

Fast Manifold Ranking With Local Bipartite Graph

Xiaojun Chen^{1b}, *Member, IEEE*, Yuzhong Ye, Qingyao Wu^{2b}, *Senior Member, IEEE*,
and Feiping Nie^{1b}, *Senior Member, IEEE*

Abstract—During the past decades, manifold ranking has been widely applied to content-based image retrieval and shown excellent performance. However, manifold ranking is computationally expensive in both graph construction and ranking learning. Much effort has been devoted to improve its performance by introducing approximating techniques. In this paper, we propose a fast manifold ranking method, namely Local Bipartite Manifold Ranking (LBMR). Given a set of images, we first extract multiple regions from each image to form a large image descriptor matrix, and then use the anchor-based strategy to construct a *local bipartite graph* in which a regional k -means (RKM) is proposed to obtain high quality anchors. We propose an iterative method to directly solve the manifold ranking problem from the *local bipartite graph*, which monotonically decreases the objective function value in each iteration until the algorithm converges. Experimental results on several real-world image datasets demonstrate the effectiveness and efficiency of our proposed method.

Index Terms—Clustering, normalized cut, large-scale data.

I. INTRODUCTION

CONTENT-BASED image retrieval (CBIR), which retrieves the most similar instances to a provided query element from a potentially large database, has been a longstanding research topic since the early 1990s. The early studies in the 1990s and early 2000s usually extract global descriptors [1], but such methods may fail to deal with image changes such as illumination, translation, occlusion and truncation. Then some approaches had been proposed to use the SIFT descriptors [2], such as Bag-of-Words (BoW) [3], spatial verification [1], hamming embedding [4] and query expansion [5]. In 2012, Krizhevsky *et al.* achieved the state-of-the-art recognition accuracy in ILSRVC 2012 with

AlexNet [6], exceeding previous best results by a large margin. Recently, research focus has begun to transfer to deep learning based methods [7], [8], [9] especially the convolutional neural network (CNN). The CNN-based retrieval models usually employ the pre-trained CNN models or perform fine-tuning of CNN models to obtain the compact descriptors for retrieval [10].

With the learned descriptors, a popular type of method to perform image retrieval is manifold ranking that learns the ranking scores of the whole images by considering their manifold structure [11]–[14]. Such kind of method assumes that those nearby images belong to the same cluster or manifold and assigns each image a relative ranking score according to the underlying structure, instead of computing absolute pairwise similarities as a traditional way [12].

Given n images, manifold ranking usually computes a $n \times n$ symmetrically normalized affinity matrix \mathbf{S} and learn a ranking vector $\mathbf{f} \in \mathbb{R}^{n \times 1}$ with an iterative procedure $\mathbf{f}_{t+1} = \alpha \mathbf{S} \mathbf{f}_t + (1 - \alpha) \mathbf{y}$ [12], where \mathbf{f}_{t+1} and \mathbf{f}_t denote the learned \mathbf{f} in the $(t+1)$ -th and t -th iterations, respectively. The above solution was proved to be able to converge to the closed-form solution, however, it is still difficult and time-consuming to obtain the converged solution for large-scale data. Much effort has been devoted to improve its performance. For example, some works use approximation methods to reduce the affinity matrix size [13], [15]–[17] or compute approximate scores [18], while the other works employ other optimization methods to solve the ranking problem [14], [17].

In this paper, we propose a fast manifold ranking method, namely Local Bipartite Manifold Ranking (LBMR). The procedure of the new method is shown in Figure 1. Given a set of n images, we first extract t regions from each image to form a large $nt \times p$ image descriptor matrix in which p is the number of features extracted from each region. These image descriptors are partitioned into m clusters and these cluster centers are used as anchors to build a *local bipartite graph* by computing the similarities between the anchors and the image descriptors. Finally, an new iterative method is proposed to learn the ranking scores. The main contributions include:

- 1) We propose a Regional k -means (RKM) for seeking anchors that are eventually distributed within and between different regions, and building a bipartite graph.
- 2) An iterative method is proposed to directly solve the manifold ranking problem with the constructed bipartite graph. Our proposed method has lower time complexity than the existing bipartite graph based method.

Manuscript received June 29, 2019; revised August 13, 2020 and January 22, 2021; accepted January 25, 2021. Date of publication July 15, 2021; date of current version July 27, 2021. This work was supported in part by the Major Project of the New Generation of Artificial Intelligence under Grant 2018AAA0102900, in part by the NSFC under Grant 61773268 and Grant 61876208, in part by the Natural Science Foundation of SZU under Grant 000346, and in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2018B010108002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giulia Boato. (Corresponding authors: Qingyao Wu; Feiping Nie.)

Xiaojun Chen is with the College of Computer Science and Software, Shenzhen University, Shenzhen 518060, China (e-mail: xjchen@szu.edu.cn). Yuzhong Ye and Qingyao Wu are with the School of Software Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: sesbyyz@mail.scut.edu.cn; qyw@scut.edu.cn).

Feiping Nie is with the Center for OPTIMAL, School of Computer Science, Northwestern Polytechnical University, Xi'an, Shanxi 710072, China (e-mail: feipingnie@gmail.com).

Digital Object Identifier 10.1109/TIP.2021.3096082

1941-0042 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

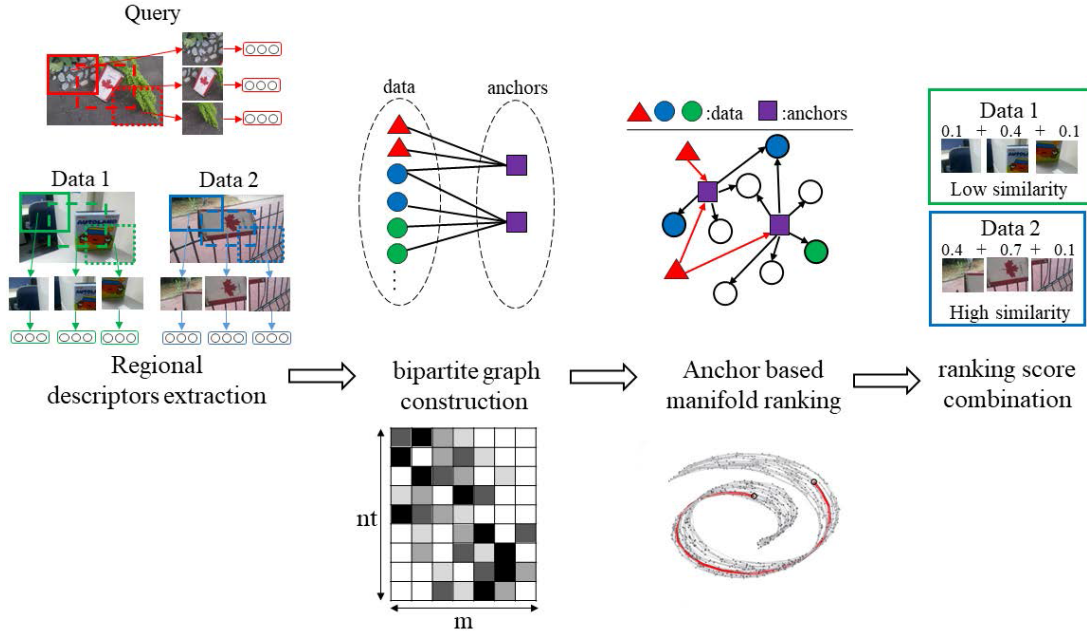


Fig. 1. Procedure of the proposed method.

Moreover, compared with existing iterative manifold ranking methods, the new method is able to monotonically decreases the objective function value in each iteration until the algorithm converges and has lower time complexity. The experimental results show that it converges more quickly and can obtain better results.

- 3) Experimental results on several real-world image datasets demonstrate the effectiveness and efficiency of our proposed method.

The rest of this paper is organized as follows. Section II presents the notations and definitions, and Section III gives a brief review of related work. The new method is proposed in Section IV. The experimental results on both synthetic and real-world datasets are reported in Section V. Conclusions and future work are given in Section VI.

II. NOTATIONS

The notations used in this paper are as follows. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i -th row is denoted as \mathbf{m}^i , and its j -th column is denoted by \mathbf{m}_j . The Frobenius norm of the matrix \mathbf{M} is defined as $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m m_{ij}^2}$. $\sqrt{\mathbf{M}} = (\sqrt{m_{ij}})$. We call a matrix as a cluster indicator matrix if each row in the matrix consists one or more elements equal to 1 to indicate the class membership while the rest elements are 0. We denote the set of all cluster indicator matrices as Ψ .

III. RELATED WORK

Given a large database, retrieving the most similar objects to a provided query object is an important research topic in computer vision. Traditionally, this is done based on analyzing

the pairwise similarities between two objects. Manifold ranking [12], also called as diffusion ranking [19], is a popular ranking method due to its ability to effectively capture the manifold structure in the data points [11].

