

## Research Article

# Online Discrete Anchor Graph Hashing for Mobile Person Re-Identification

Liang Xie and Xi Fang 

School of Science, Wuhan University of Technology, Wuhan 430070, China

Correspondence should be addressed to Xi Fang; [fangxi@whut.edu.cn](mailto:fangxi@whut.edu.cn)

Received 30 April 2021; Accepted 5 November 2021; Published 21 December 2021

Academic Editor: Alessandro Severino

Copyright © 2021 Liang Xie and Xi Fang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advance of mobile technologies, mobile devices such as unmanned aerial vehicle (UAV) become more important in video surveillance. By applying mobile person re-identification (re-id), mobile devices can monitor pedestrians in the transportation system from complex environments. Since the computing and storage resources of mobile devices are limited, traditional person re-id methods are not appropriate for mobile condition. Besides, mobile person re-id task also requires real-time processing. In this paper, we propose a novel hashing method: online discrete anchor graph hashing (ODAGH) for mobile person re-id. ODAGH integrates the advantages of online learning and hashing technology. In ODAGH, we propose an online discrete optimization algorithm to improve the efficiency of anchor graph learning in the online scenario. Experimental results demonstrate the superiority of ODAGH in terms of both effect and efficiency.

## 1. Introduction

With person re-identification (re-id) technology, it will be able to find the same identity from different and non-overlapping cameras. Person re-id can be widely used for video surveillance; moreover, person re-identification is the key technology in pedestrian traffic monitoring [1]. Detecting and tracking a person across camera is important in traffic monitoring system [2]. Also, person re-id technology can intelligently and efficiently identify and track pedestrians in streets, airports, or other transportation systems.

The task of person re-id is an image retrieval problem. Given a probe image (query), the purpose of person re-id is to search the information about established personnel characteristics in a traffic database for images that contain the same person [3]. Traditional person re-id technology is used in the scene where cameras are unable to move, such as fixed camera networks in different public areas, including urban transport systems.

In recent years, with the advance of mobile technologies, mobile systems such as unmanned aerial vehicle (UAV) are

widely used for video surveillance and traffic monitoring [2]. For example, UAV with cameras can play more important roles in tracking people in transportation system from complex environments such as rural area, mountain, and sea, where fixed cameras are lacking or cannot reach, and they can accomplish dangerous and boring visual information gathering tasks with great excellence. Therefore, mobile person re-id is required. Unlike traditional person re-id methods, mobile systems have limited computing and storages resources, and in real-word applications, the communication bandwidth is also limited. However, image processing methods usually have high time and space complexity, and traditional person re-id methods are not suitable for mobile systems.

The hashing technique is an efficient image retrieval method for fast person re-id. It converts high-dimensional data into short binary code while keeping the data similar. With the operations including XOR and bit-counting, it will make it easier to do a fast search. Also, the hashing method based on machine learning is proved to be superior to the hashing method based on random projection [4–6]. However, most existing hashing methods use batch learning

strategy, and they learn hash functions offline by using specific training data. If new person images are collected and added to the database, they also use the pre-trained model to obtain new hashing codes. If the information of new images is different to training data, the performance of offline hashing model will be affected significantly. Offline hashing methods must accumulate all the databases to retrain new hash functions and recompute all the hashing codes. They are obviously very inefficient, especially when the database is frequently updated by new collected images.

Existing hashing methods are not appropriate for the application of mobile system which requires fast real-time methods to solve their tasks such as person re-id. To cope with the real-time demand of mobile systems, a mobile person re-id method should have two important characteristics: (1) it should support fast retrieval of images and (2) it should learn hashing functions from changing training data in online manner. As an emerging technology, online hashing technique [7, 8] can be applied to cope with the online retrieval of changing database.

In the paper, online discrete anchor graph hashing (ODAGH) is proposed for mobile person re-id. ODAGH utilized the advantages of graph learning to effectively preserve the visual correlation of person images. However, traditional graph learning still requires much storage and computing resources. In ODAGH, we propose online anchor graph learning which first uses anchor graph to reduce the space cost of graph construction and then uses an online learning algorithm to optimize the graph model effectively and efficiently. The main contributions of this paper are summarized as follows:

- (i) ODAGH integrates the advantages of online learning and hashing, and as a hashing method, it can be easily applied to mobile systems with limited computing and storage resources. By leveraging online discrete algorithm, ODAGH can efficiently update hashing functions and learn discrete hashing codes when new images are collected. It can fulfil the real-time demand of mobile systems.
- (ii) ODAGH is an unsupervised hashing method, so it requires no training labels which are hard to obtain. By improving the graph learning, the visual correlation can be effectively preserved in hashing codes. Moreover, by using a GPI-based online optimization, quantization loss can be largely avoided in the hashing process.
- (iii) ODAGH only relies on one parameter, which guarantees its robustness in real applications. Also, experimental results demonstrate the effectiveness and efficiency of ODAGH compared to other person re-id methods.

The rest of this paper is organized as follows. The related work is reviewed in Section 2. We present the details of our proposed method in Section 3. The experimental configuration and results are introduced in detail in Section 4. Finally, the conclusion of the paper is given in Section 5.

