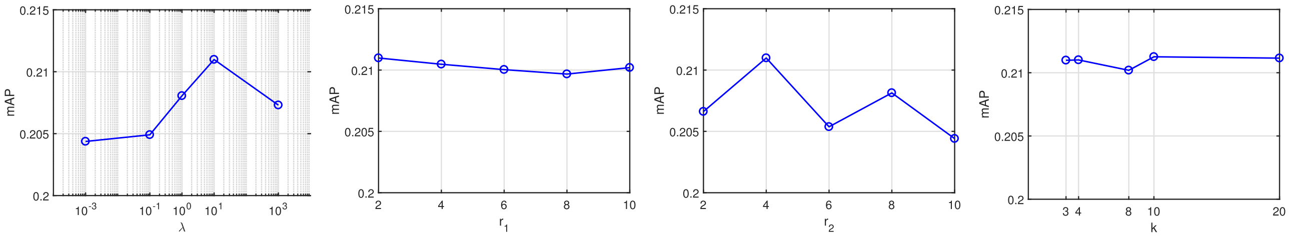
This section evaluates computational complexity of the proposed method, we compare training time of the proposed method and the comparisons on NWPU dataset with 64-b code and 20 anchors. Table VII presents training time of all the methods. As can be seen from Table VII, MvIGH is faster than CHMIS and MFH, and slower than other multiview hashing methods. The training time of MvIGH is mostly spent on curvilinear search step. The curvilinear search step enforces orthogonal property on projection matrix, which helps MvIGH improve its performance.

F. Influence of Different Parameters

The influence of different parameters on MvIGH model is discussed in this section. λ is ranged from $[10^{-3}, 10^{-1}, 1, 10, 10^{3}]$. r_1 and r_2 are varied from $[2, 4, 6, 8, 10]$. In addition, We study the sensitivity of the number of nearest neighbors, i.e., k in MvIGH. We vary k from the range of $[3, 4, 8, 10, 20]$. Fig. 9 shows the mAPs with respect to the varying parameters on the UCM dataset with the fixed 64 code length. From the results, we can see that the performance of MvIGH increases first and then decreases with the the increase of λ. The mAP results remain relatively stable with the change of r_1, r_2, and k.

TABLE VII
TRAINING TIME ON NWPU DATASET

Method	SH	CHMIS	MFH	MFKH	SU-MVSH	CMFH	MvIGH
Time(s)	0.14	1092.59	588.86	19.77	1.42	4.59	29.14



(a) (b) (c) (d)

Fig. 9. Influence of different parameters. (a) λ, (b) r_1, (c) r_2, and (d) k.

V. CONCLUSION

This article studies how to learn compact hash codes that explore the latent similarity structure among multiview RS images. The proposed MvIGH adaptively learns the weights of the multiple views to capture the latent similarity structures among RS images. In addition, MvIGH uses a new regularizer to explicitly reduce the quantization error. MvIGH is optimized in an iterative manner, and it is theoretically guaranteed to reach a local minimum. Extensive experiments on three published RS image datasets demonstrate the proposed method outperforms existing multiview hashing methods in large-scale RSIR tasks.

There are several interesting works that deserve further studies. First, the reported performance is limited as the multiview data are constructed by handcrafted visual features. Deep neural networks (CNN) can extract more useful information from remote sensing images. We will develop deep extension of the proposed method to improve retrieval performance. Second, this work assumes that all the views exist for each image, which, however, may not hold in real applications. How to deal with the incomplete multiview RS images is another challenging and interesting work.

REFERENCES

[1] Y. Xu, Z. Wu, F. Xiao, T. Zhan, and Z. Wei, “A target detection method based on low-rank regularized least squares model for hyperspectral images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1129–1133, Aug. 2016.

[2] Y. Xu, Z. Wu, J. Chanussot, M. D. Mura, A. L. Bertozzi, and Z. Wei, “Low-rank decomposition and total variation regularization of hyperspectral video sequences,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1680–1694, Mar. 2018.