Given a set of n data points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_q, \mathbf{x}_{q+1}, \dots, \mathbf{x}_n\}$, where the first q points are the query points and the rest are the points that we want to rank according to their relevances to the queries. Let $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ denote a metric on \mathcal{X} , such as Euclidean distance or cosine distance, which assigns each pair of data points \mathbf{x}_i and \mathbf{x}_j a distance $d(\mathbf{x}_i, \mathbf{x}_j)$. Let $\mathbf{f}: \mathcal{X} \rightarrow \mathbb{R}$ denote a ranking function which assigns to each point \mathbf{x}_i a ranking value f_i . $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is a vector, in which $y_i = 1$ if \mathbf{x}_i is a query image and $y_i = 0$ otherwise. If we have prior knowledge about the confidence of an image \mathbf{x}_i , then we can assign probabilistic ranking scores to \mathbf{y}_i as initial value. An affinity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ can be constructed in which a_{ij} denotes the similarity between \mathbf{x}_i and \mathbf{x}_j , and a symmetrically normalized matrix $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ can be obtained where \mathbf{D} is the diagonal matrix in which $d_{ii} = \sum_{j=1}^n a_{ij}$.

The classical manifold ranking is to minimize the following problem [11], [12]

$$\min_{\mathbf{f} \geq 0} \mathbf{f}^T \mathbf{L}_S \mathbf{f} + \mu \|\mathbf{f} - \mathbf{y}\|_2^2 \quad (1)$$

where \mathbf{L}_S is the graph Laplacian which is defined as $\mathbf{L}_S = \text{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S}$ and $\mu > 0$ is a regularization parameter.

It can be verified that problem (1) has a closed-form solution as

$$\mathbf{f} = (1 - \alpha) \mathcal{L}_\alpha^{-1} \mathbf{y} \quad (2)$$

where $\alpha = \frac{1}{1+\mu} \in [0, 1)$ and $\mathcal{L}_\alpha = \mathbf{I}_n - \alpha \mathbf{S}$. Clearly, the scaling factor $1 - \alpha$ does not make contributions for the ranking task so we can rewrite the above closed-form solution as

$$\mathbf{f} = \mathcal{L}_\alpha^{-1} \mathbf{y} \quad (3)$$

However, it is impractical to directly compute the closed-form solution due to the high time cost of the inversion of a large $n \times n$ matrix [19]. Therefore, most work focused on developing iterative solutions. Zhou *et al.* proposed an iterative method to solve problem (1) [12], in which

$$\mathbf{f}_{q+1} = \alpha \mathbf{S} \mathbf{f}_q + (1 - \alpha) \mathbf{y} \quad (4)$$

where \mathbf{f}_{q+1} and \mathbf{f}_q denote the learned \mathbf{f} in the $(q + 1)$ -th and q -th iterations, respectively. Zhou *et al.* further proved that the sequence $\{\mathbf{f}_q\}$ defined in Eq. (4) will converge to the closed-form solution defined in Eq. (2) if the number of iterations goes to infinity.

He *et al.* proposed a fast approximation algorithm to solve problem (1) [15], which partitions the graph represented by \mathbf{A} with spectral clustering and uses a low-rank approximation such as SVD to approximate the graph. However, Fujiwara *et al.* pointed out that “if the graph is not well partitioned by spectral clustering, their approach must hold the $n \times n$ matrix in the worst case. This indicates that the time complexity of FMR is $O(n^3)$ ” [18].

Xu *et al.* proposed an anchor graph approach (Efficient Manifold Ranking) EMR to speedup solving problem (1) [13]. In their method, m anchor points are selected from the data points by using the k -means algorithm and an anchor graph is built to represent each data point on the manifold as a linear combination of weights to nearby k anchor points. They compute a weight matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$ from the following Nadaraya-Watson kernel regression

$$z_{ij} = \frac{K_\lambda(\mathbf{x}_i - \mathbf{u}_j)}{\sum_{l=1}^r K_\lambda(\mathbf{x}_i - \mathbf{u}_l)} \quad (5)$$

where $K_\lambda(\mathbf{x}_i - \mathbf{u}_j)$ is the Epanechnikov quadratic kernel in which a smoothing parameter λ determines the size of the local region in which anchors can affect the target point. Denote $\mathbf{H} = \mathbf{Z} \mathbf{D}^{-\frac{1}{2}}$ where \mathbf{D} is a diagonal matrix in which $d_{ii} = \sum_{j=1}^n \mathbf{z}_i^T \mathbf{z}_j$, they rewrote Eq. (3) as

$$\mathbf{f} = (\mathbf{I}_n - \mathbf{H}^T (\mathbf{H} \mathbf{H}^T - \frac{1}{\alpha} \mathbf{I}_m)^{-1} \mathbf{H}) \mathbf{y} \quad (6)$$

which has a time complexity of $O(npm + m^3)$ and space complexity of $O(mn)$.

Fujiwara *et al.* proposed an approximate method Mogul to further speed up solving of problem (1) [18]. Their method computes the approximate scores by utilizing incomplete Cholesky factorization and prunes unnecessary approximate computations in finding the top- k data points by estimating the upper bounding scores.

Iscen *et al.* proposed a regional diffusion method [14], which first extracts features of t different regions from each of n images. To learn the ranking score for each region, they first rewrote Eq. (2) as

$$\mathcal{L}_\alpha \mathbf{f} = (1 - \alpha) \mathbf{y} \quad (7)$$

and use the conjugate gradient to solve it. Finally, the ranking score of one image is computed with sum pooling or generalized max pooling.

To improve the scalability of manifold ranking, Iscen *et al.* casted it as linear filtering over a graph, and proposed a

two-stage fast spectral ranking method [16]. The new method first uses a randomized algorithm [20] to approximate \mathbf{A} as $\mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \approx \mathbf{A}$ offline, where $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $r \ll n$. Then, it computes the ranking score online as

$$\mathbf{f} = \mathbf{U} (1 - \alpha) (\mathbf{I}_r - \alpha \mathbf{\Lambda})^{-1} \mathbf{U}^T \mathbf{y} \quad (8)$$

In 2019, Iscen *et al.* proposed a hybrid diffusion method [17], which combines both regional diffusion [14] and fast spectral ranking [16].

IV. FAST MANIFOLD RANKING WITH LOCAL BIPARTITE GRAPH

In [13], Xu *et al.* proposed to use the anchor-based strategy to construct a bipartite graph and learn the ranking vector from the bipartite graph. We call this *global bipartite graph* in the rest of the paper. However, global features still fail to deal with image changes or when the object of interest is small. Recently, local CNN features from multiple image regions have been investigated for this purpose, either aggregated [21], [22] or represented as a set [23]. In this section, we propose a fast manifold ranking method, namely Local Bipartite Manifold Ranking (**LBMR**). In the new method, we propose to build a *local bipartite graph* to capture the local structure which can be performed offline. To learn online ranking scores, we propose an iterative method to directly solve the manifold ranking problem (1) from the *local bipartite graph*.

A. Offline Graph Construction

Given a set of n images, we first use R-MAC [22] to extract features of t regions $\{\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(t)}\} \subset \mathbb{R}^{p \times 1}$ from each of the i -th image, where each of $\mathbf{x}_i^{(j)}$ is normalized. Then we can form a normalized matrix $\mathbf{Z} \in \mathbb{R}^{nt \times p}$ as follows

$$\mathbf{Z} = \begin{bmatrix} (\mathbf{x}_1^{(1)})^T \\ \vdots \\ (\mathbf{x}_1^{(t)})^T \\ \vdots \\ (\mathbf{x}_n^{(1)})^T \\ \vdots \\ (\mathbf{x}_n^{(t)})^T \end{bmatrix} \quad (9)$$

Then we propose to partition the nt regions into m disjoint clusters, and then compute a $nt \times m$ affinity matrix by calculating the distances between the anchors and the original descriptors. In [13], Xu *et al.* have tried random selection, k -means and fast k -means [24] for anchor generation. Images in real applications usually show local cluster structures, i.e., some images may exist in different clusters in different regions, so we propose to cluster all nt descriptors into m clusters.

In the anchor-based strategy, the key problem is how to effectively obtain a set of m anchors which can best represent the whole data set such that the constructed bipartite similarity matrix best retains the manifold structure in the original data. There are mainly two types of methods for anchor generation, e.g., random selection and k -means generation. k -means is preferred for anchor generation since clustering

centers have a stronger representation power than random selected data [25], [26]. However, k -means may produce very small or isolated clusters that are improper to be used as anchors. Recently, Chen *et al.* proposed a balanced k -means to obtain evenly distributed clusters for anchor generation [27]. However, since these image descriptors are extracted from t regions, intuitively, we hope that these anchors are eventually distributed within and between different regions. To achieve this goal, we propose a novel regional k -means (RKM) which is formulated as minimizing the factorization error of \mathbf{F} with the following objective function

$$\min_{\mathbf{F} \in \Psi^{nt \times m}, \mathbf{H} \in \mathbb{R}^{p \times m}} \|\mathbf{Z} - \mathbf{H}\mathbf{F}^T\|_F^2 + \gamma \text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F}) \quad (10)$$

where $\mathbf{F} \in \Psi^{nt \times m}$ is a cluster indicator matrix, $\mathbf{H} \in \mathbb{R}^{p \times m}$ is the m cluster centers and γ is a regularization parameter. $\Theta \in \Psi^{nt \times t}$ is a binary descriptor-region matrix in which $\theta_{ij} = 1$ iff the i -th descriptor is extracted from the j -th region. Note that although problem (10) uses Euclidean distance to measure the distance between an image descriptor and a cluster center, we can verify that the Euclidean distance is equivalent to cosine distance since \mathbf{Z} is normalized. It can be verified that RKM will degenerate to Balanced k -means (BKM) [27] if $t = 1$.