## 2. Related Work

*2.1. Image Hashing.* In recent years, image hashing has gained much attention from researchers in image retrieval. Generally, there are two kinds of hashing types: one is supervised hashing [9–11] and the other is unsupervised hashing [6, 12]. By learning the class labels or leveraging other supervised information, the binary hash coding can be studied in a supervised hashing method. Representative supervised hashing methods include column sampling-based discrete supervised hashing (COSDISH) [9], supervised discrete hashing (SDH) [10], deep supervised discrete hashing (DSDH) [13], and fast scalable supervised hashing (FSSH) [14]. However, because of the extremely expensive costs to annotated supervised labels, the supervised hashing methods have limited application range on large-scale image retrieval.

Unsupervised hashing methods have one obvious advantage that they can learn hash functions without any label information. Therefore, the original geometric structure, including visual information, can be retained in hash coding. Representative supervised hashing methods include spectral hashing (SH) [4], iterative quantization (ITQ) [6], angular reconstructive embeddings (AREs) [15], unsupervised hashing with binary deep neural network (UH-BDNN) [16], and similarity-adaptive deep hashing (SADH) [17]. Nevertheless, most unsupervised and supervised hashing methods learn hash functions in offline manner, and they cannot cope with the scenario where new data are continuously added to the database.

### 2.2. Pedestrian Detection and Tracking

*2.2.1. Person Re-Identification.* Traditional research on person re-id mainly includes visual feature representation [18, 19] and distance metric learning [20, 21]. In [22], feature effectiveness was identified in a query-adaptive manner for feature fusion. The method proposed in [23] learns discriminative and robust representations via dictionary learning. In [24], Fisher vectors were used for person re-id. Part loss network was proposed in [25] for deep representation learning. There are also many studies related to distance metric learning. In [26], person re-identification problem was formulated as a ranking problem, and Ensemble RankSVM was developed to solve the problem. KISSME [27] considers the scalability and learns a distance metric from equivalence constraints. Mahalanobis metric learning was also used in several methods [28, 29]. Feature presentation learning can be regarded as the pre-processing step of person re-id, and it can be combined with metric learning or our proposed hashing method.

Recently, several hashing-based person re-id methods [30–33] were proposed to improve the search efficiency. As we know, fast indexing is so essential between the raw image data and the binary hashing codes. In order to realize the end-to-end fast indexing, the convolutional neural network (CNN) is always adopted in deep regularized similarity comparison hashing (DRSCH) [34]. The spatial information

is integrated with part-based deep hashing (PDH) by representing horizontal parts to binary codes for feature discrimination [35]. A novel coarse-to-fine (CTF) method [36] complementarily learns short and long codes in a pyramid structure from CNN. Most of existing hashing-based person re-id methods use deep network for feature representation learning, and they ignore the preservation of visual correlation in hashing codes. Moreover, deep learning [37, 38] methods cost many computing and storage of resources, so they are not appropriate for the mobile person re-id task which is always performed on mobile systems with limited resources.

### 3. Proposed Method

In this section, we introduce online discrete anchor graph hashing (ODAGH) in detail. The overall framework of ODAGH is shown in Figure 1. If a mobile device (e.g., UAV) collects new images of a person, then visual features are extracted for online discrete anchor graph learning. ODAGH can efficiently update hashing functions by only new features and training variables with small sizes. Then hashing codes of new images are added to database. In the query step, users can use a query image of specific person to search images of this person from database.

**3.1. Problem Setting.** Suppose the database consists of streaming images. When new images come in, we update the hash functions. We define  $\mathbf{X} \in \mathbb{R}^{N \times d}$  as image matrix, where  $N$  is the number of all training images in database and  $d$  is the dimension of image feature. In the online learning process, image matrix  $\mathbf{X}$  can be represented as  $[\mathbf{X}_{\text{old}}^T, \mathbf{X}_{\text{new}}^T]^T$ , where  $\mathbf{X}_{\text{old}} \in \mathbb{R}^{N_{\text{old}} \times d}$  denotes old images in the database and  $\mathbf{X}_{\text{new}} \in \mathbb{R}^{N_{\text{new}} \times d}$  denotes new images,  $N = N_{\text{new}} + N_{\text{old}}$ . Our goal is to learn hashing functions and hashing codes  $\mathbf{H} \in \mathbb{R}^{N \times k}$  for all images, where  $k$  is the code length. In order to guarantee efficiency, we directly use linear projection to reduce time cost. Thus, the formulation of hashing function is defined as

$$\mathbf{H} = \text{sgn}(\mathbf{X}\mathbf{W}), \quad (1)$$

where  $\mathbf{W} \in \mathbb{R}^{d \times k}$  is the weight matrix. The main notations used in this paper are summarized in Table 1.

**3.2. Basic Formulation.** We use graph learning [39] to preserve the visual information in hashing codes. However, traditional graph learning costs many computing and storage resources, and it is hard to design an online algorithm for graph learning. Therefore, we use anchor graph learning which constructs anchor graph to approximate a graph.

For constructing the anchor graph,  $N_a$  anchors  $x_j^a|_{j=1}^{N_a}$  firstly can be randomly selected from the training data, and later we can approximate the data neighbourhood structure. Then, the truncated similarity matrix  $\mathbf{Z} \in \mathbb{R}^{N \times N_a}$  can be achieved as

$$Z_{ij} = e^{-\text{dist}(x_i - x_j^a)/\sigma}, \quad (2)$$

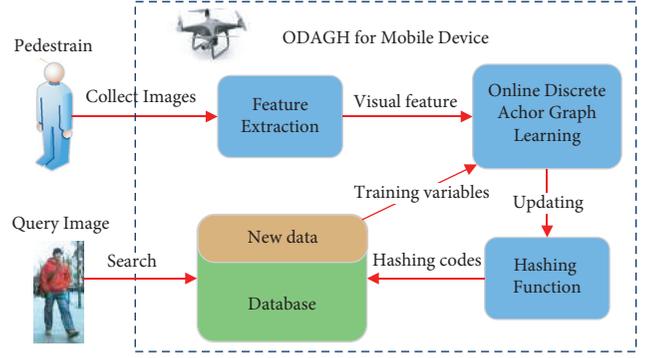


FIGURE 1: The overall framework of ODAGH for mobile person re-id.

