




Article

Low-Rank Hypergraph Hashing for Large-Scale Remote Sensing Image Retrieval

Jie Kong ¹, Quansen Sun ^{1,*}, Mithun Mukherjee ² and Jaime Lloret ³

¹ School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; kongjiepr@njjust.edu.cn

² Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China; m.mukherjee@ieee.org

³ Department of Communications, Polytechnic University of Valencia, 46022 Camino de Vera, Valencia, Spain; jlloret@dcom.upv.es

* Correspondence: sunquansen@njjust.edu.cn

Received: 11 February 2020; Accepted: 2 April 2020; Published: 4 April 2020



Abstract: As remote sensing (RS) images increase dramatically, the demand for remote sensing image retrieval (RSIR) is growing, and has received more and more attention. The characteristics of RS images, e.g., large volume, diversity and high complexity, make RSIR more challenging in terms of speed and accuracy. To reduce the retrieval complexity of RSIR, a hashing technique has been widely used for RSIR, mapping high-dimensional data into a low-dimensional Hamming space while preserving the similarity structure of data. In order to improve hashing performance, we propose a new hash learning method, named low-rank hypergraph hashing (LHH), to accomplish for the large-scale RSIR task. First, LHH employs a $l_{2,1}$ norm to constrain the projection matrix to reduce the noise and redundancy among features. In addition, low-rankness is also imposed on the projection matrix to exploit its global structure. Second, LHH uses hypergraphs to capture the high-order relationship among data, and is very suitable to explore the complex structure of RS images. Finally, an iterative algorithm is developed to generate high-quality hash codes and efficiently solve the proposed optimization problem with a theoretical convergence guarantee. Extensive experiments are conducted on three RS image datasets and one natural image dataset that are publicly available. The experimental results demonstrate that the proposed LHH outperforms the existing hashing learning in RSIR tasks.

Keywords: hashing; low-rank; hypergraph; remote sensing image retrieval

1. Introduction

With the development of satellite technology, the quality of remote sensing (RS) images increases dramatically. Retrieving similar RS images from large-scale RS datasets is very important and demanding [1–3]. Interestingly, content-based image retrieval (CBIR) [4–6] is widely involved in many real-world tasks, such as natural image retrieval and network searches. Nevertheless, large variations are usually contained in the RS images due to their large data volume, small object size and rich background [7,8], and thus how to extract valuable information and further adapt existing CBIR methods to remote sensing image retrieval (RSIR) is considered a key issue [9,10].

Hashing learning has become more and more important for large-scale retrieval, due to its superiority in terms of computation and storage [11–13]. In recent years, several hashing-based methods have been proposed for large-scale RSIR tasks [14–18]. Partial randomness hashing (PRH) [14] is proposed to employ random projections to map images to a low-dimensional Hamming space, and trains a linear model for mapping from the Hamming space back to the original space. Demir et al.

introduces two kernel-based methods to learn distinctive hash functions in the kernel space [15]. Liu et al. proposes a deep supervised hashing (DSH) method to learn compact binary code by fully employing deep learning and hashing learning [16]. To introduce deep neural networks (DHNNs) into large-scale RSIR tasks, Li et al. conducts a comprehensive study of DHNN systems [17]. To capture intra-class distribution and inter-class ranking, Fan et al. proposes a distribution consistency loss (DCL) to extract informative data and build a more informative structure [18]. These approaches only utilize the pairwise similarity to capture the relationships among data, although the relationships among RS images are more complex and high-order.

A natural way of capturing the complex structure among RS images is a hypergraph. Hypergraphs [19] generalize conventional graphs, where one edge can connect more than two vertices. Therefore, hypergraphs can capture complex and high-order relationships, and have been used in image annotation, image ranking and feature selection [19,20]. Recently, hypergraph spectral hashing (HSH) methods [21–23] have received considerable attention. For example, hypergraph spectral learning [21] is proposed for multi-label classification. To further exploit the correlation information, [22] introduces a transductive learning framework based on a probabilistic hypergraph. [23] applies a hypergraph in conventional spectral hashing for searching social images. Although these methods improve the performance with hypergraphs, all of them process with no label information. In addition, the noise of features and sample are ignored in these methods.

To address the aforementioned problems, new LHH is proposed to deal with large-scale RSIR tasks. The flowchart of the proposed LHH is shown in Figure 1, where we see that the proposed LHH is a shallow model, although its key components can also be easily considered in deep extensions of LHH. The main contributions of the proposed LHH are summarized as follows:

- (1) The LHH employs a l_{2-1} norm to constrain the projection matrix to reduce the noise and redundancy among features. In addition, low-rankness is also imposed on the projection matrix to exploit its global structure.
- (2) The LHH exploits the hypergraph to capture the high-order relationship among data, and is very suitable to explore the complex structure of RS images.
- (3) Finally, the proposed LHH is evaluated on three large-scale remote sensing datasets and one natural image dataset. The experimental results show that the proposed LHH outperforms some existing hashing methods in large-scale RSIR tasks.

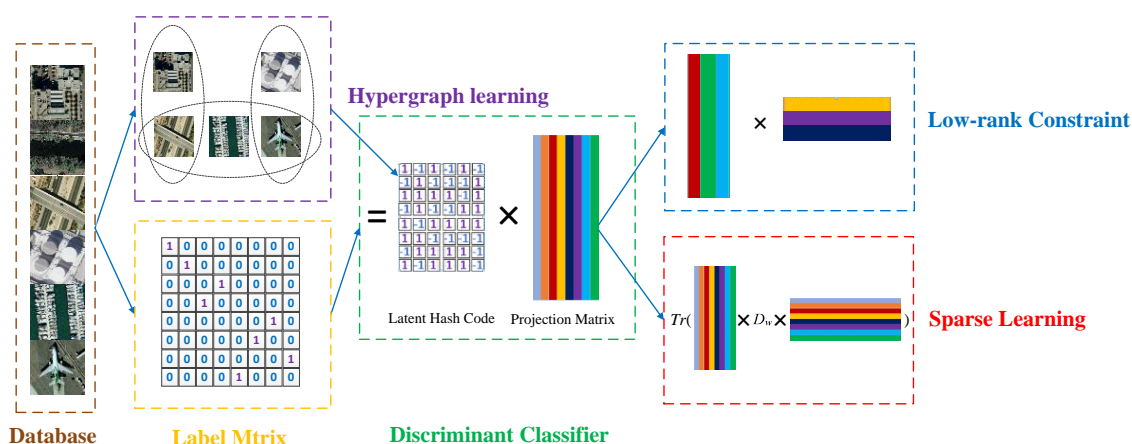


Figure 1. Illustration of the proposed low-rank hypergraph hashing (LHH).

The rest of the paper is organized as follows. The notation and related work are presented in Section 2. The proposed method is discussed in Section 3. The extensive experimental evaluations are presented in Section 4. Finally, a conclusion is given in Section 5.

2. Notation and Related Work

The notation of the paper and the recent advances of hashing techniques, low-rank analysis and hypergraph learning are reviewed in this section.

2.1. Notation

In this paper, we represent matrix and vector as a boldface italic letter, and scalar as a normal italic letter. For a matrix $\mathbf{X} = [x_{ij}]$, its i -th row and j -th column are denoted as \mathbf{x}^i and x_j , respectively. We represent the transpose operator, the inverse and the trace operator of \mathbf{X} as \mathbf{X}^T , \mathbf{X}^{-1} and $\text{tr}(\mathbf{X})$, respectively. We represent the Frobenius norm and $l_{2,1}$ -norm of \mathbf{X} , respectively. The important notations in the paper are summarized in Table 1.

Table 1. Important notation used in this paper.