In problem (10), $\text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F})$ can be rewritten as

$$\text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F}) = \sum_{s=1}^t \sum_{l=1}^m \left(\sum_{i=1}^{nt} f_{il} \theta_{is} \right)^2 \quad (11)$$

Since $\sum_{l=1}^m f_{il} = 1$ and $\sum_{s=1}^t \theta_{is} = 1$, we have $\sum_{s=1}^t \sum_{l=1}^m \sum_{i=1}^{nt} f_{il} \theta_{is} = \sum_{i=1}^{nt} (\sum_{l=1}^m f_{il}) (\sum_{s=1}^t \theta_{is}) = nt$ which is fixed. According to Theorem 1 in [28], we know that minimizing $\text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F})$ results in the most balanced partition for each region, i.e., the clusters are uniformly distributed within and between different regions. Therefore, with the regularization term $\text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F})$, solving problem (10) will obtain clusters that are suitable for anchors.

We present an iterative approach to optimize problem (10), in which \mathbf{F} and \mathbf{H} are alternatively solved. If \mathbf{F} is fixed, by setting the derivative of problem (10) w.r.t \mathbf{H} to zero, we obtain the optimal solution of \mathbf{H} as

$$\mathbf{H} = \mathbf{Z}\mathbf{F}(\mathbf{F}^T \mathbf{F})^{-1} \quad (12)$$

Then we fix \mathbf{H} to update \mathbf{F} by solving the following problem

$$\min_{\mathbf{F} \in \Psi^{nt \times m}} \|\mathbf{Z} - \mathbf{H}\mathbf{F}^T\|_F^2 + \gamma \text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F}) \quad (13)$$

which can be rewritten as

$$\min_{\mathbf{F} \in \Psi^{nt \times m}} \text{Tr}(\mathbf{F}\mathbf{H}^T \mathbf{H}\mathbf{F}^T) + \gamma \text{Tr}(\mathbf{F}^T \Theta \Theta^T \mathbf{F}) - 2\text{Tr}(\mathbf{Z}^T \mathbf{H}\mathbf{F}^T) \quad (14)$$

It can be verified that $\text{Tr}(\mathbf{F}^T \mathbf{F}) = m$ and

$$\begin{aligned} \text{Tr}(\mathbf{F}\mathbf{H}^T \mathbf{H}\mathbf{F}^T) &= \sum_{i=1}^{nt} \sum_{l=1}^c \sum_{j=1}^p \sum_{s=1}^c f_{il} h_{jl} h_{js} f_{is} \\ &= \text{Tr}(\mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H}) \mathbf{F}^T) \end{aligned} \quad (15)$$

where \circ is the Hadamard product between two matrices.

Algorithm 1 Regional k -means Clustering (RKM) to solve problem (10)

-
- 1: **Input:** the data matrix \mathbf{Z} , descriptor-region matrix Θ and a regularization parameter γ .
 - 2: Initialize the cluster indicator matrix \mathbf{F} and set $\delta = 4$.
 - 3: **repeat**
 - 4: Update \mathbf{H} as $\mathbf{H} = \mathbf{Z}\mathbf{F}(\mathbf{F}^T \mathbf{F})^{-1}$.
 - 5: **repeat**
 - 6: Update $\mathbf{G} = \gamma(\delta \mathbf{I}_{nt} - \Theta \Theta^T) \mathbf{F}$
 - 7: Independently update each row of \mathbf{F} according to Eq. (18) and (19).
 - 8: **until** \mathbf{F} does not change
 - 9: **until** problem (10) converges
 - 10: **Output:** the clustering result \mathbf{F} .
-

Therefore, problem (13) can be rewritten as

$$\begin{aligned} \max_{\mathbf{F} \in \Psi^{nt \times m}} & \gamma \text{Tr}(\mathbf{F}^T (\delta \mathbf{I}_{nt} - \Theta \Theta^T) \mathbf{F}) \\ & + \text{Tr}((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}^T) \end{aligned} \quad (16)$$

where δ is a proper constant such that $(\delta \mathbf{I}_{nt} - \Theta \Theta^T)$ is positive-semi definite. We will discuss how to set proper δ in the latter of this subsection.

In this paper, we propose an iterative method to solve the above problem, in which each iteration consists of the following two steps:

- 1) Update $\mathbf{G} = \gamma (\delta \mathbf{I}_{nt} - \Theta \Theta^T) \mathbf{F}$;
- 2) Solve $\max_{\mathbf{F} \in \Psi^{nt \times m}} \text{Tr}(\mathbf{F}^T \mathbf{G}) + \frac{1}{2} \text{Tr}((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}^T)$.

Note that the problem in step (2) can be rewritten as

$$\max_{\mathbf{F} \in \Psi^{nt \times m}} \text{Tr}((\mathbf{G} + \mathbf{Z}^T \mathbf{H} - \frac{1}{2} \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}^T) \quad (17)$$

Then the optimal solution of f_{il} is

$$f_{il} = \langle l = \arg \max_{l' \in [1, m]} q_{il'} \rangle \quad (18)$$

where $\langle . \rangle$ is 1 if the argument is true or 0 otherwise, and $\mathbf{Q} = [q_{il}]$ is defined as

$$\mathbf{Q} = \mathbf{G} + \mathbf{Z}^T \mathbf{H} - \frac{1}{2} \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H}) \quad (19)$$

1) *Determining δ :* To make $(\delta \mathbf{I}_{nt} - \Theta \Theta^T)$ positive-semi definite, we only need to make the eigenvalues of $(\delta \mathbf{I}_{nt} - \Theta \Theta^T)$ all non-negative. Suppose the SVD of Θ is $\Theta = \mathbf{U}_\Theta \Lambda_\Theta \mathbf{V}_\Theta^T$ in which Λ_Θ is a diagonal matrix with eigenvalues in the diagonal line, it can be verified that $\delta = \max \text{diag}(\Lambda_\Theta \Lambda_\Theta^T)$ will make the eigenvalues of $(\delta \mathbf{I}_{nt} - \Theta \Theta^T)$ all non-negative. However, it is time-consuming to perform eigendecomposition on Θ , we turn to give an approximate value for δ . According to the Gershgorin circle theorem and $\sum_{j=1}^t \theta_{ij} = 1$, we know that each of the i -th eigenvalue of Θ should satisfy $|(\Lambda_\Theta)_{ii} - \theta_{ii}| \leq 1$. Note that $\theta_{ii} \in \{0, 1\}$, we have $(\Lambda_\Theta)_{ii} \leq 2$. Therefore, we can set $\delta = 4$.

The detailed algorithm to solve problem (10), namely the regional k -means (RKM), is summarized in Algorithm 1.

In the new algorithm, the cluster centers \mathbf{H} and cluster indicator matrix \mathbf{F} are iteratively updated until converged. In this algorithm, since $\mathbf{F}^T \mathbf{F}$ is a diagonal matrix, we only need $O(pntm)$ time to compute \mathbf{H} , where t is the number of extracted regions, p is the number of extracted features, n is the number of images and m is the number of anchors. We need $O(ntm)$ time to compute \mathbf{G} . Therefore, the overall computational complexity of RKM is $O(pntm)$ which is the same as the computational complexity of k -means. Note that \mathbf{F} can be solved row independently, we can use parallel technique to speed up the whole process. Moreover, the convergence of Algorithm 1 can be ensured by the following theorem (see Appendix for proof):

Theorem 1: Algorithm 1 monotonically decreases problem (10) in each iteration until the algorithm converges.

With the m cluster centers \mathbf{H} as m anchors, the next step is to compute a bipartite similarity matrix $\mathbf{B} \in \mathbb{R}^{nt \times m}$, in which b_{ij} denotes the similarity between the i -th sample and the j -th anchor. Xu *et al.* used the Nadaraya-Watson kernel regression to compute \mathbf{B} and proposed an empirical formula to determine the parameter λ [24]. In this paper, we proposed to use the parameter-free method in [29] to compute \mathbf{B} as follows

$$b_{ij} = \begin{cases} \frac{d_{i,k+1} - d(\mathbf{z}_i, \mathbf{h}_j)}{kd_{i,k+1} - \sum_{h=1}^k d_{i,h}} & \mathbf{h}_j \in \mathcal{N}_k(\mathbf{z}_i) \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where $d(\mathbf{z}_i, \mathbf{h}_j)$ is the distance between \mathbf{z}_i and \mathbf{h}_j , $\mathcal{N}_k(\mathbf{z}_i)$ consists of the k nearest neighbors of \mathbf{z}_i in \mathbf{H} . $d_{i,k+1}$ is the distance between \mathbf{z}_i and its $(k+1)$ -th nearest neighbor in \mathbf{H} . The classical anchor-based strategy defines a bipartite affinity matrix $\mathbf{P} = \mathbf{B} \Delta^{-1/2}$ where \mathbf{B} is computed according to Eq. (20) and $\Delta \in \mathbb{R}^{m \times m}$ is a diagonal matrix defined as $\Delta = \text{diag}(\mathbf{B}^T \mathbf{1}^T)$ and the approximate full affinity matrix $\mathbf{A} = \mathbf{P} \mathbf{P}^T$. In this paper, according to [29], we know that $\Delta = \mathbf{I}_m$, then we can use $\mathbf{P} = \mathbf{B}$ for the subsequent computation.