TABLE 1: Summary of main notations.

Notations	Explanations
$N$	Size of whole database
$d$	Feature dimension
$k$	Hashing code length
$\mathbf{X}$	Image feature matrix
$\mathbf{H}$	Hashing code matrix
$\mathbf{Z}$	Truncated similarity matrix
$\mathbf{W}$	Weight matrix for hashing
$\mathbf{X}_{\text{new}}$	Feature matrix of new images
$\mathbf{X}_{\text{old}}$	Feature matrix of old images

where  $\text{dist}(x_i - x_j^a)$  is the distance between image  $x_i$  and anchor  $x_j^a$  (in this paper, we use L2 distance for similarity matrix) and  $\sigma$  is the mean of all distances. In addition, we can approximate the graph matrix of each modality by  $\mathbf{A} = \mathbf{Z}\mathbf{Z}^T$ .

Based on the construction of anchor graph, we can formulate anchor graph learning for hashing as

$$\begin{cases} \min & \text{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) \\ \text{s.t.} & \mathbf{H} \in \{0, 1\}^{n \times l}. \end{cases} \quad (3)$$

The above formulation is similar to traditional graph learning. The biggest difference is that the Laplacian matrix should be computed as  $\mathbf{L} = \mathbf{I} - \mathbf{A}$ , and  $\mathbf{A}$  is approximated graph matrix.  $\text{Tr}(\cdot)$  denotes trace operator.

It is unavailable to directly solve the discrete constraint of hashing matrix  $\mathbf{H}$  in equation (3). Therefore, we relax the hard discrete constraint by introducing a continuous matrix  $\mathbf{F} = \mathbf{X}\mathbf{W}$  to replace  $\mathbf{H}$  in equation (3). We also add the term  $\|\mathbf{F} - \mathbf{H}\|_F^2$  to make  $\mathbf{H}$  close to  $\mathbf{F}$ . Then, we can obtain the overall basic formulation as

$$\begin{cases} \min & \text{Tr}(\mathbf{W}^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{W}) + \alpha \|\mathbf{X}\mathbf{W} - \mathbf{H}\|_F^2 \\ \text{s.t.} & \mathbf{W}^T \mathbf{W} = \mathbf{I}, \mathbf{H} \in \{0, 1\}^{n \times l}, \end{cases} \quad (4)$$

where  $\alpha$  is the parameter for the second term. The constraint  $\mathbf{W}^T \mathbf{W} = \mathbf{I}$  is used to avoid the trivial solution of  $\mathbf{W}$ .

The objective function (4) can be solved by an alternate iteration method. At each step, we optimize one variable and fix other variables. When we fix  $\mathbf{H}$  and optimize  $\mathbf{W}$ , the objective function can become

$$\begin{cases} \max & Tr(\mathbf{W}^T \mathbf{B} \mathbf{W}) + 2\alpha Tr(\mathbf{W}^T \mathbf{X}^T \mathbf{H}) \\ \text{s.t.} & \mathbf{W}^T \mathbf{W} = \mathbf{I}, \end{cases} \quad (5)$$

where

$$\mathbf{B} = \mathbf{X}^T \mathbf{Z} \mathbf{Z}^T \mathbf{X} - \mathbf{X}^T \mathbf{X}. \quad (6)$$

It is obvious that equation (5) is a quadratic problem on the Stiefel manifold [40], and the Lagrangian function of this objective function is

$$\begin{aligned} L(\mathbf{W}, \Lambda) = & Tr(\mathbf{W}^T \mathbf{B} \mathbf{W}) + 2\alpha Tr(\mathbf{W}^T \mathbf{X}^T \mathbf{H}) \\ & - Tr((\mathbf{W}^T \mathbf{W} - \mathbf{I}) \Lambda). \end{aligned} \quad (7)$$

By setting the derivative of equation (5) w.r.t  $\mathbf{W}$  to 0, we have

$$\frac{\partial L(\mathbf{W}, \Lambda)}{\partial \mathbf{W}} = 2\mathbf{B} \mathbf{W} + 2\alpha \mathbf{X}^T \mathbf{H} - 2\mathbf{W} \Lambda. \quad (8)$$

We can use generalized power iteration (GPI) [40] to solve  $\mathbf{W}$ , and the detailed algorithm of GPI is shown in Algorithm 1.