Notation	Description
\mathbf{X}	Data matrix
\mathbf{Y}	Label matrix
\mathbf{W}	Projection matrix
\mathbf{A}	Hash code basis matrix
\mathbf{B}	Hash code matrix
\mathbf{C}	Hash code coefficient matrix
\mathbf{H}	Hypergraph similarity matrix
\mathbf{L}	Hypergraph Laplace matrix
\mathbf{M}	Data basis matrix
\mathbf{N}	Data coefficient matrix
n	Sample number
m	Sample dimensionality
n	Class number
l	Hash code length
r	Rank of projection matrix
k	Hyperedge number
t	Iteration number
λ_1	Sparse regularization parameter
λ_2	Hypergraph regularization parameter

2.2. Hashing Learning

Hashing has been a key step to facilitate large-scale image retrieval [24]. Essentially, hashing maps the high-dimensional data into a low-dimensional Hamming space while preserving similarity structure among data. The representative hashing methods include local sensitive hashing (LSH) [25], spectral hashing (SH) [26], partial randomness hashing (PRH) [14], supervised hashing with kernels (KSH) [27] and supervised discrete hashing (SDH) [12]. LSH [25] obtains hash functions via random projections. Original metrics are theoretically guaranteed to be preserved in the Hamming space, and LSH often requires long code length to get high precision. The SH [26] preserves the similarity distribution among data and the binary codes are imposed to be balanced and uncorrelated. PRH [14] establishes a partial stochastic strategy to enable good approximation and fast learning to construct hashing functions. KSH [27] maps the data into binary codes whose Hamming distance are minimized on similar pairs and maximized on dissimilar pairs. Based on the equivalence between optimizing the code inner product and Hamming distance, KSH can train the hash function efficiently. SDH [12] aims to learn the hash codes that are good for classification. To deal with the non-deterministic polynomial hard (NP-hard) binary constraint, the SDH develops a cyclic coordinate descent to generate good hash codes, which admits an analytical solution.

2.3. Low-Rankness Analysis

The low-rankness property is attracting more and more attention, since it enables the finding of low-rank structures from high-dimensional data that is corrupted with noise and outliers [28,29]. With low-rank constraints, the computational complexity can be greatly reduced. [30] has proved that low-rank regression equals regression in the subspace generated by linear discriminant analysis (LDA). Low-rank representation (LRR) [31] seeks to find the lowest rank representation where data can be represented by linear combinations of the basis in a given dictionary. To further enhance LRR, latent low-rank representation (LatLRR) [32,33] is proposed to recover the unobserved data in LRR.

Since low-rank optimization problem is a NP-hard problem, we instead solve the nuclear norm minimization problem via alternating direction method of multipliers (ADMM) [34]. Low-rank can be used in matrix decomposition due to its advantage in "de-correlation". Specifically, given a matrix \mathbf{X} with the rank r , low-rank decomposition solves the following optimization problems:

$$\begin{aligned} & \underset{M,N}{\operatorname{argmin}} \mathbf{X} \approx \mathbf{MN} \\ & \text{s.t. } r < \min\{m, n\}, \mathbf{X} \in \mathbb{R}_{m \times n}, \mathbf{M} \in \mathbb{R}_{m \times r}, \mathbf{N} \in \mathbb{R}_{r \times n} \end{aligned} \quad (1)$$

It is evident that the size of \mathbf{X} is larger than the sum of those of \mathbf{M} and \mathbf{N} . Therefore, the low rank matrix decomposition can remove the correlation among data, thereafter reducing the storage.

2.4. Hypergraph Learning

Since conventional graphs fail to exploit the relationships among data, the hypergraph has been widely used to characterize the complex relationships among complex data [22,23]. Specifically, a hypergraph generalizes the conventional graph where one edge can connect more than two vertices and capture high-order information. Different from simple graphs, hypergraphs contain local grouping information that is beneficial to clustering. In [35], Huang et al. construct hyperedges among images based on shape and appearance features in their region of interests (ROI), and perform spectral clustering for unsupervised image categorization. In [14], a transductive learning framework is introduced to further explore the correlations. This approach constructs a probabilistic hypergraph, and hypergraph ranking is further employed. To accelerate similarity search, the authors in [23] extend the traditional unsupervised hashing method to a hypergraph to capture the high-order information for social images. Besides, Bu et al. have modeled the relationship of different entities using the hypergraph for recommendation in the social-media community [36].

3. Proposed Low-Rank Hypergraph Hashing (LHH) Method

This section first introduces the proposed LHH. Then, Section 3.1 gives the notation and problem statement. The details of the proposed LHH are presented in Section 3.2. After this, Section 3.3 presents the optimization of the proposed LHH. Moreover, Section 3.4 introduces learning the hashing function. Finally, Section 3.5 presents the convergence analysis.

3.1. Problem Statement

Suppose that $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^n$ is a set of images, and we are given its feature $\mathbf{X} = [\mathbf{x}^1; \dots; \mathbf{x}^n] \in \mathbb{R}_{n \times m}$, where m is the dimensionality and n is the number of the images. We represent the hash code matrix $\mathbf{B} = [\mathbf{b}^1; \dots; \mathbf{b}^n] \in \{-1, 1\}^{n \times l}$, where $\mathbf{b}^i \in \{-1, 1\}^{1 \times l}$ is the hash code of \mathbf{o}_i and l is the code length. The hash function $\mathbf{HF}(\mathbf{x}) = \operatorname{sgn}(\mathbf{F}(\mathbf{x}))$ encodes \mathbf{x} by l -bit hash code, where $\operatorname{sgn}(\cdot)$ is the sign function, which outputs $+1$ for positive numbers and -1 otherwise. The LHH aims to learn \mathbf{B} and the hash function \mathbf{HF} to preserve the similarity structure of the images.

3.2. Low-Rank Hypergraph Hashing

To consider the supervised information, we regard learning the binary codes in the context of classification. We enable the binary codes to be optimal for the jointly learned classifier. Thus, the good binary codes are ideal for the classification.

Given binary code \mathbf{b} , we adopt the following multi-class classification formulation

$$\mathbf{y} = \mathbf{G}(\mathbf{b}) = \mathbf{b}\mathbf{W} = [\mathbf{b}w_1, \dots, \mathbf{b}w_c] \quad (2)$$

where $\mathbf{W} = [w_1, \dots, w_c] \in \mathbb{R}^{l \times c}$, $w_k \in \mathbb{R}^{l \times 1}$, $k = 1, \dots, c$ is the projection for class k and $\mathbf{y} \in \mathbb{R}^{1 \times c}$ is the label vector, and the maximum value indicates the assigned class of \mathbf{x} .

We choose to optimize the following problem

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{W}} \sum_{i=1}^n L(\mathbf{y}_i - \mathbf{b}_i \mathbf{W}) + \lambda_1 R(\mathbf{W}) \\ \text{s.t. } \mathbf{b}_i \in \{-1, 1\}^{1 \times l}, i = 1, \dots, n \end{aligned} \quad (3)$$

where $L(\cdot)$ is the loss function, $R(\mathbf{W})$ is a regularizer and λ_1 is a regularization parameter. $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n \in \mathbb{R}^{n \times c}$ is the ground truth label matrix, where $y_{ik}=1$ if x_i belongs to class k and 0 otherwise.

Equation (3) is flexible, and we can define any loss function for $L(\cdot)$. For simplicity, we can choose the simple l_2 loss, which minimizes the difference between the label \mathbf{Y} and prediction $\mathbf{G}(\mathbf{b})$. The problem in Equation (3) can be transformed into the following problem:

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{W}} \|\mathbf{Y} - \mathbf{B}\mathbf{W}\|_F^2 + \lambda_1 \|\mathbf{W}\|_F^2 \\ \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times l} \end{aligned} \quad (4)$$

To enable the coefficients of data in the same space to be highly correlated, we apply the low rank constraint to capture the global structure of the whole data. In addition, the low-rank structure can relieve the impact from noises, and makes regression more accurate [37,38]. In order to consider the low-rank structure of \mathbf{W} , we need to make:

$$\text{rank}(\mathbf{W}) = r \leq \min(l, c) \quad (5)$$

We decompose \mathbf{W} into two low-rank matrices, i.e., $\mathbf{W} = \mathbf{A}\mathbf{C}$, where $\mathbf{A} \in \mathbb{R}^{l \times r}$, $\mathbf{C} \in \mathbb{R}^{r \times c}$, and r is the rank of \mathbf{W} . Then, Equation (4) can be further transformed into

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathbf{Y} - \mathbf{B}\mathbf{A}\mathbf{C}\|_F^2 + \lambda_1 \|\mathbf{A}\mathbf{C}\|_F^2 \\ \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times l}, \mathbf{A}\mathbf{A}^T = \mathbf{I} \end{aligned} \quad (6)$$

where $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ ($\mathbf{I} = \mathbb{R}^{r \times r}$), which is introduced for identifiability. Besides, we additionally enforce the sparsity, i.e., l_{21} -norm for feature selection by [39]. Thus, the above problem is rewritten as:

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathbf{Y} - \mathbf{B}\mathbf{A}\mathbf{C}\|_F^2 + \lambda_1 \|\mathbf{A}\mathbf{C}\|_{2,1}^2 \\ \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times l}, \mathbf{A}\mathbf{A}^T = \mathbf{I} \end{aligned} \quad (7)$$

In Equation (7), we consider both low-rankness and sparsity to learn the regression coefficient matrix. Low-rankness deals with the noises, and the $l_{2,1}$ -norm selects features by setting some rows of \mathbf{W} to be zero.