B. Online Manifold Ranking

In [13], Xu *et al.* have designed an adjacency matrix and substituted it into the closed-form solution in Eq. (2) to produce a solution with a computational complexity of $O(nm + m^3)$, which is lower than the original closed-form solution in Eq. (3). However, it is still time-consuming to compute the inversion of a $m \times m$ matrix since m may be large in large-scale data. Moreover, it is time-consuming to deal with out-of-sample query since it has to compute the inversion again.

To address the above problem, in this paper, we propose a new manifold ranking method. In the new method, with the bipartite affinity matrix \mathbf{P} , we first construct a weighted bipartite graph $\mathcal{G} = (\mathcal{V}^D, \mathcal{V}^A, \mathbf{C})$ where two sets of disjoint data points \mathcal{V}^D and \mathcal{V}^A are associated with nt image descriptors and m anchors, respectively. \mathbf{C} is defined as

$$\mathbf{C} = \begin{bmatrix} 0_{nt} & \mathbf{P}^T \\ \mathbf{P} & 0_m \end{bmatrix}. \quad (21)$$

With the bipartite matrix \mathbf{C} , the next step is to learn ranking scores. Let $\mathbf{f} \in \mathbb{R}^{nt \times 1}$ denote a set of nt ranking scores for nt

image regions, $\mathbf{g} \in \mathbb{R}^{m \times 1}$ denotes a set of m ranking scores for anchors and \mathcal{Q} denotes the indices of $q \times t$ descriptors extracted from q query objects. We want to learn \mathbf{f} and \mathbf{g} by solving the following problem

$$\min_{\mathbf{f} \geq 0, \mathbf{g} \geq 0} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}^T \mathbf{L}_C \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (22)$$

where $\mathbf{L}_C = \text{diag}(\mathbf{C} \mathbf{1}_{nt+m}) - \mathbf{C}$ is the graph Laplacian and \mathbf{C} is defined in Eq. (21).

To make \mathbf{f} for query images to be close to 1, we define a vector $\mathbf{y} \in \mathbb{R}^{(nt+m) \times 1}$, in which $y_i = 1$ iff $i \in \mathcal{Q}$ and $y_i = 0$ otherwise. If we have prior knowledge about the confidences of queries, we can also relax \mathbf{y} to be continuous values in $[0, 1]$ and assign different initial ranking scores to \mathbf{y} . Then we introduce an additional constraint into problem (22) to form the following objective function

$$\min_{\mathbf{f} \geq 0, \mathbf{g} \geq 0} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}^T \mathbf{L}_C \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} + \left(\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} - \mathbf{y} \right)^T \mathbf{R} \left(\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} - \mathbf{y} \right) \quad (23)$$

where \mathbf{R} is a $(nt+m) \times (nt+m)$ diagonal matrix which consists of $(nt+m)$ regularization parameters, in which r_{ii} is defined as

$$r_{ii} = \begin{cases} \beta & \text{if } i \in \mathcal{Q} \\ \eta & \text{otherwise} \end{cases} \quad (24)$$

where β is a regularization parameter which is used to make \mathbf{f} to be close to 1 for query samples, and η is a regularization parameter which is used to make \mathbf{f} to be close to 0 for non-query samples.

According to the special structure of \mathbf{C} , we know that

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}^T (\mathbf{L})_C \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} = \frac{1}{2} \sum_{i=1}^{nt} \sum_{j=1}^m p_{ij} (f_i - g_j)^2 \quad (25)$$

and

$$\begin{aligned} \left(\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} - \mathbf{y} \right)^T \mathbf{R} \left(\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} - \mathbf{y} \right) &= \frac{1}{2} \sum_{i=1}^{nt} r_{ii} (f_i - y_i)^2 \\ &+ \frac{1}{2} \sum_{j=1}^m r_{j+nt, j+nt} (g_j - y_{j+nt})^2 \end{aligned} \quad (26)$$

Note that $r_{j+nt, j+nt} = \eta$ and $y_{j+nt} = 0$ because queries are outside of anchors, problem (23) can be rewritten as

$$\begin{aligned} \min_{\mathbf{f} \geq 0, \mathbf{g} \geq 0} \frac{1}{2} \sum_{i=1}^{nt} \sum_{j=1}^m p_{ij} (f_i - g_j)^2 \\ + \frac{1}{2} \sum_{i=1}^{nt} r_{ii} (f_i - y_i)^2 + \frac{1}{2} \eta \sum_{j=1}^m g_j^2 \end{aligned} \quad (27)$$

In this paper, we propose to solve problem (27) with the alternating optimization technique. In the following, we show how to alternatively update each of the two variables.

Algorithm 2 Algorithm to solve problem (23): LBMR

-
- 1: **Input:** n points \mathbf{X} , \mathbf{y} .
 - 2: Set $\mathbf{f} = \mathbf{y}_{1,\dots,nt}$ and $\mathbf{g} = \mathbf{y}_{nt+1,\dots,nt+m}$.
 - 3: **repeat**
 - 4: Update \mathbf{f} according to Eq. (30).
 - 5: Update \mathbf{g} according to Eq. (33).
 - 6: **until** problem (23) converges
 - 7: **Output:** \mathbf{f} .
-

1) *Update \mathbf{f} With \mathbf{g} Fixed:* if \mathbf{g} are fixed, \mathbf{f} can be obtained by solving the following problem

$$\min_{\mathbf{f} \geq 0} \frac{1}{2} \sum_{i=1}^{nt} \sum_{j=1}^m p_{ij} (f_i - g_j)^2 + \frac{1}{2} \sum_{i=1}^{nt} r_{ii} (f_i - y_i)^2 \quad (28)$$

Note that problem (28) is independent between different i , so f_i can be obtained by solving the following problem

$$\min_{f_i} \frac{1}{2} \sum_{j=1}^m p_{ij} (f_i - g_j)^2 + \frac{1}{2} r_{ii} (f_i - y_i)^2 \quad (29)$$

which has the following optimal solution

$$f_i = \frac{\sum_{j=1}^m p_{ij} g_j + r_{ii} y_i}{\sum_{j=1}^m p_{ij} + r_{ii}} \quad (30)$$

2) *Update \mathbf{g} With \mathbf{f} Fixed:* If \mathbf{f} are fixed, \mathbf{g} can be obtained by solving the following problem

$$\min_{\mathbf{g} \geq 0} \frac{1}{2} \sum_{i=1}^{nt} \sum_{j=1}^m p_{ij} (f_i - g_j)^2 + \frac{1}{2} \eta \sum_{j=1}^m g_j^2 \quad (31)$$

Note that problem (31) is independent between different j , so g_j can be obtained by solving the following problem

$$\min_{g_j} \frac{1}{2} \sum_{i=1}^{nt} p_{ij} (f_i - g_j)^2 + \frac{1}{2} \eta g_j^2 \quad (32)$$

which has the following optimal solution

$$g_j = \frac{\sum_{i=1}^{nt} p_{ij} f_i}{\sum_{i=1}^{nt} p_{ij} + \eta} \quad (33)$$

3) *Ranking Algorithm:* The algorithm to solve problem (23), named Local Bipartite Manifold Ranking (LBMR), is summarized in Algorithm 2. In this algorithm, \mathbf{f} and \mathbf{g} are iteratively updated until converges. Different from the iterative solution in [12] which is not directly solving the manifold ranking problem, our proposed method directly solve problem (23). Compared with the closed-form solutions in Eqs. (3) and (6), our proposed method is more efficient because it does not involve inversion of matrix. Since \mathbf{f} and \mathbf{g} are alternatively directly solved, each of which is an optimal solution, we know that Algorithm 2 will monotonically decrease problem (23) until converges.

C. Computing the Final Ranking Scores

With the learned $n \times t$ ranking scores in \mathbf{f} , a simple way to obtain the final ranking vector $\hat{\mathbf{f}}$ is to compute $\hat{\mathbf{f}} = \mathbf{E}\mathbf{f}$ where $\mathbf{E} \in \Psi^{n \times nt}$ is a binary matrix in which $e_{ij} = 1$ iff the j -th region is extracted from the i -th image.