Then, we fix  $\mathbf{W}$  and optimize  $\mathbf{H}$ , and (4) becomes

$$\begin{cases} \min & \|\mathbf{X} \mathbf{W} - \mathbf{H}\|_F^2 \\ \text{s.t.} & \mathbf{H} \in \{0, 1\}^{n \times l}. \end{cases} \quad (9)$$

The solution of (4) can be easily obtained by

$$\mathbf{H} = \text{sgn}(\mathbf{X} \mathbf{W}). \quad (10)$$

**3.3. Online Algorithm.** In this section, we consider the online optimization of (4). In the online learning process, image matrix  $\mathbf{X}$  can be represented as  $[\mathbf{X}_{\text{old}}^T, \mathbf{X}_{\text{new}}^T]^T$ . Similarly, hashing matrix  $\mathbf{H}$  can be represented as  $[\mathbf{H}_{\text{old}}^T, \mathbf{H}_{\text{new}}^T]^T$ , where  $\mathbf{H}_{\text{old}}$  denotes the hashing codes of old images and  $\mathbf{H}_{\text{new}}$  denotes the hashing codes of new images.

Then, we consider the online improvement of GPI, and we can obtain that

$$\mathbf{C}_{\text{new}} = \mathbf{X}^T \mathbf{X} = \mathbf{C}_{\text{old}} + \mathbf{X}_{\text{new}}^T \mathbf{X}_{\text{new}}, \quad (11)$$

where  $\mathbf{C}_{\text{old}} = \mathbf{X}_{\text{old}}^T \mathbf{X}_{\text{old}}$ , and we can find that  $\mathbf{C}_{\text{old}}$  is only related to old images; it can be preserved in previous learning. So, in the online process, we only need to compute  $\mathbf{X}_{\text{new}}^T \mathbf{X}_{\text{new}}$ , whose time complexity is  $O(N_{\text{new}})$ . The time of computing  $\mathbf{C}_{\text{new}}$  is linear to the size of new images, and it is irrelevant to size of the database.

The truncated similarity matrix  $\mathbf{Z}$  also can be represented as  $[\mathbf{Z}_{\text{old}}^T, \mathbf{Z}_{\text{new}}^T]^T$ . Therefore, we can obtain that

$$\mathbf{D}_{\text{new}} = \mathbf{Z}^T \mathbf{X} = \mathbf{D}_{\text{old}} + \mathbf{Z}_{\text{new}}^T \mathbf{X}_{\text{new}}, \quad (12)$$

where  $\mathbf{D}_{\text{old}} = \mathbf{Z}_{\text{old}}^T \mathbf{X}_{\text{old}}$ . Similar to (8), we can find that the time complexity of computing  $\mathbf{D}$  is also linear to the size of new images, and it is irrelevant to the size of the whole database.

Based on (11) and (12), we can obtain online updation of  $\mathbf{B}_{\text{new}}$  with linear time complexity which is irrelevant to the size of the database:

$$\mathbf{B}_{\text{new}} = \mathbf{D}_{\text{new}}^T \mathbf{D}_{\text{new}} - \mathbf{C}_{\text{new}}. \quad (13)$$

Similarly, when we compute  $\mathbf{M}$ , its term  $\mathbf{X}^T \mathbf{H}$  can be computed in an online manner:

$$\mathbf{E}_{\text{new}} = \mathbf{X}^T \mathbf{H} = \mathbf{E}_{\text{old}} + \mathbf{X}_{\text{new}}^T \mathbf{H}_{\text{new}}, \quad (14)$$

where  $\mathbf{E}_{\text{old}} = \mathbf{X}_{\text{old}}^T \mathbf{H}_{\text{old}}$ , and the time complexity of computing  $\mathbf{E}_{\text{new}}$  is linear to the size of new data.

According to the above division for online updating, we propose an online optimization algorithm based on GPI to solve  $\mathbf{W}$  and optimize  $\mathbf{H}$ . The detailed procedure is summarized in Algorithm 2.

In Algorithm 2, according to above analysis, we can find that the time complexities of steps 1–5 and steps 7–8 are all  $O(N_{\text{new}})$ , and  $N_{\text{new}}$  is the size of new images. The time complexity of steps 9 and 11 is  $O(d^2 k)$ ,  $d$  is the dimension of image features, and  $k$  is the code length. Since the size of  $\mathbf{M}$  is  $d \times k$ , performing SVD on  $\mathbf{M}$  costs  $O(d^2)$ .  $d$  and  $k$  are irrelevant to the new data size  $N_{\text{new}}$ , and thus they can be ignored in the computation of whole time complexity. As a result, the overall time complexity of Algorithm 2 is  $O(N_{\text{new}})$ . The time complexities of the steps in Algorithm 1 are all irrelevant to the size of the database. Thus, even when the database is very large, our algorithm is constantly efficient.

**3.4. Overall Process.** Suppose the initial database has  $N_I$  images  $\mathbf{X}_I$ , and it becomes larger when new images come in. The overall process of our OAGH is described in Algorithm 3.

Suppose that the database size is  $N$  currently. We can easily find that the time complexity of Algorithm 1 is  $O(N_I)$ , and the time complexity of Algorithm 2 is  $O(N_{\text{new}})$ . Therefore, the time cost of our overall iterative online process is linear to the size of all images. Also, it is equivalent to the time cost of one-round learning of offline hashing.

## 4. Experiments

**4.1. Datasets.** We use Market-1501 dataset [41] to evaluate the performance of our method. There are 32,668 bounding boxes of 1,501 identities in Market-1501 dataset. It has the largest person re-id dataset with 14.8 cross-camera ground truths for each query on average. Also, it is closer towards realistic situations than previous ones. Market-1501 dataset can better show the effectiveness of online methods. Since we mainly focus on the efficiency of person image retrieval and do not focus on feature learning, we directly use IDE\_R-esNet\_50 features.