Until now, we do not consider the similarity structure among data. If two samples are similar, we need to ensure that two corresponding binary codes are close. To preserve the original local similarity structure, we aim to minimize

$$\min Rf(\mathbf{W}) = \frac{1}{2} \sum_{i,j}^n s_{i,j} \|\hat{\mathbf{y}}^i - \hat{\mathbf{y}}^j\|_2^2 \tag{8}$$

where $\mathbf{S} (s_{i,j} \in \mathbf{S})$ is the similarity matrix that records similarities among data, in which $s_{i,j}$ represents the relationship between the i -th and the j -th sample. Normally, we use the following formulation to construct graphs

$$f(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{\|\mathbf{a} - \mathbf{b}\|_2^2}{2\sigma^2}\right) \tag{9}$$

where σ is the kernel width and the term $\|\mathbf{a} - \mathbf{b}\|_2^2$ denotes the distance between two samples.

Here we instead use the hypergraph to measure the similarity among data. Figure 2 shows the distinction between a normal graph and hypergraph. As can be seen, the normal graph only connects two samples, while a hypergraph can connect more than two samples. Therefore, a hypergraph can reveal more complex relationships among data [23]. We formulate the incidence matrix \mathbf{H} between the vertices and the hyperedges of the hypergraph as:

$$\mathbf{H}(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

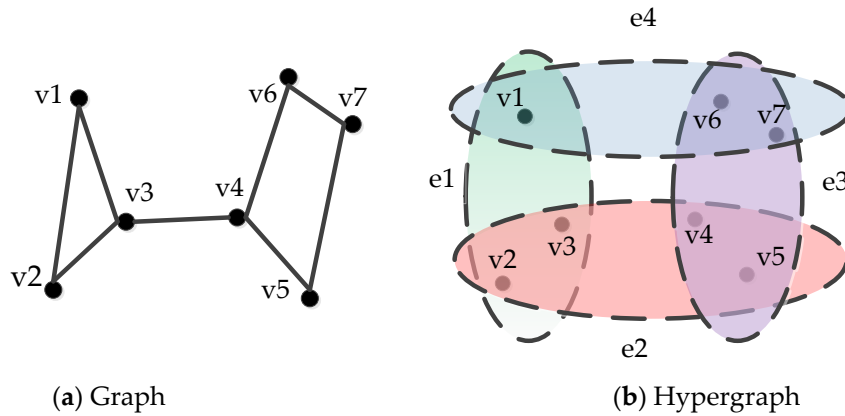


Figure 2. Illustration of a graph and hypergraph.

The degree $d(v)$ of vertex v and the degree $\delta(e)$ of hyperedge are defined as follows:

$$\begin{aligned} d(v) &= \sum_e S(e)\mathbf{H}(v, e) \\ \delta(e) &= \sum_v \mathbf{H}(v, e) \end{aligned} \tag{11}$$

With the above definition, the normalized distance between v_i and v_j on e_k is $\frac{w(e_k)}{\delta(e_k)} \left(\frac{y_i}{\sqrt{d(v_i)}} - \frac{y_j}{\sqrt{d(v_j)}} \right)^2$. To preserve the similarity of hash codes, we aim to map data on the same hyperedge into more similar hash codes. Thus, we seek the hash codes by minimizing the average Hamming distance between hash codes of data on the same hyperedge:

$$\begin{aligned} \min_B & \frac{1}{2} \sum_{e_k \in E} \sum_{v_i, v_j \in e_k} \frac{S(e_k)}{\delta(e_k)} \left(\frac{b_i}{\sqrt{d(v_i)}} - \frac{b_j}{\sqrt{d(v_j)}} \right)^2 \\ \text{s.t. } & \mathbf{b}_i \in \{-1, 1\}^{1 \times l}, i = 1, \dots, n \end{aligned} \tag{12}$$

By introducing the hypergraph Laplacian, we further rewrite Equation (12) as

$$\begin{aligned} & \min_{\mathbf{B}} \text{tr}(\mathbf{B}^T \mathbf{L} \mathbf{B}) \\ & \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times l} \end{aligned} \tag{13}$$

where the hypergraph Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}}$, \mathbf{I} is the identity matrix, \mathbf{H} is the incidence matrix and \mathbf{D}_v and \mathbf{D}_e are diagonal matrices, where the diagonal element of \mathbf{D}_v and \mathbf{D}_e are degrees of the hypergraph vertex $d(v_i)$ and hyperedge $\delta(e_i)$, respectively.

Combining Equations (7) and (13), the final objective function of LHH is defined as:

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathbf{Y} - \mathbf{BAC}\|_F^2 + \lambda_1 \|\mathbf{AC}\|_{2,1} + \lambda_2 \text{tr}(\mathbf{B}^T \mathbf{L} \mathbf{B}) \\ & \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times l}, \mathbf{AA}^T = \mathbf{I} \end{aligned} \tag{14}$$

where λ_2 is a regularization parameter. In Equation (14), to learn high-quality binary codes, the first term learns the classifier with a binary code, the second term minimizes the $l_{2,1}$ -norm of the projection matrix to explore its low-rankness and sparsity, and the third term preserves the intrinsic complex structure of data via a hypergraph.

3.3. Optimization Algorithm

It is clear that Equation (14) is difficult to find a global solution for, as it is nonconvex. We alternatively solve the sub-problems for the following variables.

(1) C-step: Update \mathbf{C} by fixing \mathbf{A} and \mathbf{B} .

Algorithm 1 Curvilinear Search Algorithm Based on Cayley Transformation

Input: initial point $\mathbf{A}^{(0)} \in \mathbb{R}^{l \times r}$, matrix \mathbf{B} , \mathbf{C} , hash code length l

Output: $\mathbf{A}^{(t)}$.

- 1: Initialize $t = 0$, $\varepsilon > 0$ and $\lambda_1 = 1, \lambda_2 = 1e - 2$.
 - 2: **Repeat**
 - 3: Compute the gradient according to (18);
 - 4: Generate the skew-symmetric matrix $\mathbf{F} = \mathbf{G}^T \mathbf{A} - \mathbf{A}^T \mathbf{G}$;
 - 5: Compute the step size τ_t , that satisfies the Armijo-Wolfe conditions [33] via the line search along the path $\mathbf{J}_t(\tau)$ defined by (19);
 - 6: Set $\mathbf{A}^{(t+1)} = \mathbf{J}(\tau_t)$;
 - 7: Update $t = t + 1$;
 - 8: **Until** convergence
-

In this case, the objective function is simplified as:

$$\min_{\mathbf{C}} \text{Loss}(\mathbf{C}) = \|\mathbf{Y} - \mathbf{BAC}\|_F^2 + \lambda_1 \|\mathbf{AC}\|_{2,1} \tag{15}$$

Equation (15) can be rewritten as:

$$\min_{\mathbf{C}} \text{Loss}(\mathbf{C}) = -2\text{tr}(\mathbf{C}^T \mathbf{A}^T \mathbf{B}^T \mathbf{Y}) + 2\text{tr}(\mathbf{C}^T \mathbf{A}^T \mathbf{B}^T \mathbf{BAC}) + \lambda_1 \|\mathbf{AC}\|_{2,1} \tag{16}$$

We have the derivative of Equation (16) with respect to \mathbf{C} equal to 0, and receive

$$\mathbf{C} = (\mathbf{A}^T \mathbf{B}^T \mathbf{B} \mathbf{A} + \lambda_1 \mathbf{A}^T \mathbf{D}_w \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B}^T \mathbf{Y} \tag{17}$$

where $\mathbf{D}_W = \frac{1}{2\|w^i\|_2} = \frac{1}{2\|(\mathbf{AC})^i\|_2}$ is a diagonal matrix.