1) *Feasible \mathbf{y} for Handling New Queries:* In prior work on manifold ranking, a query point is considered to be contained in the dataset \mathcal{X} [11], [13]. However, in real applications, the query point may be absent in the dataset. Conventional method to handle new queries is to append extra rows and columns for new query features [13], [30]. However, it is time consuming to update the whole affinity matrix and recompute the ranking scores. In this paper, given a set of new query images Ω , we follow the Iscens's method [14] to fix the bipartite similarity matrix \mathbf{P} and only update \mathbf{y} as $\mathbf{y}_i = \sum_{l=1}^t \sum_{\omega \in \Omega} s_k(\mathbf{x}_i^l, \omega)$ where $s_k(\mathbf{x}_i^l, \omega)$ is defined as

$$s_k(\mathbf{x}_i^l, \omega) = \begin{cases} s(\mathbf{x}_i^l, \omega) & \mathbf{x}_i^l \in \mathcal{N}_k(\Omega) \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

where $s(\mathbf{x}_i^l, \omega)$ is the cosine similarity between \mathbf{x}_i^l and ω , and $\mathcal{N}_k(\Omega)$ consists of all k nearest neighbors of $\omega \in \Omega$ in \mathcal{X} .

2) *Anchor Selection for Online Ranking:* Although we have built a bipartite affinity matrix $\mathbf{P} \in \mathbb{R}^{nt \times m}$, in the online ranking stage, it is not necessary use the whole \mathbf{P} since many anchors are far from query images. In this paper, we propose to select r nearest anchors to the query images and then only select the columns of the selected anchors to form $\mathbf{P}' \in \mathbb{R}^{nt \times r}$ to perform online ranking.

3) *Complexity Analysis:* In local bipartite graph constructing, we need $O(pntm)$ time to run RKM to obtain anchors and $O(pntm)$ time to compute the bipartite matrix, where t is the number of extracted regions, p is the number of extracted features, n is the number of images, m is the number of k-means anchors and r is the number of reduced anchors. In Algorithm 2, we need $O(ntr)$ time to update \mathbf{f} and \mathbf{g} if we compute a r -nearest neighbors affinity matrix from \mathbf{P} and use it for Algorithm 2. Table I summarizes the time and space complexities of classical image retrieval methods, which shows the superiority of our method in time cost.

Compared with the time complexity $O(nptm + m^3)$ of EMR in [13], our proposed method has a significant reduction in time cost if the number of anchors m is large.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental results on both synthetic and real-world datasets.

A. Toy Example

We used a two-moon synthetic data with 200 data points to validate our proposed method, which is plotted in Figure 2a. A query point is given in the lower moon, and the task is to rank these points according to their relevances to the query point. Intuitively, the relevant degrees of the points in the lower moon to the query should decrease along the moon shape and the relevant degrees of the points in the upper moon to the query should be very small. In this experiment, we set the number of regions to 1 and directly compute the bipartite affinity matrix \mathbf{P} . We used RKM with $\gamma = 1$ to obtain 20 anchors from the original toy data points and then construct a 2-nearest neighbor bipartite graph as shown in Figure 2b. We set $\beta = 1$ and $\eta = 0.1$ to run LBMR and compared our

TABLE I

TIME AND SPACE COMPLEXITIES OF CLASSICAL IMAGE RETRIEVAL METHODS. n IS THE NUMBER OF IMAGES, t IS THE NUMBER OF EXTRACTED REGIONS, p IS THE DIMENSION OF EXTRACTED FEATURES, k IS THE NUMBER OF NEAREST NEIGHBORS, q IS THE NUMBER OF FOURIER BASIS FOR THE DATABASE DESCRIPTORS, m IS THE NUMBER OF k -MEANS ANCHORS. IF WE USE MR AND EMR FOR REGIONAL DESCRIPTORS, n OR m SHOULD BE REPLACED WITH nt OR mt

Methods	Time complexity		Space complexity	
	Offline	Online	Offline	Online
MR	$O(n^2p)$	$O(n(p+k))$	$O(n(p+k))$	
EMR	$O(nmp + m^3)$	$O(n(p+m))$	$O(mp + n(p+k))$	
Regional diffusion	$O((nt)^2p)$	$O(nt(p+k))$	$O(nt(p+k))$	
FSR	$O((nt)^2p + ntpq(k+q))$	$O(nt(p+q))$	$O(nt(p+q+k))$	$O(nt(p+q))$
Hybrid diffusion	$O((nt)^2p + ntpq(k+q))$	$O(nt(p+q+k))$	$O(nt(p+q+k))$	
LBM (ours)	$O(nTMP + ntm)$	$O(nt(p+k))$	$O(mtp + nt(p+k))$	

proposed method with Manifold Ranking (MR) [12] and the results are reported in Figures 4 and 2d. From Figure 4 we can see that our proposed method produced 99.18% result in terms of the mean average precision (mAP) at the 9-th iteration while MR produced their best result until nearly 20 iterations. This results indicate the superior performance of our proposed method in comparison with MR.

B. Experimental Setup

1) *Data*: In this experiment, we used three datasets to perform performance comparison, i.e., Oxf5k [31], Par6k [32] and INSTRE [33]. We also used the preprocessed versions of the Oxf5k and Par6k datasets [34], i.e., Oxf5k-m, Oxf5k-h, Par6k-m and Par6k-h. For large scale task, a commonly-used approach is to add 100k images from Flickr [31] to Oxf5k and Par6k datasets and form the so called Oxf105k and Par106k datasets. We also used the Oxf1m and Par1m datasets to test the retrieval scale, each of which contains more than 1M distracting images [34]. Table II shows the characteristics of the all datasets. We used the mean average precision (mAP) to evaluate the retrieval performance on all datasets.

2) *Image Representation*: We applied our proposed method to both global and regional image descriptors extracted from the three datasets, which were used in [14], [16] and can be downloaded from <ftp://ftp.irisa.fr/local/texmex/corpus/diffusion/>. For the Oxf1m and Par1m datasets, global descriptors were processed by supervised whitening [35]. After obtaining the CNN feature map extracted by the pre-trained VGG/ResNet,¹ three square sliding windows are used on the feature map to obtain regions. The scales of square windows are equals to 1 times, 2/3, and 1/2 of the short side of the feature map. Any two adjacent regions keep around 40% overlap with the sliding of windows. 20 regions were extracted from each image and the full image was added as a separate region [14], resulting in a total of 21 regions for each image. Each region was feed into the MAC layer to obtain a regional vector and 21 regional vectors were obtain for each image. The global descriptors can be obtained by performing sum pooling over all regional vectors.

¹The pre-trained models are released in <http://cmp.felk.cvut.cz/cnnimageretrieval/>

3) *Implementation Details*: We compared our algorithm with five manifold ranking methods, i.e., Manifold Ranking (MR) [12], Efficient Manifold Ranking (EMR) [13], Regional Diffusion (RD) [14], Fast Spectral Ranking (FSR) [16] and Hybrid Diffusion (HD) [17]. The cosine similarity search (SIM) was used as baseline. We have implemented our proposed method and comparison methods in MATLAB2017a and ran them in a computer with 3.4 GHZ($\times 4$) CPU.

We employed the cosine distance to build the matrix \mathbf{B} according to Eq. (20), in which the number of nearest neighbors k was searched between $\{10, 15, 20\}$ for global descriptors and $\{20, 25, 30\}$ for regional descriptors, for all methods. We set $\gamma = 0.01$ to run RKM, and search hyper-parameters as follows: the number of anchors was set to $m \in [nt/10, nt/3]$ for EMR and our proposed method on all datasets, with $r \in [m/10, m/3]$, $\beta \in [0.01, 1]$ and $\eta \in [0.001, 0.1]$ to run LBM. For the rest parameters in the competitor methods, we used the values reported in the corresponding papers.

C. Offline Anchor Generation and Online Selection

In the anchor-based strategy, anchor generation is very important since the constructed bipartite graph will affect the subsequent ranking performance. To investigate the performance of our proposed method for anchor generation, we selected the global descriptors of the Oxf5k dataset to compare several anchor generation methods, i.e., random selection (RS), k -means, fast k -means (FKM) [24] and our proposed method RKM (RKM will degenerate to BKM in global descriptors task). The results are shown in Table III. From this table, we can observe that RKM (and BKM) achieve the best performance which indicates the superior performance of RKM. We also noticed that RKM was slower than k -means and fast k -means (FKM) since it involves more matrix computation. However, such time cost is acceptable since the anchor generation can be conducted in offline stage.