**4.2. Experimental Settings.** We compared our methods with two non-hashing methods: Euclidean [41] and KISSME [27], and two supervised offline hashing methods: COSDISH [9] and SDH [10]. We set  $N_I = 1000$ , which means that the

```

(i) Input:  $\mathbf{B}, \mathbf{X}, \mathbf{H}$ 
(ii) Output:  $\mathbf{W}$ 
(1) Initialize an orthogonal matrix  $\mathbf{W}$ ;
(2) for iter < max_iter do
(3) Update  $\mathbf{H}$  according to equation (10);
(4) Compute  $\mathbf{M} = 2\mathbf{B}\mathbf{W} + 2\alpha\mathbf{X}^T\mathbf{H}$ ;
(4) Perform the compact SVD  $\mathbf{M} = \mathbf{U}\mathbf{S}^V\mathbf{T}$ ;
(6) Compute  $\mathbf{W} = \sqrt{n}\mathbf{U}^V\mathbf{T}$ ;
(7) end for

```

ALGORITHM 1: Generalized power iteration for solving  $\mathbf{W}$  and  $\mathbf{H}$ .

```

(i) Input:  $\mathbf{C}_{old}, \mathbf{D}_{old}, \mathbf{E}_{old}, \mathbf{H}_{old}, \mathbf{X}_{old}, \mathbf{X}_{old}$ 
(ii) Output:  $\mathbf{W}, \mathbf{C}_{old}, \mathbf{D}_{old}, \mathbf{E}_{old}, \mathbf{H}$ 
(1) Compute truncated similarity matrix  $\mathbf{Z}_{new}$ ;
(2) Compute  $\mathbf{C}_{new}$  according to (11);
(3) Compute  $\mathbf{D}_{new}$  according to (12);
(4) Compute  $\mathbf{B}_{new}$  according to (13);
(5) Initial an orthogonal matrix  $\mathbf{W}$ ;
(6) for iter < max_iter do
(7) Compute  $\mathbf{H}_{new} = \text{sgn}(\mathbf{X}_{new}\mathbf{W})$ ;
(8) Update  $\mathbf{E}_{new}$  according to (14);
(9) Compute  $\mathbf{M} = 2\mathbf{B}_{new}\mathbf{W} + 2\alpha\mathbf{E}_{new}$ ;
(10) Perform the compact SVD  $\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ ;
(11) Compute  $\mathbf{W} = \sqrt{n}\mathbf{U}\mathbf{V}^T$ ;
(12) end for
(13) Update  $\mathbf{C}_{old} = \mathbf{C}_{new}, \mathbf{D}_{old} = \mathbf{D}_{new}, \mathbf{E}_{old} = \mathbf{E}_{new}$ ;
(14) Add new hashing codes by  $\mathbf{H} = [\mathbf{H}_{old}^T, \mathbf{H}_{new}^T]^T$ 

```

ALGORITHM 2: Online optimization algorithm based on GPI.

```

(i) Input:  $\mathbf{X}_I, \mathbf{X}_{new}$ 
(ii) Output:  $\mathbf{W}, \mathbf{H}$ 
(1) Using  $\mathbf{X}_I$  as training data, compute  $\mathbf{W}_{old}, \mathbf{H}_{old}$  according to Algorithm 1;
(2) While new images  $\mathbf{X}_{new}$  come in
(3) Update  $\mathbf{W}, \mathbf{C}_{old}, \mathbf{D}_{old}, \mathbf{E}_{old}, \mathbf{H}$  by Algorithm 2;
(5) End While

```

ALGORITHM 3: Overall process of ODAGH.

database contains 1000 images at the beginning. Then, we set  $N_{new} = 1000$ , which means that at each round, 1000 new images are added to the database. Finally, the database contains 32668 images; at the last round, 668 images are added. Since COSDISH and SDH are offline methods and it is obviously time-consuming to train in every round, to be consistent with real applications, we only train hash functions at first round for COSDISH and SDH. Deep learning-based hashing methods [34–36] usually require many computing and storage resources, which are not appropriate for mobile embedded systems. Therefore, we do not use them for comparison.

Our method only relies on one parameter  $\alpha$ . We select a proper value of  $\alpha$  from a candidate set  $\{0.1, 1, 10, 10^2, 10^3, 10^4, 5 \times 10^4, 10^5\}$ , and we finally choose

the best  $\alpha = 5 \times 10^4$ . The hashing code length for all hashing methods is set to 512.

Non-interpolated mean average precision (MAP) score is utilized [42] to evaluate the performance of all compared methods. Given a query, the average precision (AP) is defined as

$$\text{AP} = \frac{1}{p} \sum_{i=1}^N \text{pre}(i) \text{rel}(i), \quad (15)$$

where  $p$  is the number of relevant images,  $\text{pre}(i)$  is the precision of top  $i$  retrieved images, and  $\text{rel}(i) = 1$  if the image is relevant to  $i$ -th query; otherwise,  $\text{rel}(i) = 0$ . The MAP score is the mean of AP scores from all queries. Besides

TABLE 2: Comparison of MAP results on Market-1501 dataset.

Method	MAP
Euclidean	0.5422
KISSME	0.5677
COSDISH	0.3829
SDH	0.2989
OAGH	<b>0.5723</b>

The best result is highlighted in bold.

MAP, precision-recall (PR) curves are also used to measure the performance of all methods.