(2) A-step: Update \mathbf{A} by fixing \mathbf{B} and \mathbf{C} .

It is hard to obtain an optimal solution in Equation (14) with respect to \mathbf{A} , due to the orthogonal constraint. Here we apply a gradient descent with a curvilinear search to seek a locally optimal solution.

First, we denote G as the gradient of Equation (16) with respect to A , and it is defined as:

$$G = -2B^T Y C^T + 2B^T B A C C^T + 2\lambda_1 D_W A C C^T \tag{18}$$

A skew-symmetric matrix is defined as $F = G^T A - A^T G$. The next point is decided by a Crank-Nicolson scheme

$$J(\tau) = A - \frac{\tau}{2} F^T (A + J(\tau)) \tag{19}$$

where τ is the step size. We can get a closed-form solution of $J(\tau)$:

$$\begin{aligned} J(\tau) &= AM, \\ M &= \left(I - \frac{\tau}{2} F^T \right) \left(I + \frac{\tau}{2} F^T \right)^{-1} \end{aligned} \tag{20}$$

Here, Equation (20) is called the Cayley transformation [33,40,41]. The iteration terminates when τ_t satisfies the Armijo-Wolfe condition. The algorithm solving the sub-problem is illustrated in Algorithm 1.

(3) **B**-step: Update B by fixing A and C .

The objective function is simplified as follows:

$$\begin{aligned} \min_B \text{Loss}(B) &= \text{tr}[(Y - BAC)^T (Y - BAC)] + \lambda_2 \text{tr}(B^T L B) \\ \text{s.t. } B &\in \{-1, 1\}^{n \times l} \end{aligned} \tag{21}$$

The above problem is challenging due to the discrete constraint, and it has no closed-form solution. Inspired by the recent study in nonconvex optimization, we optimize Equation (21) with the proximal gradient method, which iteratively optimizes a surrogate function. In the j -th iteration, we define a surrogate function $\hat{\text{Loss}}_j(B)$ that is a discrete approximation of $\text{Loss}(B)$ at the point $B^{(j)}$. Given $B^{(j)}$, the next discrete point is obtained by optimizing:

$$B^{(j+1)} \in \text{argmax}_{B \in \{-1, 1\}^{n \times l}} \hat{\text{Loss}}_j(B) := \text{Loss}(B^{(j)} + \langle \nabla \text{Loss}(B^{(j)}), B - B^{(j)} \rangle) \tag{22}$$

Note that $\nabla \text{Loss}(B^{(j)})$ may include zero entries and that multiple solutions for $B^{(j+1)}$ may exist, thus we introduce function $\text{Cf}(x, y) = \begin{cases} x, & x \neq 0 \\ y, & y = 0 \end{cases}$ to eliminate the zero entries. The updated rule for $B^{(j+1)}$ is defined as [42,43]:

Algorithm 2 Low-Rank Hypergraph Hashing

Input: label matrix $Y \in \mathbb{R}^{n \times c}$, hash code length l , hyperedge number k ;

Output: $A^{(t)}$, $B^{(t)}$, $C^{(t)}$;

1: Initialize $A^{(0)} \in \mathbb{R}^{l \times r}$, $B^{(0)} \in \{-1, 1\}^{n \times l}$, $C^{(0)} \in \mathbb{R}^{r \times c}$, $H \in \mathbb{R}^{n \times k}$;

2: Initialize $t=0$, $\varepsilon > 0$, and $\lambda_1 = 1$, $\lambda_2 = 1e-2$;

3: **Repeat**

4: **C**-step: Update $C^{(t)}$ using (17);

5: **A**-step: Update $A^{(t)}$ by Algorithm 1;

6: **B**-step: Update $B^{(t)}$ using (23);

7: Update $t=t+1$;

8: **Until** convergence

$$B^{(j+1)} \text{sgn}(C(\nabla \text{Loss}(B^{(j)}), B^{(j)})) = \text{sgn}(Cf(-2Y(AC)^T + 2B(AC)(AC)^T + 2\lambda_2 LB, B^{(j)})) \tag{23}$$

The learning algorithm of LHH is shown in Algorithm 2.

3.4. Hash Function Learning

The optimal hash code has been learned, and we need to further learn a mapping from the original space to Hamming space. Here we assume that there is a linear mapping between the two spaces, and the transformation matrix is learned by optimizing the following problem [12]:

$$\min_P \|B - XP\|_F^2 \quad (24)$$

In Equation (24), it measures the fitting error between data and hash codes. The solution of the problem admits the following form:

$$P = (X^T X)^{-1} X^T B \quad (25)$$

Finally, the hash function is defined as

$$HF = \text{sgn}(xP), \quad (26)$$

where x is an arbitrary sample.

3.5. Convergence Analysis and Computational Complexity Analysis

Firstly, we discuss the convergence of LHH, which is presented in the following theorem.

Theorem 1: The alternating iteration scheme of Algorithm 2 monotonically reduces the objective function value of Equation (14), and Algorithm 2 converges to a local minimum of Equation (14).

Proof: LHH includes three sub-problems. The sub-problem **C** is convex, thus it clearly has the optimal solution. The sub-problems with respect to **A** and **B** are non-convex, but **A** and **B** steps decrease the objective function value. Thus, Algorithm 2 decreases the objective function value in each step. In addition, the objective function value is non-negative. Thus, Algorithm 2 can converge to a local optimal solution of LHH.

Then, we present the computational complexity of the proposed LHH method. The computational complexity of LHH mainly consists of the following several parts. In the step of updating **A**, its complexity is $O(nlr)$. In the step of updating **A**, due to the orthogonal constraint, we use the Cayley transformation for solving this problem. Computing the gradient of **A** requires $(nl^2 + nlc)$ and updating **A** for each iteration is $O(4nl^2 + l^3)$ [40]. Thus, the complexity of optimizing **A** is $O(t1(nl^2 + nlc + 4nl^2 + l^3))$, where $t1$ is the number of iterations for updating **A**. In the step of updating **B**, its complexity is $O(n^2l)$, and it is time-consuming, as it contains hypergraph Laplacian matrix computing. In summary, the total computational complexity of LHH is $O(t(nlr + n^2l + t1(nl^2 + nlc + 4nl^2 + l^3)))$, where t is the number of total iterations in Algorithm 2. Finally, the computational complexity of hashing mapping matrix **P** requires the time complexity of $O(nd^2 + ndl)$. For the query part, the computational cost for encoding any query x is $O(cd)$.

4. Experiments

We compare the proposed method with some state-of-the-art methods on four benchmark datasets, and their performance is evaluated in large-scale remote-sensing retrieval tasks.

4.1. Datasets

We adopt four benchmark datasets: UC Merced Land Use Dataset (UCMD) [44]; SAT4 [45]; SAT6 [45]; and CIFAR10 [46]. Their descriptions are as follows:

- UCMD is generated by manually labeling aerial image scenes, and it covers 21 land cover categories. More specifically, each land cover category includes 100 images of 256×256 pixels. The spatial resolution of this public domain imagery is 0.3 meters. Here we randomly sample 420 samples as the query set, and use the remaining 1680 samples for training.

- SAT4 consists of a total of 500,000 image patches covering four broad land cover classes. These include barren land, grassland, trees and a class that consists of all land cover classes other than the above three. Each image patch is size normalized to 28×28 pixels, and the spatial resolution of each pixel is 1 m. we randomly select 100,000 samples as the query set, and the other 400,000 samples as a training set.
- SAT6 consists of a total of 405,000 image patches covering six broad land cover classes. These include barren land, buildings, grassland, roads, trees and water bodies. The image size and spatial resolution of SAT6 are similar with these of SAT4. We randomly select 81,000 samples as the query set, and the other 324,000 samples as a training set.
- CIFAR10 dataset consists of sixty thousand 32×32 color images of 10 classes and 6,000 images in each class. We randomly select 10,000 samples as the query set, and the remaining 50,000 samples as a training set.

The statistics of the four datasets are summarized in Table 2, and some sample images are presented in Figure 3.