Finally, we selected five datasets to show the anchor selection results of our proposed method in Table IV. The column “speed up” denotes the ratio of the time cost with r selected anchors to the time cost with the total m generated anchors, and the column “mAP drop” denotes the difference between the mAP with r selected anchors and the mAP with the total m generated anchors. From this table, we can observe that

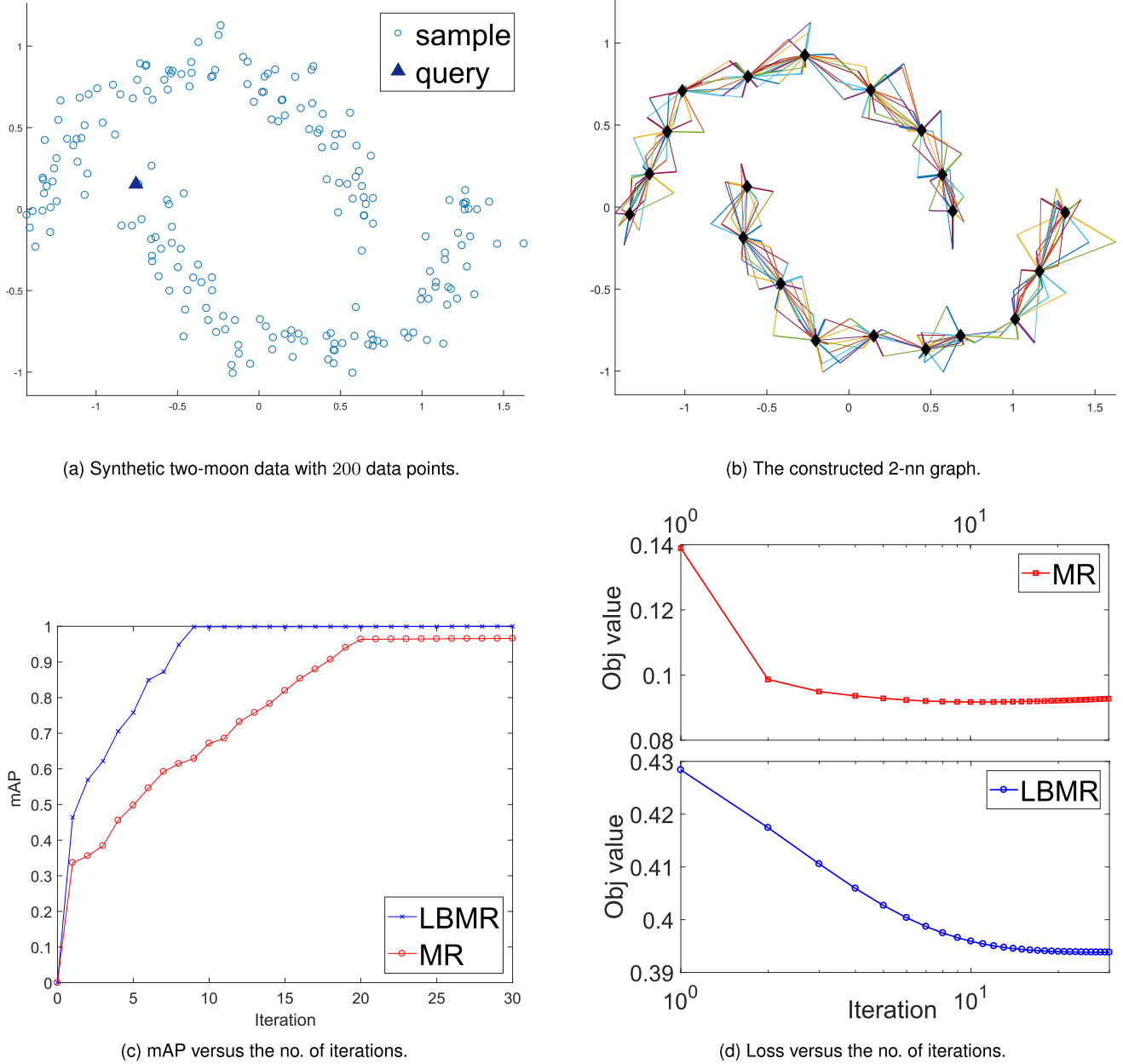


Fig. 2. The toy example on a synthetic two-moons dataset.

anchor selection can accelerate online ranking, though suffers a little performance degradation.

D. Parameter Sensitivity

The retrieval performances (mAP) of our proposed method versus the number of selected anchors r , the number of neighbors k and two regularization parameters β and η of our proposed method, with the global descriptors of three datasets, are shown in Figure 3. Figure 3a shows that the increase of r does not increase the retrieval performance too much on the Oxf5k-m and INSTRE datasets, but greatly increase the retrieval performance on the Par6k-h dataset. From Figure 3b, we can observe that the retrieval performances




increase rapidly first and then slowly drops with the increase of k . In Figures 3c and 3d, with the increase of β and η , the retrieval performances does not change too much on the Oxf5k-m and INSTRE datasets, but drops rapidly on the Par6k-h dataset.

E. Experimental Results with the Global Descriptors

Our proposed method can deal with global descriptor by simply setting the number of regions to 1. In this experiment, we used the global descriptors with both 512 features and 2048 features extracted from all 7 datasets to compare our proposed method with the state-of-the-art. The results are shown in Table V, which indicates that our proposed method

TABLE II

CHARACTERISTICS AND QUERY SAMPLES OF 11 DATASET. GT/QUERY INDICATES THE AVERAGE NUMBER OF GROUND TRUTH IMAGES PER QUERY

Dataset	No. of images	No. of queries	GT/query	Query examples
Oxford5k	5063	55	51	
Oxford5k-m	4993	70	79	
Oxford5k-h	4993	70	35	
Oxf105k	105134	55	51	
Oxf1m	1002128	55	51	
Paris6k	6392	55	162	
Paris6k-m	6322	70	246	
Paris6k-h	6322	70	147	
Par106k	106463	55	162	
Par1m	1003457	55	162	
INSTRE	27293	1250	193	

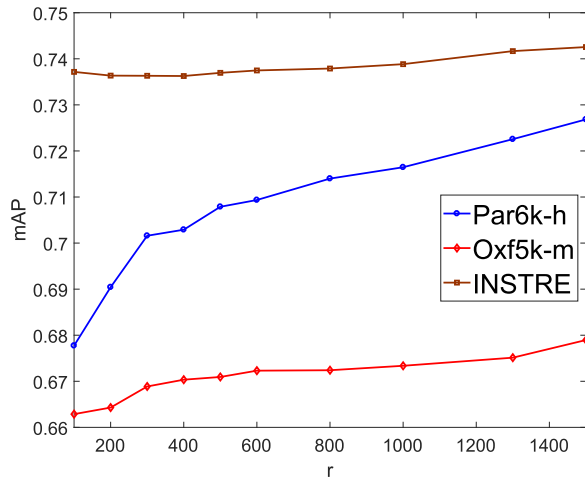
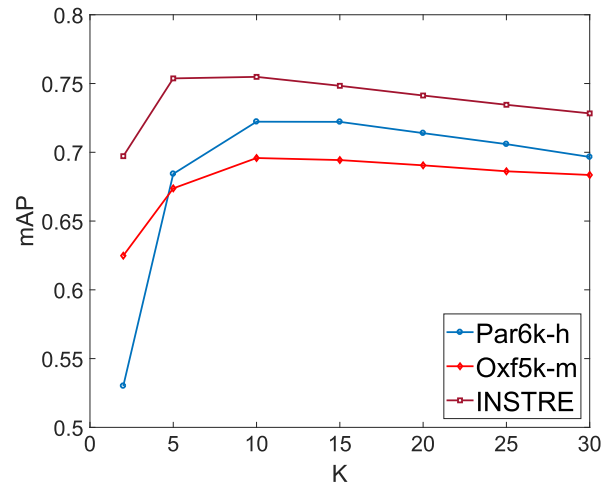
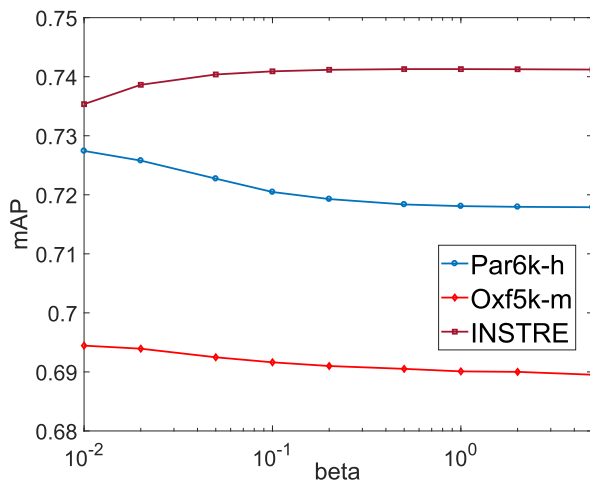
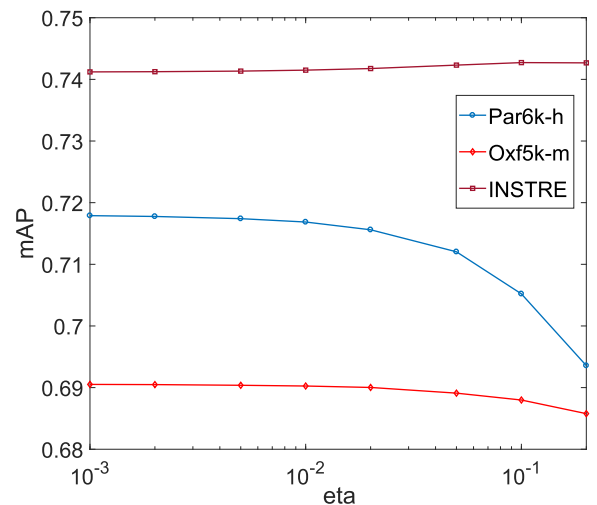
(a) mAP versus the number of selected anchors r .(b) mAP versus the number of neighbours k .(c) mAP versus β .(d) mAP versus η .

Fig. 3. Retrieval performance (mAP) versus four parameters of LBMR with the global descriptors of three datasets.