*4.3. Experimental Results.* Table 2 shows the MAP score of all compared methods on Market-1501 dataset. The results of Euclidean and KISSME are reported in [41], and we carefully tune the parameters of COSDISH and SDH to report best results. From Table 2, we can find that OAGH performs best. Although OAGH is an unsupervised method, it can effectively preserve visual correlation of images in hashing codes. The performance of two offline hashing methods is much worse than other methods, even though they are supervised methods. The main reason is that they cannot support online learning of new images, and they only use 1000 initial images for training. The MAP scores obtained by the non-hashing method KISSME are very close to OAGH. The main reason is that hashing will introduce quantization loss. It is reasonable that hashing methods perform even worse than non-hashing methods.

Figure 2 shows the PR curves of all compared methods, and we can further observe the retrieval performance of them. We can obtain similar results as MAP scores. OAGH and two non-hashing methods significantly outperform other two hashing methods. Also, the yellow curve of KISSME is very close to blue curve of OAGH.

Both MAP scores and PR curves demonstrate the superiority of our ODAGH. It can easily outperform offline hashing methods. Also, it can even outperform classical non-hashing methods. The above results prove that the proposed online discrete anchor graph learning process has good applicability and makes the process effective.

At last, we also evaluate the total training time of ODAGH. Table 3 shows the comparison of training time, and the experiment described in Table 3 is conducted on a PC with Core i5 2.11 GHz CPU and 16 GB memory. For offline hashing, we use SDH as representative. When new images come in, offline hashing trains the whole database, while ODAGH only needs to process new data. As a result, in our experiment, ODAGH totally processes 32668 images, and SDH processes 560668 images in fact. Therefore, the training time of ODAGH is significantly less than SDH. If new data continue to come in and the database continues to be enlarged, the superiority of our method in total training time will be much more significant.

*4.4. Influence of Code Length.* Traditional graph-based hashing methods cannot directly solve the discrete constraint of hashing codes, which will introduce much quantization loss. When code length increases, the

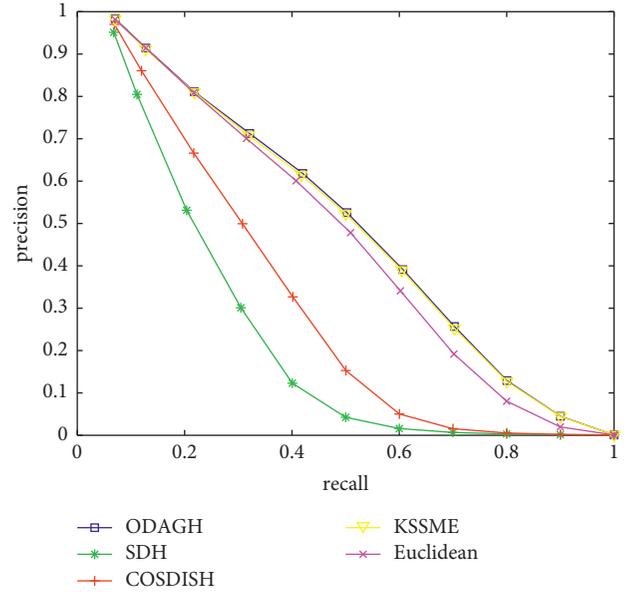


FIGURE 2: PR curves of all compared methods on Market-1501 dataset.

TABLE 3: Comparison of total training time.

Method	Time (seconds)
Offline hashing (SDH)	1964.8
ODAGH	<b>141.4</b>

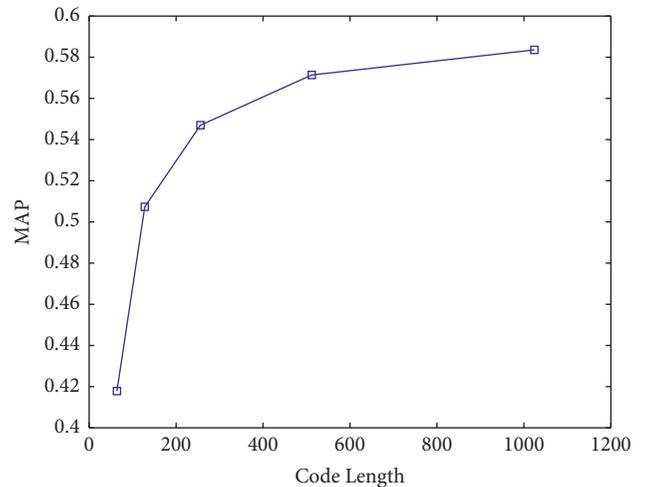


FIGURE 3: MAP variations with code length of 64, 128, 256, 512, and 1024.

quantization loss will also increase, and thus their performance cannot be improved and even deteriorates when the code length increases [43].

Figure 3 shows the performance of ODAGH with increasing code length, and we can observe that the performance increases constantly with the increase of code length. The main reason is that ODAGH uses the GPI-based online discrete optimization procedure which can effectively avoid the quantization loss. We can also find that when that code

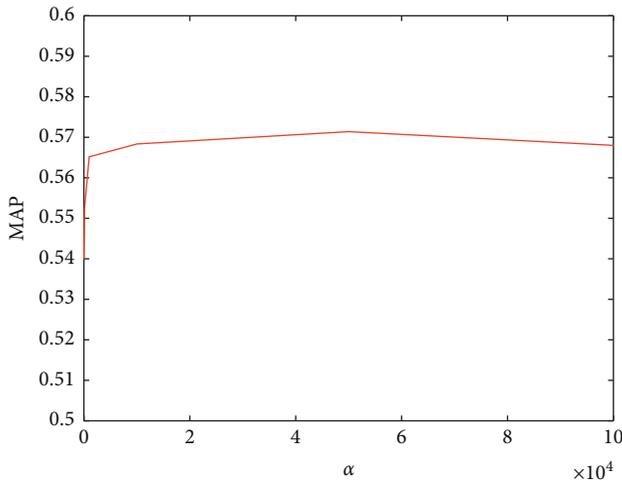


FIGURE 4: MAP variations with different values of  $\alpha$ .

length is larger than 512, the increase of MAP score is not significant, which indicates that 512 bits are sufficient for this person re-id task.