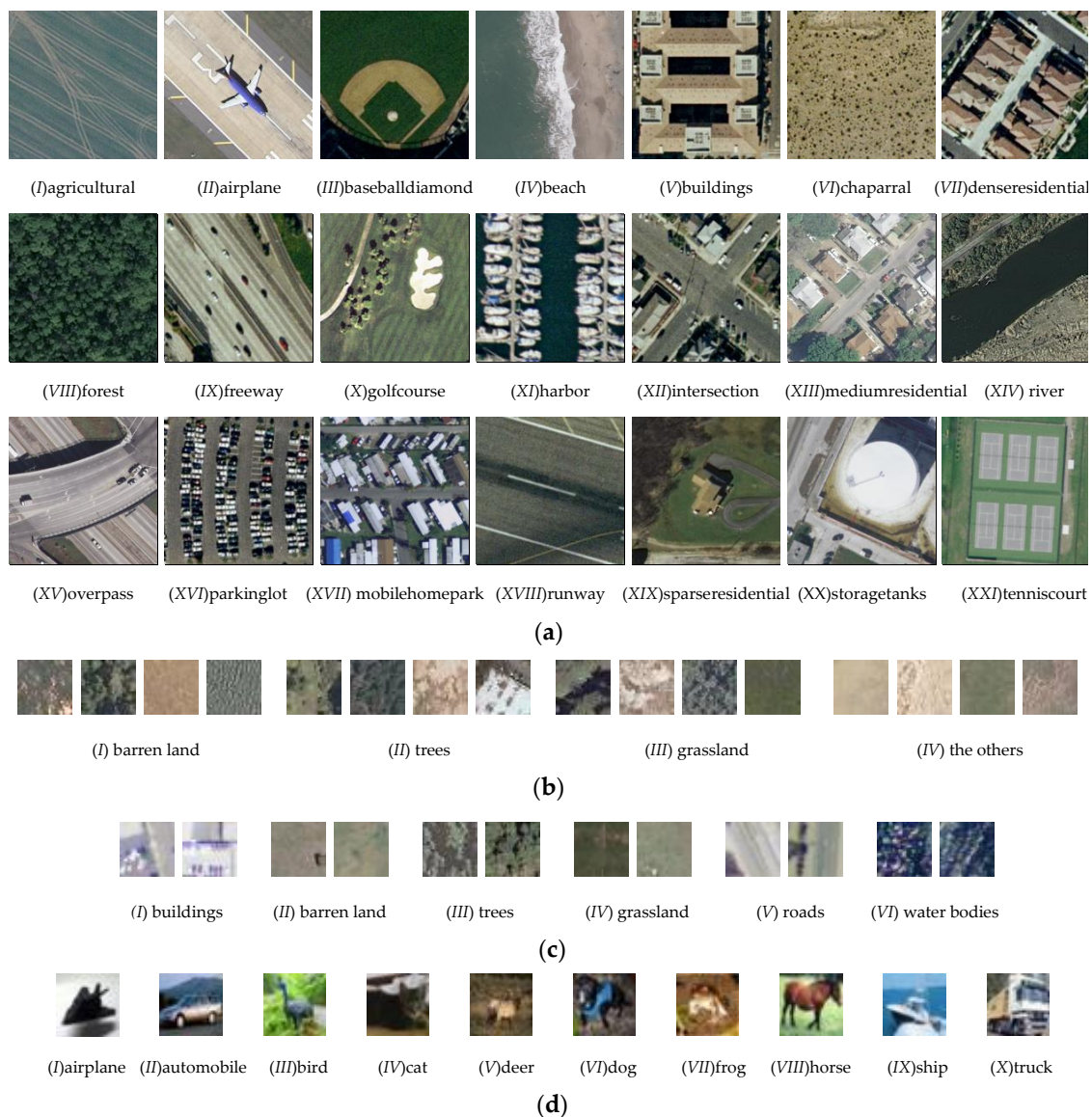


Figure 3. Sample images of four datasets: (a) UCMD; (b) SAT4; (c) SAT6; and (d) CIFAR10.

Table 2. Statistics and several parameter settings of four datasets.

	UCMD	SAT4	SAT6	CIFAR10
Dataset Size (n)	2100	500,000	405,000	60,000
Training Set	1680	400,000	324,000	50,000
Query Set	420	100,000	81,000	10,000
Image Size	256 × 256	28 × 28	28 × 28	32 × 32
Class Number (c)	21	4	6	10
Rank of Projection Matrix (r)	10	2	3	5
Hyperedge Number (k)	21	4	6	10

4.2. Experimental Setting

To verify the effectiveness of the proposed low-rank hypergraph hashing (LHH), we select several hashing methods for the performance comparison, including local sensitive hashing (LSH) [25], spectral hashing (SH) [26], partial randomness hashing (PRH) [14], HSH [21], supervised hashing with kernels (KSH) [27] and supervised discrete hashing (SDH) [12].

In the experiment, the samples are represented as 512-dimensional gistification (GIST) vectors. The experiments are conducted on a standard PC with Intel Core i7-8550U, CPU 2.70 GHz and 8GB RAM. In the experiment, λ_1 and λ_2 are empirically set as 1 and 0.01 respectively. Other parameter setting of the used four datasets is summarized in Table 2.

The retrieval performance is measured with two widely used metrics: mean average precision (mAP) and Precision-Recall (P-R) curve [47]. The mAP score is calculated by

$$mAP = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{L_q} \sum_{r=1}^R P_q(r) \delta_q(r) \quad (27)$$

where $q \in Q$ is a query, and $|Q|$ is the volume of query set. L_q is the number of the true neighbors in the retrieved list. $P_q(r)$ denotes the precision of the top r retrieved results, $\delta_q(r) = 1$ if the r -th result is the true neighbor, and 0 otherwise [48].

4.3. Performance Evaluation

4.3.1. Qualitative Analysis

We illustrate the retrieval results of several hashing methods on UCMD and CIFAR10 in Figures 4 and 5 respectively. Figure 4 illustrates the retrieved images of ‘building’ in UCMD, and Figure 5 illustrates the retrieved images of ‘dog’ in CIFAR10. The top nine images are returned, and the false images are with red rectangles.

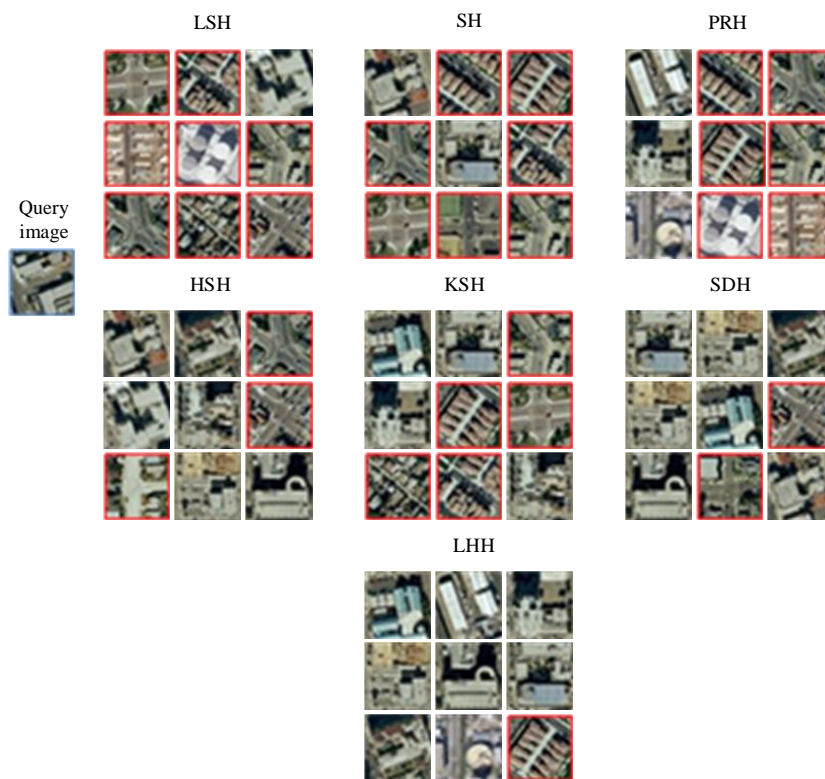


Figure 4. Visualized retrieval results on the UCMD by six hashing methods with hash codes of 32 bits.



Figure 5. Visualized retrieval results on the CIFAR10 by six hashing methods with hash codes of 32 bits.

From Figures 4 and 5, we can see that the proposed LHH can retrieve the most images among all the methods. LSH, SH and PRH retrieve 2–3 correct images, and HSH, KSH and SDH retrieve more than five similar images. The experiment validates the effectiveness of LHH.