[3] Y. Zheng, H. Song, L. Sun, Z. Wu, and B. Jeon, “Spatiotemporal fusion of satellite images via very deep convolutional networks,” *Remote Sens.*, vol. 11, no. 22, 2019, Art. no. 2701.

[4] C.-R. Shyu, M. Klaric, G. J. Scott, A. S. Barb, C. H. Davis, and K. Palaniappan, “Geoiris: Geospatial information retrieval and indexing system-content mining, semantics modeling, and complex queries,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 839–852, Apr. 2007.

[5] W. Liu and I. W. Tsang, “Making decision trees feasible in ultrahigh feature and label dimensions,” *J. Mach. Learn. Res.*, vol. 18, no. 81, pp. 1–36, 2017.

[6] L. Xie, D. Tao, and H. Wei, “Early expression detection via online multi-instance learning with nonlinear extension,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1486–1496, May 2019.

[7] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” in *Proc. Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[8] B. Kulis and K. Grauman, “Kernelized locality-sensitive hashing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.

[9] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[10] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, “Hashing with graphs,” in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[11] J. Wang, S. Kumar, and S.-F. Chang, “Semi-supervised hashing for large-scale search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[12] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[13] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, “Discrete graph hashing,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.

[14] K. Ding, C. Huo, B. Fan, and C. Pan, “kNN hashing with factorized neighborhood representation,” in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1098–1106.

[15] M. Zhang, F. Shen, H. Zhang, N. Xie, and W. Yang, “Hashing with inductive supervised learning,” in *Proc. Pacific-Rim Conf. Adv. Multimedia Inf. Process.*, 2015, pp. 447–455.

[16] X. Shen, W. Liu, I. W. Tsang, F. Shen, and Q. Sun, “Compressed k-means for large-scale clustering,” in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2527–2533.

[17] F. Shen, Y. Yang, L. Liu, W. Liu, D. Tao, and H. T. Shen, “Asymmetric binary coding for image search,” *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2022–2032, Sep. 2017.

[18] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang, “Unsupervised t-distributed video hashing and its deep hashing extension,” *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5531–5544, Nov. 2017.

[19] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and S. H. Tao, “Unsupervised deep hashing with similarity-adaptive and discrete optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[20] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” *CoRR*, vol. abs/1304.5634, 2013.

[21] Y. Li, Y. Zhang, X. Huang, and J. Ma, “Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6521–6536, Nov. 2018.

[22] W. Xiong, Z. Xiong, Y. Zhang, Y. Cui, and X. Gu, “A deep cross-modality hashing network for SAR and optical remote sensing images retrieval,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5284–5296, 2020, doi: 10.1109/JSTARS.2020.3021390.

[23] D. Zhang, F. Wang, and L. Si, “Composite hashing with multiple information sources,” in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 225–234.

[24] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, “Multiple feature hashing for real-time large scale near-duplicate video retrieval,” in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 423–432.

[25] X. Liu, J. He, D. Liu, and B. Lang, “Compact kernel hashing with multiple features,” in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 881–884.

[26] S. Kim, Y. Kang, and S. Choi, “Sequential spectral learning to hash with multiple representations,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 538–551.

[27] L. Liu, M. Yu, and L. Shao, “Multiview alignment hashing for efficient image search,” *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.

[28] X. Shen, F. Shen, Q.-S. Sun, Y. Yang, Y.-H. Yuan, and H. T. Shen, “Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view retrieval,” *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4275–4288, Dec. 2017.

[29] X. Shen, W. Liu, I. Tsang, Q.-S. Sun, and Y.-S. Ong, “Multilabel prediction via cross-view search,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4324–4338, Sep. 2018.

[30] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, “A fast optimization method for general binary code learning,” *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5610–5621, Dec. 2016.

[31] M. Datcu *et al.*, “Information mining in remote sensing image archives: System concepts,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2923–2936, Dec. 2003.