**4.5. Parametric Analysis.** In this section, we analyse the influence of parameter on our method. Figure 4 shows the MAP scores with different values of  $\alpha$ , where  $\alpha$  varies from 1 to  $10^5$ . We can observe that the performance of ODAGH is stable with the variation of  $\alpha$ , and thus our method is not sensitive to the parameter and is robust in person re-id. Moreover, we can find that MAP score achieves relatively high value when  $\alpha > 10^4$ , which has the same magnitude with image number. This phenomenon indicates that we can easily set the parameter  $\alpha$  according to database size.

## 5. Conclusions

In this paper, we propose online discrete anchor graph hashing (ODAGH) for mobile person re-id. We first introduce the basic formulation of discrete anchor graph learning which can effectively preserve visual correlation of images and avoid quantization loss. Then, we derive a novel online optimization algorithm to update hashing functions and compute discrete hashing codes in online manner. The time complexity of online optimization algorithm is linear to the size of new images and irrelevant to the database size. Experimental results on real-world dataset Market-1501 demonstrate both effectiveness and efficiency of ODAGH compared to several state-of-the-art non-hashing and offline hashing person re-id methods.

In future work, we will continue to improve ODAGH to make it more suitable for the real-world application of mobile person re-id. For example, several UAVs usually work together with each other, and distributed hashing [44] is required for the person re-id of UAV swam.

## Data Availability

The Market-1501 data used in this study are from previously reported study which has been cited, and they are publicly

available on <https://github.com/zhunzhong07/IDE-baseline-Market-1501>. The experimental result data used to support the findings of this study are available from the first author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61702388), Equipment Pre-Research Fund (JZX7Y20190253036101), Equipment Pre-Research Ministry of Education Joint Fund (6141A02033703), and Hubei Natural Science Foundation (2019CFC897).

## References

- [1] J. Liu, W. Jing, and M. Liu, "UAV monitoring and forecasting model in intelligent traffic oriented applications," *Computer Communications*, vol. 153, pp. 499–506, 2020.
- [2] G. Kataev, V. Varkentin, and K. Nikolskaia, "Method to estimate pedestrian traffic using convolutional neural network," *Transportation Research Procedia*, vol. 50, pp. 234–241, 2020.
- [3] X. Liu, Y. Mu, D. Zhang, B. Lang, and Y. Li, "Large-scale unsupervised hashing with shared structure learning," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1811–1822, 2014.
- [4] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," *Advances in Neural Information Processing Systems*, vol. 282, no. 3, pp. 1753–1760, 2009.
- [5] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2074–2081, Providence, RI, USA, June 2012.
- [6] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.
- [7] L.-K. Huang, Q. Yang, and W.-S. Zheng, "Online hashing," *IEEE Trans Neural Network Learning System*, vol. 29, no. 6, pp. 2309–2322, 2017.
- [8] C. Leng, J. Wu, J. Cheng, X. Bai, and H. Lu, "Online sketching hashing," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition IEEE Computer Society*, pp. 2503–2511, Boston, MA, USA, June 2015.
- [9] W. Kang, W. Li, and Z. Zhou, "Column sampling based discrete supervised hashing," in *Proceedings of the AAAI Conference Artificial Intelligence (AAAI)*, pp. 1230–1236, Phoenix, AZ, USA, February 2016.
- [10] X. Shi, F. Xing, J. Cai, Z. Zhang, Y. Xie, and L. Yang, "Kernel-based supervised discrete hashing for image retrieval," in *Proceedings of the European Conference on Computer Vision Springer International Publishing, Amsterdam, The Netherlands, October 2016*.
- [11] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE*, Boston, MA, USA, June 2015.
- [12] Q. Jiang and W. Li, "Scalable graph hashing with feature transformation," in *Proceedings of the Joint Conference*