4.3.2. Quantitative Analysis

(1) The comparison of mAP in several hashing methods under different datasets is shown in Tables 3–6.

Table 3. Comparison of mAP with different hash code lengths on UCMD.

Code Length \ Method	LSH	SH	PRH	HSH	KSH	SDH	LHH
8-bits	0.1256	0.1394	0.1354	0.2395	0.2791	0.2684	0.3102
16-bits	0.1375	0.1479	0.1483	0.2468	0.2913	0.2897	0.3385
32-bits	0.1421	0.1534	0.1594	0.2587	0.3147	0.3214	0.3573
64-bits	0.1476	0.1589	0.1718	0.2671	0.3278	0.3348	0.3760

Table 4. Comparison of mAP with different hash code lengths on SAT4.

Code Length \ Method	LSH	SH	PRH	HSH	KSH	SDH	LHH
8-bits	0.3147	0.3086	0.3980	0.5551	0.5037	0.5467	0.6046
16-bits	0.3175	0.3149	0.4023	0.5632	0.5112	0.5589	0.6295
32-bits	0.3197	0.3218	0.4095	0.5729	0.5268	0.5727	0.6573
64-bits	0.3209	0.3256	0.4078	0.5756	0.5189	0.5671	0.6922

Table 5. Comparison of mAP with different hash code lengths on SAT6.

Code Length \ Method	LSH	SH	PRH	HSH	KSH	SDH	LHH
8-bits	0.3387	0.3242	0.5054	0.6237	0.5941	0.6189	0.6592
16-bits	0.3438	0.3317	0.5117	0.6319	0.6011	0.6348	0.6727
32-bits	0.3417	0.3478	0.5172	0.6402	0.6147	0.6514	0.7035
64-bits	0.3422	0.3449	0.5184	0.6446	0.6214	0.6621	0.7333

Table 6. Comparison of mAP with different hash code lengths on CIFAR10.

Code Length \ Method	LSH	SH	PRH	HSH	KSH	SDH	LHH
8-bits	0.1214	0.1648	0.2185	0.3919	0.3726	0.4367	0.4618
16-bits	0.1256	0.1687	0.2256	0.4031	0.4017	0.4613	0.4845
32-bits	0.1302	0.1724	0.2336	0.4163	0.4225	0.4967	0.5034
64-bits	0.1287	0.1693	0.2312	0.4203	0.4412	0.5111	0.5259

In Tables 3–6, it is clearly observed that the LHH generally achieves the best performance. The SDH and KSH have similar results, where the KSH is better than the SDH on UCMD. Moreover, the SDH outperforms the KSH on the SAT4, SAT6 and CIFAR10 datasets. The HSH has a satisfactory performance on both the SAT4 and SAT6 datasets. For the other two methods, the PRH is generally superior to LSH and SH. Therefore, these results indicate that the LHH can have a promising retrieval performance on these four datasets.

(2) Figure 6 gives the comparison of P-R curves of six hashing methods under different datasets. In Figure 6, the P-R curve of LHH is mostly above than those of the other methods—thus LHH can obtain a larger area under curve (AUC), which is important for evaluating information retrieval. The performances of LSH, SH and PRH are worst, as their AUC areas are the smallest. The AUC

areas of HSH are smaller than those of KSH and SDH, indicating that HSH underperforms KSH and SDH. The above results demonstrate the superiorities of the proposed LHH over the comparisons in large-scale retrieval tasks.

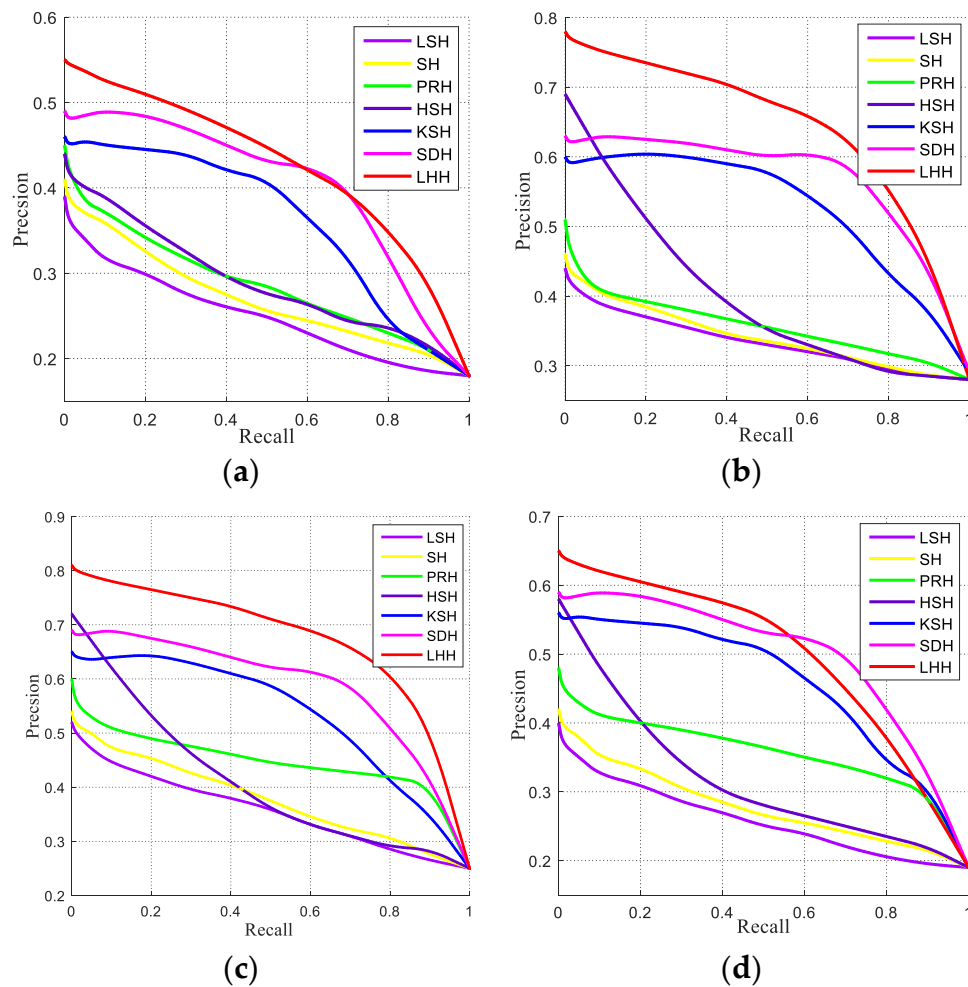


Figure 6. P-R curves of several hashing methods on (a) UCMD; (b) SAT4; (c) SAT6; and (d) CIFAR10.

4.4. Convergence Analysis

This section empirically studies the convergence of LHH. Figure 7 illustrates the convergence curves of LHH on these data sets. From Figure 7, we can clearly see that LHH quickly converges within around eight iterations. The empirical results corroborate Theorem 1.

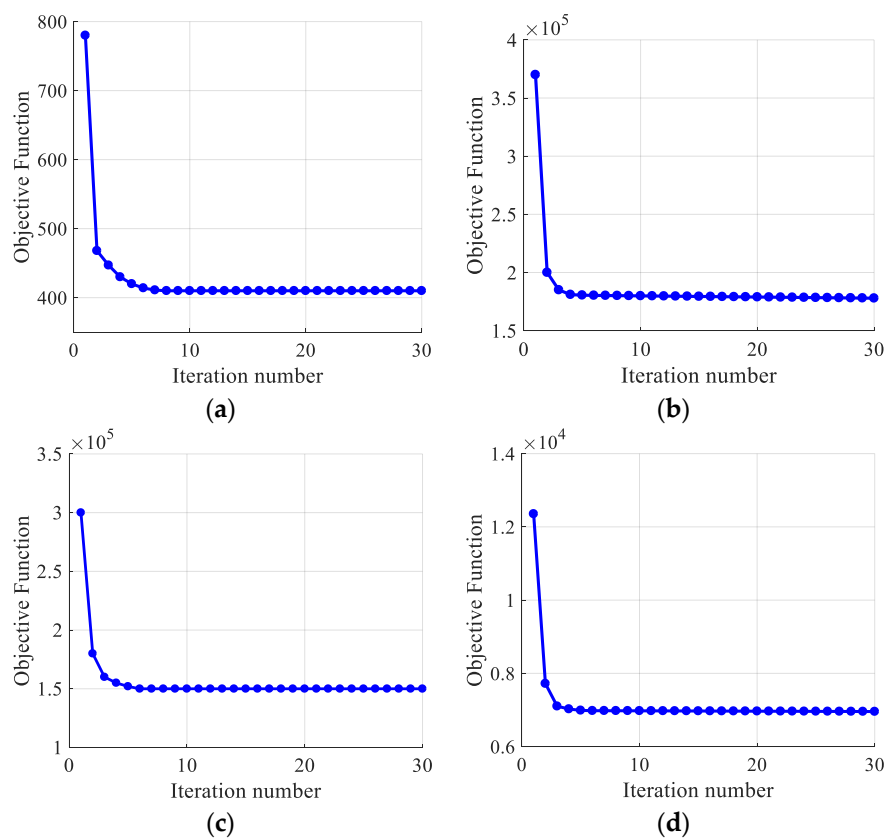


Figure 7. The convergence analysis of LHH: (a) UC Merced Land Use Dataset (UCMD); (b) SAT4; (c) SAT6; and (d) CIFAR10.