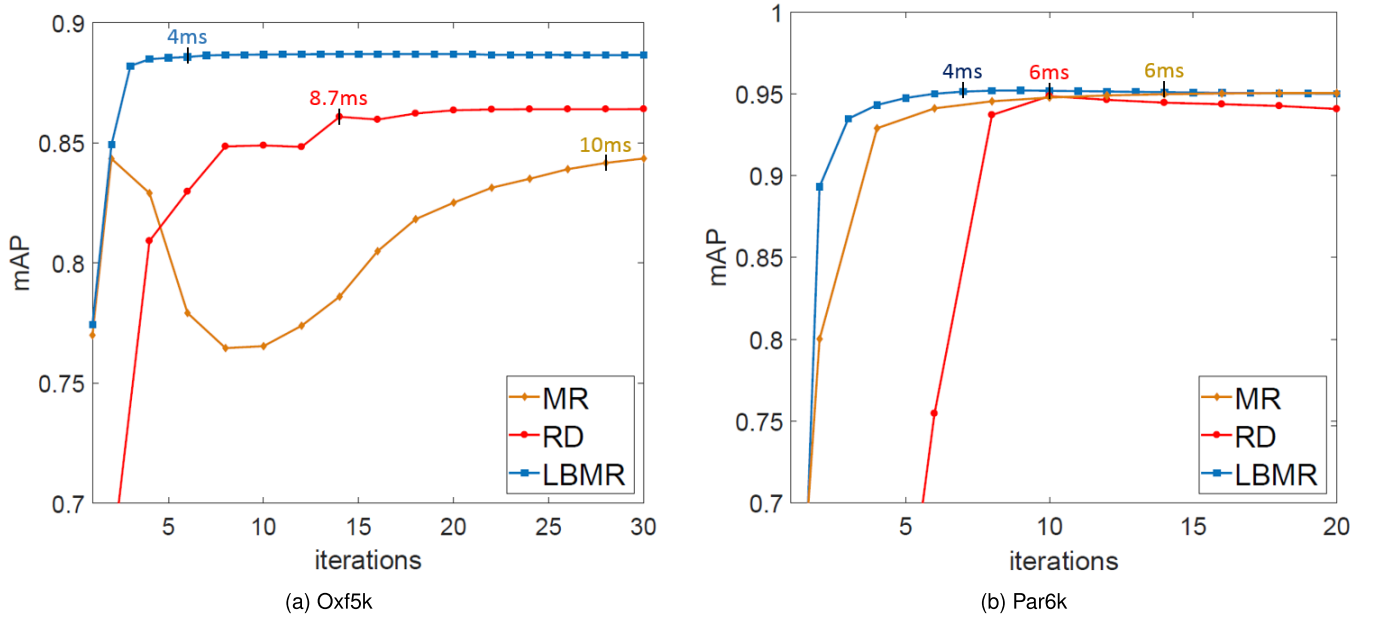


Fig. 4. Ranking mAP versus the number of iterations of three iterative methods with the global descriptors.

TABLE III
RANKING mAP (%) VERSUS DIFFERENT ANCHOR GENERATING
METHODS ON THE Oxf5k DATASET

Method	Max	Min	Average
Global Descriptors			
RS	87.3	80.0	84.3
<i>k</i> -means	89.3	85.4	88.1
FKM	87.1	82.9	84.5
BKM	90.6	87.1	88.9
Regional Descriptors			
RS	90.7	90.0	90.3
<i>k</i> -means	92.6	92.0	92.4
FKM	92.0	91.5	91.8
RKM	92.8	92.4	92.6

TABLE IV
ANCHOR SELECTION RESULTS OF LBMR. *m* REPRESENTS THE
NUMBER OF GENERATED ANCHORS AND *r* REPRESENTS
THE NUMBER OF SELECTED ANCHORS

Dataset	<i>m</i>	<i>r</i>	Speed up	mAP drop (%)
Global Descriptors				
Oxf5k	1500	300	1.3	-0.2
INSTRE	1,500	300	2	-0.5
Par106k	20,000	4,000	2.5	-0.2
Regional descriptors				
Oxf5k	30,000	5,000	1.8	-0.1
Par6k	20,000	5,000	1.6	-0.3

outperforms other methods on most datasets, especially on the INSTRE and all versions of the Oxford dataset. For example, on the Oxf5k-m dataset with 512 features, our proposed method achieves a greater than 14% improvement compared to the second-best method RD and a nearly 6% improvement compared to the second-best method RD. We did not report the

results of all methods on the Oxf1m and Par1m datasets with 2048 features, and EMR on the Oxf105k and Par106k datasets due to the out-of-memory error. Since our proposed method performs ranking on a sample-anchor approximate affinity matrix, its performance relies on the selection of anchors. Our proposed method performs equal or a slight worse than MR on some versions of the Paris dataset, which may be caused by the poor quality of the selected anchors. Besides LBMR, FSR produced better results than the rest five methods on the INSTRE and all versions of the Oxford dataset, while MR produced better results than the rest five methods on all versions of the Paris dataset.

For three iterative methods MR, RD and LBMR, we selected two datasets to show the ranking mAP versus the number of iterations in Figure 4. Both figures indicate that our proposed method can reach a better retrieval performance within less iterations than the other two methods. For example, our proposed method produced better results than MR and RD at the 3-th iteration on the Oxf5k dataset and at the 8-th iteration on the Paris6k dataset. To investigate the online ranking speed of our proposed method, we selected the Oxf5k dataset to compute the average time cost per query in online ranking stage, and the results of MR, EMR, RD, FSR and LBMR are 17ms, 700ms, 8.75ms, 14ms and 4ms, respectively. These results indicate the superior efficiency of our proposed method.

F. Experimental Results with the Regional Descriptors

We used the six datasets in Table II except five large-scale datasets Oxf105k, Par106k, Oxf1m, Par1m and INSTRE (too large to perform regional ranking) to compare the performance our proposed method with the state-of-the-art methods, with the regional descriptors. The results are shown in Table VI. Compared with the results in Table V, most methods (includ-

TABLE V

PERFORMANCE COMPARISON (mAP %) RESULTS WITH THE GLOBAL DESCRIPTORS OF 11 DATASETS. EACH VALUE IS THE RESULT ON 512 FEATURES FOLLOWED WITH THE RESULT ON 2048 FEATURES. # INDICATES THAT THE CORRESPONDING METHOD CANNOT PRODUCE RESULTS DUE TO THE OUT-OF-MEMORY ERROR

Dataset	INSTRE	Oxf5k	Oxf5k-m	Oxf5k-h	Par6k	Par6k-m	Par6k-h	Oxf105k	Par106k	Oxf1m	Par1m
SIM	47.7/62.6	77.7/83.9	56.3/64.7	30.0/38.5	84.1/93.8	69.1/77.2	44.1/56.3	70.1/80.8	76.8/89.9	70.9/#	78.8/#
MR	69.5/81.5	86.2/86.1	58.5/63.3	32.5/40.5	94.8/ 96.8	85.5 /89.3	72.5/ 81.3	79.9/86.7	92.3/ 95.8	86.3/#	94.9 /#
EMR	71.9/80.4	83.7/86.3	59.2/60.3	33.7/34.2	94.6/96.6	83.6/86.1	67.5/73.5	#	#	#	#
RD	70.3/80.5	86.4/87.4	64.2/72.3	35.7/44.1	94.1/96.6	84.3/89.1	70.9/80.3	82.7/87.4	92.5 /92.4	87.4/#	94.7/#
FSR	71.2/80.5	87.2/87.5	56.9/70.4	31.2/41.8	94.5/96.5	84.0/88.6	70.5/77.9	85.0/87.9	92.4/95.3	79.1/#	88.3/#
HD	71.2/79.3	86.4/87.5	64.2/64.7	35.7/40.5	94.6/96.4	84.0/89.0	70.7/80.2	82.5/87.4	92.5 /93.6	86.6/#	93.8/#
LBMR	72.5 / 82.5	88.9 / 90.6	67.8 / 73.8	40.7 / 48.9	95.0 /96.7	84.9/ 90.4	72.7 /80.0	85.1 / 88.1	92.5 /95.3	88.6 /#	94.5/#

TABLE VI

PERFORMANCE COMPARISON (mAP %) RESULTS ON THE REGIONAL DESCRIPTORS. EACH VALUE IS THE RESULT ON 512 FEATURES FOLLOWED WITH THE RESULT ON 2048 FEATURES. # INDICATES THAT THE CORRESPONDING METHOD CANNOT PRODUCE RESULTS DUE TO THE OUT-OF-MEMORY ERROR

Dataset	Oxf5k	Oxf5k-m	Oxf5k-h	Par6k	Par6k-m	Par6k-h
SIM	81.5/88.1	53.2/62.4	26.6/34.5	86.1/94.9	65.4/75.7	39.1/56.5
MR	91.1/92.1	68.4/78.8	44.6/55.3	95.4/96.1	86.0 /92.5	74.8 /84.5
EMR	#	#	#	#	#	#
RD	93.2 /95.8	69.3/81.2	46.0/58.7	96.5 /96.9	85.4/ 92.8	73.8/85.0
FSR	93.0/95.8	69.0/80.9	45.1/58.5	96.5 /97.0	85.2/92.7	72.6/84.5
HD	92.6/ 96.0	69.3/81.2	46.0/58.6	96.4/97.2	85.4/ 92.8	73.4/ 85.2
LBMR	92.6/94.4	69.5 / 82.0	47.0 / 60.6	96.5 / 97.3	85.0/92.4	73.7/84.5

ing our proposed method) achieve improvements, which indicates that regional descriptor is able to improve the ranking performance. An examination of the results in Table VI shows that our proposed method produces the best results on the Oxf5k-m, Oxf5k-h and Par6k datasets, and comparable results on the rest three datasets which may be caused by the poor quality of the selected anchors. Since the regional descriptors contains 20 more times of features than the global descriptors, it is more difficult to find high quality anchors with the regional descriptors than with the global descriptors. Besides LBMR, RD produces better results than the rest five methods on most datasets. EMR even produces none of results on all datasets due to the out-of-memory error. On the Oxf5k-m dataset, the online ranking time per query for MR, RD, FSR and LBMR are 2000ms, 600ms, 240ms and 450ms respectively. FSR is the fastest one but it requires performing the time-consuming rank-r eigendecomposition of a large matrix in an offline step. Our proposed method is faster than MR and RD. Therefore, our proposed method is able to achieve a trade-off between the effectiveness and efficiency.