- Artificial Intelligence (IJCAI)*, pp. 2248–2254, Buenos Aires, Argentina, July 2015.
- [13] J. Gui, “Deep supervised discrete hashing,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, p. 1, 2017.
- [14] X. Luo, L. Nie, X. He, Y. Wu, Z. Chen, and X. Xu, “Fast scalable supervised hashing,” in *Proceedings of the 41st International ACM SIGIR Conference Research Development Information Retrieval (SIGIR)*, pp. 735–744, Ann Arbor, MI, U.S.A., July 2018.
- [15] M. Hu, Y. Yang, F. Shen, N. Xie, and H. T. Shen, “Hashing with angular reconstructive embeddings,” *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 545–555, 2018.
- [16] T.-T. Do, A.-D. Doan, and N.-M. Cheung, “Learning to hash with binary deep neural network,” in *Proceedings of the Computer Vision - ECCV 2016*, pp. 219–234, Amsterdam, The Netherlands, October 2016.
- [17] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, “Unsupervised deep hashing with similarity-adaptive and discrete optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.
- [18] H. P. Zhu, “Massive-scale image retrieval based on deep visual feature representation,” *Journal of Visual Communication and Image Representation*, vol. 70, Article ID 102738, 2020.
- [19] W. He, X. Zhu, D. Cheng, R. Hu, and S. Zhang, “Unsupervised feature selection for visual classification via feature-representation property,” *Neurocomputing*, vol. 236, pp. 5–13, 2017.
- [20] X. Deng, K. Y. Liao, Y. L. Zheng, and H. Lei, “A deep multi-feature distance metric learning method for pedestrian re-identification,” *Multimedia Tools and Applications*, vol. 80, pp. 1–19, 2021.
- [21] Y. Ruan, Y. Xiao, Z. Hao, and B. Liu, “A nearest-neighbor search model for distance metric learning,” *Information Sciences*, vol. 552, pp. 261–277, 2021.
- [22] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, “Query-adaptive late fusion for image search and person re-identification,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1741–1750, Boston, MA, USA, June 2015.
- [23] S. Li, M. Shao, and Y. Fu, “Cross-view projective dictionary learning for person re-identification,” in *Proceedings of the 24th International Conference on Artificial Intelligence IJCAI’15*, pp. 2155–2161, Buenos Aires, Argentina, July 2015.
- [24] B. Ma, Y. Su, and F. Jurie, “Local descriptors encoded by fisher vectors for person re-identification,” in *Proceedings of the 12th International Conference on Computer Vision - Volume Part I ECCV’12*, pp. 413–422, Florence, Italy, October 2012.
- [25] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, “Deep representation learning with Part Loss for person Re-identification,” *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2860–2871, 2019.
- [26] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, “Person Re-identification by support vector ranking,” in *Proceedings of the British Machine Vision Conference*, pp. 1–11, Aberystwyth, UK, August 2010.
- [27] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2288–2295, Providence, RI, USA, June 2012.
- [28] M. Hirzer, P. M. Roth, and H. Bischof, “Person Re-identification by efficient impostor-based metric learning,” in *Proceedings of the 2012 IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance*, pp. 203–208, Beijing, China, September 2012.
- [29] P. M. Roth, M. Hirzer, M. Köstinger, C. Belezni, and H. Bischof, “Mahalanobis distance learning for person Re-identification,” *Person Re-Identification*, Springer, Berlin, Germany, pp. 247–267, 2014.
- [30] J. Chen, Y. Wang, J. Qin, L. Liu, and L. Shao, “Fast person Re-identification via cross-camera semantic binary transformation,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5330–5339, Honolulu, HI, USA, July 2017.
- [31] Z. Liu, J. Qin, A. Li, and L. Van Gool, “Adversarial binary coding for efficient person Re-identification,” in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 700–705, Shanghai, China, July 2019.
- [32] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, “Structured deep hashing with convolutional neural networks for fast person Re-identification,” *Computer Vision and Image Understanding*, vol. 167, pp. 63–73, 2017.
- [33] F. Zheng and L. Shao, “Learning cross-view binary identities for fast person Re-identification,” in *Proceedings of the 25th International Joint Conference on Artificial Intelligence IJCAI’16*, pp. 2399–2406, New York, Ny, USA, July 2016.
- [34] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, “Bit-Scalable deep hashing with regularized similarity learning for image retrieval and person Re-identification,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.
- [35] F. Zhu, X. Kong, L. Zheng, H. Fu, and Q. Tian, “Part-based deep hashing for large-scale person Re-identification,” *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4806–4817, 2017.
- [36] G. a. Wang, S. Gong, J. Cheng, and Z. Hou, “Faster person Re-identification,” in *Proceedings of the Computer Vision - ECCV 2020*, pp. 275–292, Glasgow, UK, August 2020.
- [37] A. Brunetti, D. Buongiorno, G. Francesco Trotta, G. F. Trotta, and V. Bevilacqua, “Computer vision and deep learning techniques for pedestrian detection and tracking: a survey,” *Neurocomputing*, vol. 300, pp. 17–33, 2018.
- [38] M. Zhong, C. Li, L. Liu, J. Wen, J. Ma, and X. Yu, “Fuzzy n learning for deep 3-D segmentation of point cloud,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3181–3192, 2020.
- [39] L. Liu, F. Nie, A. Wiliem, Z. Li, T. Zhang, and B. C. Lovell, “Multi-modal Joint clustering with application for unsupervised attribute discovery,” *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4345–4356, 2018.
- [40] F. Nie, R. Zhang, and X. Li, “A generalized power iteration method for solving quadratic problem on the Stiefel manifold,” *Science China*, vol. 60, no. 11, pp. 146–155, 2017.
- [41] L. Zheng, L. Shen, L. Tian, S. Wang, T. Wang, and Q. Tian, “Scalable person Re-identification: a benchmark,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) IEEE*, Santiago, Chile, December 2015.
- [42] L. Xie, L. Zhu, P. Pan, and Y. Lu, “Cross-Modal Self-Taught Hashing for large-scale image retrieval,” *Signal Processing*, vol. 124, pp. 81–92, 2016.
- [43] W. Liu, C. Mu, and S. Kumar, “Discrete graph hashing,” *Advances in Neural Information Processing Systems*, vol. 4, pp. 3419–3427, 2014.
- [44] S. Wang, C. Li, and H.-L. Shen, “Distributed graph hashing,” *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1896–1908, 2019.