4.5. Parameter Analysis

We discuss the sensitivity of the sparse regularization parameter λ_1 and hypergraph regularization parameter λ_2 in the proposed LHH. We show their influences on the mAP with a 32-bit code. In the experiment, λ_1 and λ_2 are varied from the range of $[10^{-4}, 10^{-2}, 10^0, 10^2, 10^4]$. From Figure 8, we see that the mAP slightly changes with the two parameters. As λ_1 and λ_2 increase, mAP slowly rises and then drops on four datasets. The mAP change with λ_2 is larger than that of λ_1 . In general, the LHH can achieve acceptable results on the four datasets when $\lambda_1, \lambda_2 \in [0.01, 1]$. These results demonstrate that sparse and hypergraph terms can help improve the retrieval.

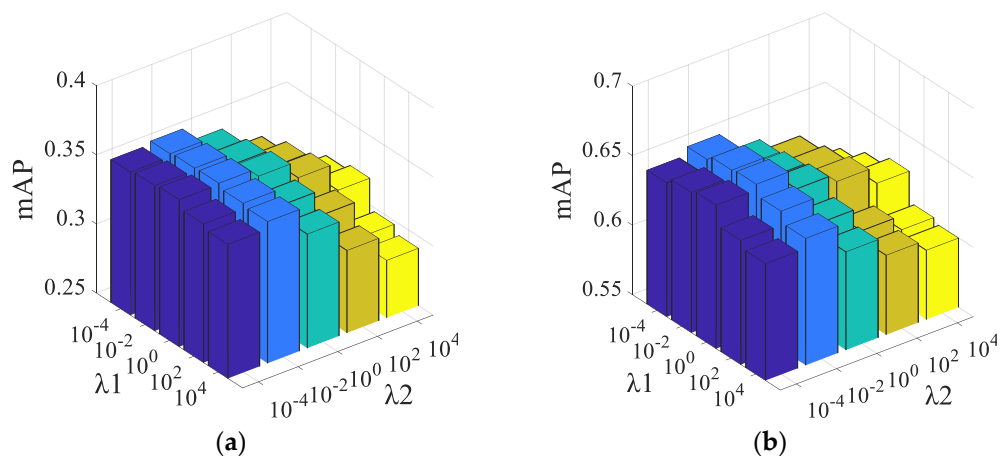


Figure 8. Cont.

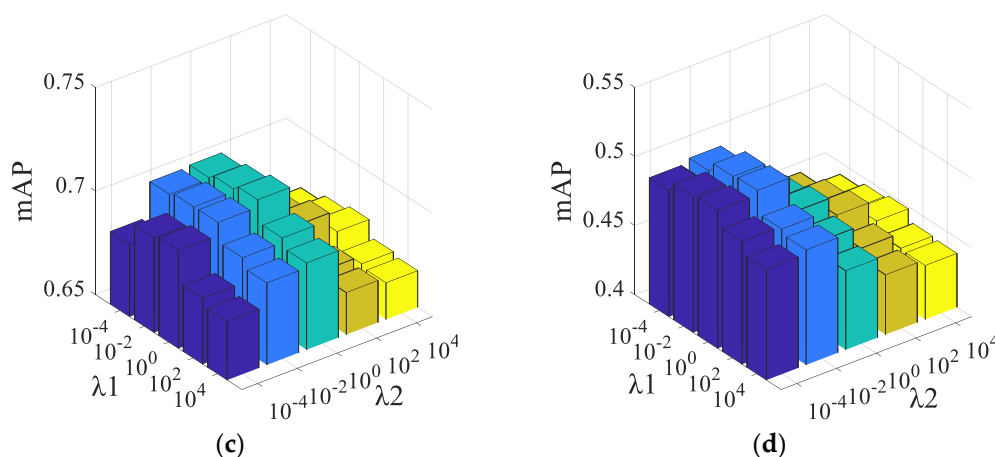


Figure 8. Parameter analysis of λ_1 and λ_2 in LHH on: (a) UCMD; (b) SAT4; (c) SAT6; and (d) CIFAR10.

5. Discussion

The experimental results on four datasets reveal the following interesting points:

- Section 4.3.1 qualitatively shows that the proposed low-rank hypergraph hashing (LHH) has better retrieval performance on large-scale remote sensing (RS) image datasets. Specifically, LHH can retrieve more correct images than the comparison methods, as shown in Figures 4 and 5.
- Section 4.3.2 quantitatively reveals that the proposed LHH is obviously superior than the existing methods on four large-scale datasets, including three remote sensing and one natural image dataset. Specifically, Tables 3–6 illustrate that LHH has a higher mean average precision (mAP) than comparison methods, and Figure 6 illustrates that LHH also has better Precision-Recall (P-R) curves.
- Section 4.4 shows that LHH converges very quickly within eight iterations on several datasets. This indicates that LHH may have less training time in real applications.
- Section 4.5 shows that LHH is relatively robust to these parameters. From Figure 8, LHH generally performs well when $\lambda_1, \lambda_2 \in [0.01, 1]$. It demonstrates the effectiveness of the sparse and hypergraph terms.
- The LHH works very well for efficient large-scale RS image retrieval. It can effectively explore complex structures among RS image datasets and extract more discriminative hash codes.

6. Conclusions

This work focuses on applying a hashing technique for efficient large-scale remote sensing image retrieval (RSIR) tasks. We propose a new low-rank hypergraph hashing (LHH) method to generate compact hash codes on remote sensing (RS) images. LHH constraints low-rankness and sparsity on the transformation matrix to explore its global structure and filter unrelated features. LHH uses hypergraphs to capture the high-order relationship among data, and is very suitable to explore the complex structure of RS images. Extensive experiments are conducted on three RS image datasets and one natural image dataset that are publicly available. The experimental results demonstrate that the proposed LHH outperforms the existing hashing learning in RSIR tasks. In the future, we will explore the deep learning extension of LHH to further improve the performance of large-scale RS image retrieval.