VI. CONCLUSION

This paper presents a fast manifold ranking method, namely Local Bipartite Manifold Ranking (**LBMR**). In the new method, a regional k -means (**RKM**) was proposed to obtain high quality anchors for constructing a *local bipartite graph* that is able to capture the local structure of the dataset. An iterative method is proposed to directly solve the manifold ranking problem from the *local bipartite graph* and is able to monotonically decrease objective function until converges. Compared

with the original manifold ranking methods, the new method has good monotonicity property and is empirically proved to be able to produce better results.

However, in our experiments, RKM is still too space consuming to deal with large-scale datasets such as Oxf105k and Par106k. Especially our experimental results show that we have to set a large number of anchors m in order to ensure the performance. In the future work, we will introduce approximation technique to improve RKM for large-scale task, and improve the offline graph construction method in order to reduce the number of anchors m . Moreover, our proposed method separably ranks multiple regions but it may be more reasonable to rank them by considering that some regions are extracted from the same image.

APPENDIX

Proof: Denote \mathbf{F} in the it -th and $it + 1$ -th iterations as \mathbf{F}_{it+1} and \mathbf{F}_{it} , respectively. Let $E = \gamma(\delta \mathbf{I}_{nt} - \Theta \Theta^T)$. Since \mathbf{F}_{it+1} is the optimal solution to problem (17), we have

$$\begin{aligned} & Tr(\mathbf{F}_{it+1}^T \mathbf{E} \mathbf{F}_{it}) + \frac{1}{2} Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it+1}^T) \\ & \geq Tr(\mathbf{F}_{it}^T \mathbf{E} \mathbf{F}_{it}) + \frac{1}{2} Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it}^T) \quad (35) \end{aligned}$$

It can be verified that all eigenvalues in \mathbf{E} are non-negative, so \mathbf{E} is positive semi-definite. Then we can rewrite $\mathbf{E} = \mathbf{Q}_E^T \mathbf{Q}_E$ via Cholesky factorization. Then Eq. (35) can be rewritten as

$$\begin{aligned} & Tr(\mathbf{F}_{it+1}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it}) \\ & + \frac{1}{2} Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it+1}^T) \end{aligned}$$

$$\begin{aligned} &\geq Tr(\mathbf{F}_{it}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it}) \\ &\quad + \frac{1}{2} Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it}^T) \end{aligned} \quad (36)$$

Rewrite the inequation $\|\mathbf{Q}_E \mathbf{F}_{it+1} - \mathbf{Q}_E \mathbf{F}_t\|_F^2 \geq 0$ as

$$\begin{aligned} &Tr(\mathbf{F}_{it+1}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it+1}) - 2Tr(\mathbf{F}_{it+1}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it}) \\ &\quad + Tr(\mathbf{F}_{it}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it}) \geq 0 \end{aligned} \quad (37)$$

Multiplying Eq. (36) by 2 and summing over it with Eq. (37) gives

$$\begin{aligned} &Tr(\mathbf{F}_{it+1}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it+1}) \\ &\quad + Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it+1}^T) \\ &\geq Tr(\mathbf{F}_{it}^T \mathbf{Q}_E^T \mathbf{Q}_E \mathbf{F}_{it}) + Tr((2\mathbf{Z}^T \mathbf{H} \\ &\quad - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it}^T) \end{aligned} \quad (38)$$

which equals to

$$\begin{aligned} &Tr(\mathbf{F}_{it+1}^T \mathbf{E} \mathbf{F}_{it+1}) + Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_d^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it+1}^T) \\ &\geq Tr(\mathbf{F}_{it}^T \mathbf{E} \mathbf{F}_{it}) + Tr((2\mathbf{Z}^T \mathbf{H} - \mathbf{1}_{nt} \mathbf{1}_p^T (\mathbf{H} \circ \mathbf{H})) \mathbf{F}_{it}^T) \end{aligned} \quad (39)$$

The above inequation can be further rewritten as

$$\begin{aligned} &\|\mathbf{Z} - \mathbf{H} \mathbf{F}_{it+1}^T\|_F^2 + \gamma Tr(\mathbf{F}_{it+1}^T \Theta \Theta^T \mathbf{F}_{it+1}) \\ &\leq \|\mathbf{Z} - \mathbf{H} \mathbf{F}_{it}^T\|_F^2 + \gamma Tr(\mathbf{F}_{it}^T \Theta \Theta^T \mathbf{F}_{it}) \end{aligned} \quad (40)$$

Therefore, Algorithm 1 monotonically decrease problem (10) in each iteration. ■

REFERENCES

- [1] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
- [2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [3] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2003, pp. 1470–1477.
- [4] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV)*, vol. 5302, 2008, pp. 304–317.
- [5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [7] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
- [8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8689, Sep. 2014, pp. 584–599.
- [9] Y. H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 53–61.
- [10] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.
- [11] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 321–328.
- [12] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds., 2003, pp. 169–176.
- [13] B. Xu *et al.*, "Efficient manifold ranking for image retrieval," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 525–534.
- [14] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 926–935.
- [15] R. He, Y. Zhu, and W. Zhan, "Fast manifold-ranking for content-based image retrieval," in *Proc. Int. Colloq. Comput., Commun., Control, Manage. (ISECS)*, vol. 2, Aug. 2009, pp. 299–302.
- [16] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum, "Fast spectral ranking for similarity search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–10.
- [17] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chu, "Hybrid diffusion: Spectral-temporal graph filtering for manifold ranking," in *Proc. 14th Asian Conf. Comput. Vis.*, 2018, pp. 301–316.
- [18] Y. Fujiwara, G. Irie, S. Kuroyama, and M. Onizuka, "Scaling manifold ranking based image retrieval," *Proc. VLDB Endowment*, vol. 8, no. 4, pp. 341–352, Dec. 2014.
- [19] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1320–1327.
- [20] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *Siam J. Matrix Anal. Appl.*, vol. 31, no. 3, p. 2009, 2009.
- [21] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision—ECCV 2014*, 2014, pp. 392–407.
- [22] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. ICLR*, 2016, pp. 1–12.
- [23] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual instance retrieval with deep convolutional networks," *ITE Trans. Media Technol. Appl.*, vol. 4, no. 3, pp. 251–258, 2016.
- [24] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [25] W. Liu, J. He, and S. F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 679–686.
- [26] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.
- [27] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1206–1215.
- [28] X. Chen, J. Z. Huang, F. Nie, R. Chen, and Q. Wu, "A self-balanced min-cut algorithm for image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2061–2069.
- [29] X. Chen, F. Nie, J. Z. Huang, and M. Yang, "Scalable normalized cut with improved spectral rotation," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 1518–1524.
- [30] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 660–673.
- [31] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
- [32] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [33] S. Wang and S. Jiang, "Instre: A new benchmark for instance-level object retrieval and recognition," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 3, pp. 37:1–37:21, Feb. 2015.
- [34] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, O. Chum, and I. Rennes, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5706–5715.
- [35] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–20.



Xiaojun Chen (Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology in 2011. He is currently an Associate Professor with the College of Computer Science and Software, Shenzhen University. His research interests include machine learning and data mining.



Qingyao Wu (Senior Member, IEEE) received the B.S. degree in software engineering from the South China University of Technology, China, in 2007, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 2009 and 2013, respectively. He is currently a Professor with the School of Software Engineering, South China University of Technology. His current research interests include computer vision and data mining.



Yuzhong Ye is currently pursuing the master's (M.A.) degree with the School of Software Engineering, South China University of Technology, China. His research interests include image processing and deep learning.



Feiping Nie (Senior Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, China, in 2009. He has published more than 100 articles in the following top journals and conferences: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *IJCV*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS/IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *TKDD*, *Bioinformatics*, *ICML*, *NIPS*, *KDD*, *IJCAI*, *AAAI*, *ICCV*, and *CVPR ACM MM*. His research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He is also serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.