[32] Y. Yang and S. Newsam, “Geographic image retrieval using local invariant features,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 818–832, Feb. 2013.

[33] X. Tang, X. Zhang, F. Liu, and L. Jiao, “Unsupervised deep feature learning for remote sensing image retrieval,” *Remote Sens.*, vol. 10, no. 8, 2018, Art. no. 1243. [Online]. Available: https://www.mdpi.com/2072-4292/10/8/1243

[34] G. Scott and C.-R. Shyu, “Knowledge-driven multidimensional indexing structure for biomedical media database retrieval,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 3, pp. 320–331, May 2007.

[35] X. Tang, L. Jiao, W. J. Emery, F. Liu, and D. Zhang, “Two-stage reranking for remote sensing image retrieval,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5798–5817, Oct. 2017.

[36] X. Tang, L. Jiao, and W. J. Emery, “SAR image content retrieval based on fuzzy similarity and relevance feedback,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1824–1842, May 2017.

[37] J. Kang, R. Fernandez-Beltran, Z. Ye, X. Tong, and A. Plaza, “Deep hashing based on class-discriminated neighborhood embedding,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5998–6007, 2020, doi: 10.1109/JSTARS.2020.3027954.

[38] C. Liu, J. Ma, X. Tang, F. Liu, X. Zhang, and L. Jiao, “Deep hash learning for remote sensing image retrieval,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3420–3443, Apr. 2021.

[39] X. X. Zhu *et al.*, “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[40] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, “Supervised hashing with kernels,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2074–2081.

[41] B. Demir and L. Bruzzone, “Hashing-based scalable remote sensing image search and retrieval in large archives,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 892–904, Feb. 2016.

[42] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 977–986.

[43] C. Hou, F. Nie, H. Tao, and D. Yi, “Multi-view unsupervised feature selection with adaptive similarity and view weight,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1998–2011, Sep. 2017.

[44] W. Liu, J. He, and S. Chang, “Large graph construction for scalable semi-supervised learning,” in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 679–686.

[45] X. Chen and D. Cai, “Large scale spectral clustering with landmark-based representation,” in *Proc. AAAI Conf. Artif. Intell.*, 2011, pp. 313–318.

[46] J. Nocedal and S. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 2006.

[47] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.

[48] G.-S. Xia *et al.*, “Aid: A benchmark data set for performance evaluation of aerial scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

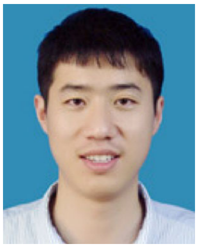
[49] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *in Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

Yinghui Sun received the B.Sc. degree in mathematics with finance from the University of Liverpool, Liverpool, U.K., in 2015, and the M.Sc. degree in statistics from the University of Bristol, Bristol, U.K., in 2017. She is currently working toward the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.

Her primary research interests include graph representation learning, multiview learning, and hashing.

Wei Wu received the B.Sc. degree in computer science and technology in 2020 from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, where she is currently working toward the postgraduate degree in computer science and technology.

Her primary research interests include distributed learning, multiview learning, and hashing.



Xiaobo Shen (Member, IEEE) received the B.Sc. degree in software engineering and Ph.D. degree in pattern recognition and intelligent system both from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2011 and 2017, respectively.

He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He has authored over 50 technical papers in prominent journals and conferences, such as IEEE TNNLS, IEEE TIP, IEEE TCYB, ICML, NIPS, ACM MM, AAAI, and IJCAI. His primary research interests include machine learning and pattern recognition.

Zhen Cui (Member, IEEE) received the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences, Beijing, China, in 2014.

He was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore from September 2014 to November 2015. He also spent half a year as a Research Assistant with Nanyang Technological University (NTU), Singapore, from June 2012 to December 2012. Currently, he is a Professor with the Nanjing University of Science and Technology, Nanjing, China. His research interests include computer vision, pattern recognition and machine learning, especially focusing on vision perception and computation, graph deep learning, etc.