Author Contributions: All the authors contributed to this study; conceptualization, J.K.; methodology, J.K.; software, J.K.; writing, J.K.; writing—review and editing, Q.S., M.M. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Science Foundation of China under Grant 61673220.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Arias, L.; Cifuentes, J.; Marín, M.; Castillo, F.; Garcés, H. Hyperspectral imaging retrieval using MODIS satellite sensors applied to volcanic ash clouds monitoring. *Remote Sens.* **2019**, *11*, 1393. [[CrossRef](#)]
2. Lloret, J.; Bosch, I.; Sendra, S.; Serrano, A. A wireless sensor network for vineyard monitoring that uses image processing. *Sensors* **2011**, *11*, 6165–6196. [[CrossRef](#)] [[PubMed](#)]
3. Tang, X.; Zhang, X.; Liu, F.; Jiao, L. Unsupervised deep feature learning for remote sensing image retrieval. *Remote Sens.* **2018**, *10*, 1243. [[CrossRef](#)]
4. Feng, Q.; Wei, Y.; Yi, Y.; Hao, Q.; Dai, J. Local ternary cross structure pattern: A color LBP feature extraction with applications in CBIR. *Appl. Sci.* **2019**, *9*, 2211. [[CrossRef](#)]
5. Hou, Y.; Wang, Q.J. Research and improvement of content-based image retrieval framework. *Int. J. Pattern Recogn. Artif. Intell.* **2018**, *32*, 1850043. [[CrossRef](#)]
6. Rashno, A.; Sadri, S. Content-based image retrieval with color and texture features in neutrosophic domain. In Proceedings of the 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), Shahrekord, Iran, 19–20 April 2017.
7. Zhuang, L.; Lin, C.H.; Figueiredo, M.A.T.; Bioucas-Dias, J.M. Regularization parameter selection in minimum volume hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9858–9877. [[CrossRef](#)]
8. Chen, C.; Gong, W.; Chen, Y.; Li, W. Object detection in remote sensing images based on a scene-contextual feature pyramid network. *Remote Sens.* **2019**, *11*, 339. [[CrossRef](#)]
9. Zhang, T.; Yan, W.; Su, C.; Ji, S. Accurate object retrieval for high-resolution remote-sensing imagery using high-order topic consistency potentials. *Int. J. Remote Sens.* **2015**, *36*, 4250–4273. [[CrossRef](#)]
10. Li, P.; Zhang, X.; Zhu, X.; Ren, P. Online hashing for scalable remote sensing image retrieval. *Remote Sens.* **2018**, *10*, 709. [[CrossRef](#)]
11. Sajjad, M.; Haq, I.U.; Lloret, J.; Ding, W.; Muhammad, K. Robust Image Hashing Based Efficient Authentication for Smart Industrial Environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6541–6550. [[CrossRef](#)]
12. Shen, F.M.; Shen, C.H.; Liu, W.; Shen, H.T. Supervised discrete hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–13 June 2015; pp. 37–45.
13. Zhou, Y.; Liu, C.; Li, N.; Li, M. A novel locality-sensitive hashing algorithm for similarity searches on large-scale hyperspectral data. *Remote Sens. Lett.* **2016**, *7*, 965–974. [[CrossRef](#)]
14. Li, P.; Ren, P. Partial randomness hashing for large-scale remote sensing image retrieval. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 464–468. [[CrossRef](#)]
15. Demir, B.; Bruzzone, L. Hashing-based scalable remote sensing image search and retrieval in large archives. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 892–904. [[CrossRef](#)]
16. Liu, H.; Wang, R.; Shan, S.; Chen, X. Deep supervised hashing for fast image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
17. Li, Y.; Zhang, Y.; Huang, X.; Zhu, H.; Ma, J. Large-scale remote sensing image retrieval by deep hashing neural networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 950–965. [[CrossRef](#)]
18. Fan, L.L.; Zhao, H.W.; Zhao, H.Y. Distribution consistency loss for large-scale remote sensing image retrieval. *Remote Sens.* **2020**, *12*, 175. [[CrossRef](#)]
19. Welsh, D.J.A. Graphs and hypergraphs. *Bull. London Math. Soc.* **1974**, *6*, 218–220. [[CrossRef](#)]
20. Jian, P.; Chen, K.; Zhang, C. A hypergraph-based context-sensitive representation technique for VHR remote-sensing image change detection. *Int. J. Remote Sens.* **2016**, *37*, 1814–1825. [[CrossRef](#)]
21. Sun, L.; Ji, S.; Ye, J. Hypergraph spectral learning for multi-label classification. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 668–676.
22. Huang, Y.; Liu, Q.; Zhang, S.; Metaxas, D.N. Image retrieval via probabilistic hypergraph ranking. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3376–3383.
23. Liu, Y.; Shao, J.; Xiao, J.; Wu, F.; Zhuang, Y. Hypergraph spectral hashing for image retrieval with heterogeneous social contexts. *Neurocomputing* **2013**, *119*, 49–58. [[CrossRef](#)]
24. Ding, K.; Meng, F.; Liu, Y.; Xu, N.; Chen, W. Perceptual hashing based forensics scheme for the integrity authentication of high resolution remote sensing image. *Information* **2018**, *9*, 229. [[CrossRef](#)]

25. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. In Proceedings of the 25th International VLDB Conference, Edinburgh, UK, 7–10 September 1999; pp. 518–529.
26. Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. In Proceedings of the Advances in Neural Information Processing Systems 21 (NIPS 2008), Vancouver, BC, Canada, 8–11 December 2008.
27. Liu, W.; Wang, J.; Ji, R.R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
28. Du, B.; Huang, Z.; Wang, N.; Zhang, Y.; Jia, X. Joint weighted nuclear norm and total variation regularization for hyperspectral image denoising. *Int. J. Remote Sens.* **2018**, *39*, 334–355. [[CrossRef](#)]
29. Deng, Y.J.; Li, H.C.; Fu, K.; Du, Q.; Emery, W.J. Tensor low-rank discriminant embedding for hyperspectral image dimensionality reduction. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7183–7194. [[CrossRef](#)]
30. Cai, X.; Ding, C.; Nie, F.; Huang, H. On the equivalent of low-rank regressions and linear discriminant analysis based regressions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1124–1132.
31. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184. [[CrossRef](#)] [[PubMed](#)]
32. Liu, G.; Yan, S. Latent low-rank representation for subspace segmentation and feature extraction. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1615–1622.
33. Yang, H.; Yin, J.; Jiang, M. Perceptual image hashing using latent low-rank representation and uniform LBP. *Appl. Sci.* **2018**, *8*, 317. [[CrossRef](#)]
34. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006; pp. 43–55, 511–522.
35. Huang, Y.C.; Liu, Q.S.; Lv, F.J.; Gong, Y.H.; Metaxas, D.N. Unsupervised image categorization by hypergraph partition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1266–1273. [[CrossRef](#)]
36. Bu, J.J.; Tan, S.L.; Chen, C.; Wang, C.; Wu, H.; Zhang, L.J.; He, X.F. Music recommendation by unified hypergraph: Combining social media information and music content. In Proceedings of the 18th ACM International Conference on Multimedia, Florence, Italy, 25–29 October 2010; pp. 391–400.
37. Zhu, X.; Xie, Q.; Zhu, Y.; Liu, X.; Zhang, S. Multi-view multi-sparsity kernel reconstruction for multi-class image classification. *Neurocomputing* **2015**, *169*, 43–49. [[CrossRef](#)]
38. Cheng, X.; Zhu, Y.; Song, J.; Wen, G.; He, W. A novel low-rank hypergraph feature selection for multi-view classification. *Neurocomputing* **2017**, *253*, 115–121. [[CrossRef](#)]
39. Zhu, X.; Suk, H.I.; Lee, S.W.; Shen, D. Subspace regularized sparse multitask learning for multiclass neurodegenerative disease Identification. *IEEE Trans. Biomed. Eng.* **2016**, *63*, 607–618. [[CrossRef](#)]
40. Wen, Z.; Yin, W. A feasible method for optimization with orthogonality constraints. *Math. Program.* **2013**, *142*, 397–434. [[CrossRef](#)]
41. Shen, X.B.; Shen, F.M.; Sun, Q.S.; Yang, Y.; Yuan, Y.H.; Shen, H.T. Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view Retrieval. *IEEE Trans. Cybern.* **2016**, *47*, 4275–4288. [[CrossRef](#)]
42. Liu, W.; Mu, C.; Kumar, S.; Chang, S.F. Discrete graph hashing. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 13 December 2014; pp. 3419–3427.
43. Shen, X.B.; Shen, F.M.; Liu, L.; Yuan, Y.H.; Liu, W.; Sun, Q.S. Multiview discrete hashing for scalable multimedia search. *ACM Trans. Intell. Syst. Technol.* **2018**, *9*, 53. [[CrossRef](#)]
44. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
45. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; DiBiano, R.; Karki, M.; Nemani, R. DeepSat - A learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3 November 2015.
46. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Department University of Toronto, Toronto, ON, Canada, 2009.

47. Shao, Z.; Yang, K.; Zhou, W. Performance Evaluation of Single-Label and Multi-Label Remote Sensing Image Retrieval Using a Dense Labeling Dataset. *Remote Sens.* **2018**, *10*, 964. [[CrossRef](#)]
48. Ye, F.; Luo, W.; Dong, M.; He, H.; Min, W. SAR Image retrieval based on unsupervised domain adaptation and clustering. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1482–1486. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).