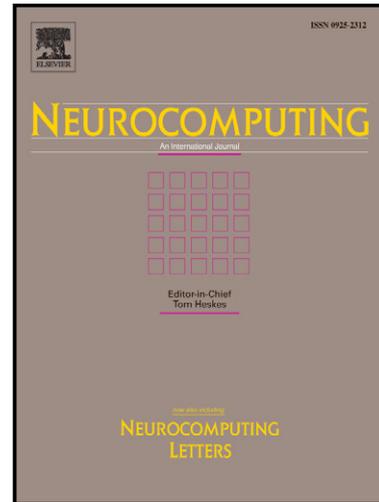


# Author's Accepted Manuscript

Multi-Query parallel field ranking for image retrieval

Ji Yang, Bin Xu, Binbin Lin, Xiaofei He



[www.elsevier.com/locate/neucom](http://www.elsevier.com/locate/neucom)

PII: S0925-2312(14)00051-4  
DOI: <http://dx.doi.org/10.1016/j.neucom.2013.12.033>  
Reference: NEUCOM13857

To appear in: *Neurocomputing*

Received date: 30 July 2013  
Revised date: 4 December 2013  
Accepted date: 19 December 2013

Cite this article as: Ji Yang, Bin Xu, Binbin Lin, Xiaofei He, Multi-Query parallel field ranking for image retrieval, *Neurocomputing*, <http://dx.doi.org/10.1016/j.neucom.2013.12.033>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Multi-Query Parallel Field Ranking for Image Retrieval

Ji Yang<sup>a</sup>, Bin Xu<sup>b</sup>, Binbin Lin<sup>a</sup>, Xiaofei He<sup>a</sup>

<sup>a</sup>State Key Lab of CAD&CG, College of Computer Science, Zhejiang University,  
Hangzhou, China

<sup>b</sup>Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science,  
Zhejiang University, Hangzhou, China

---

## Abstract

Relevance feedback image retrieval is an effective scheme bridging the gap between low-level features and high-level concepts. It is essentially a multi-query ranking problem where the user submitted image and provided positive examples are considered as queries. Most of the existing approaches either merge the multiple queries into a single query or consider them independently, and then the geodesic distances on the image manifold are used to measure the similarities between the query image and the other images in database. In this paper, we propose a novel approach called Multi-Query Parallel Field Ranking (**MQPFR**) which finds an optimal ranking function whose gradient field is as parallel as possible. In this way, the obtained ranking function varies linearly along the geodesics of the data manifold, and achieves the highest value at the multiple queries simultaneously. Extensive experiments are carried out on a large image database and demonstrate the effectiveness of the proposed approach.

*Keywords:* Image Retrieval, Parallel Vector Field, Multi-query

---

## 1. Introduction

Content-Based Image Retrieval (CBIR) has received considerable interest recently [1], partly because of the rapid growth of the mobile devices. Unlike traditional keyword based search systems, CBIR utilizes the low-level visual features automatically extracted from images, including global features (e.g. color moment, edge histogram, LBP) and local features (e.g. SIFT). How to narrow down the semantic gap between low-level features and high-level concepts is a challenging problem.

To bridge the semantic gap, relevance feedback is introduced into CBIR to capture the subjectivity of human perception of images through interactions with the user [2]. It was shown to dramatically increase the retrieval performance. Most of previous relevance feedback methods can be classified into two categories according to the way they deal with the user submitted query, as well as the user provided positive examples. The first merges multiple queries into a single one and considers image retrieval as a ranking problem [3]. The other considers the problem as classification [4, 5]. However, sometimes the users only provide positive examples, in which case classification algorithms can not be directly applied due to the lack of negative examples. In this work, we consider relevance feedback image retrieval as a multi-query ranking problem and aims to learn a ranking function whose highest values are achieved at the multiple queries simultaneously.

Many manifold-based ranking approaches have been proposed [6, 7, 8, 9, 10, 11], following the intuition that naturally occurring data (e.g. images) may be generated by structured systems with possibly much fewer degrees of freedom than the ambient dimension would suggest [12, 13]. These approaches usually estimate the data manifold by an affinity graph, and the Laplacian regularizer constructed over the graph is thus adopted to ensure the smoothness of the learned ranking function along the geodesics of the data manifold. It has been shown that manifold-based approaches have significantly improved image retrieval performance. However, one of the major limitations is that Laplacian regularization can only ensure smoothness, while an optimal ranking function should preserve the ranking order of the data points along the geodesics. In other words, the ranking function should vary monotonically along the geodesics on the data manifold.

In this paper, we propose a novel algorithm, called Multi-Query Parallel Field Ranking (**MQPFR**), for learning an optimal ranking function on the data manifold which varies linearly along the geodesics and achieves the highest ranking score at the multiple queries. In order to find such a function, we note that recent theoretical works show that its gradient field has to be a parallel vector field [14, 15]. Thus, we adopt the same idea to learn a ranking function  $f$  and a vector field  $V$  simultaneously such that  $\nabla f$  is as close to  $V$  as possible and  $\nabla V$  vanishes. Moreover, the user submitted query and the provided positive examples are equally treated as multiple queries by requiring that, for each one of them, the tangent vectors of its nearest neighbors should all point to it. In this way, our proposed approach effectively makes use of the multiple queries and the intrinsic distribution of data.

## 2. Related Work

Image retrieval has been researched intensively, with a large number of content-based ranking methods proposed every year. While earlier image retrieval algorithms rank data directly according to the Euclidean distance of simple image features like color features [16], more recently proposed algorithms like [17] learns a parameterized similarity function among all data based on pairwise similarity. In order to exploit the intrinsic distribution of all data, Zhou et al proposed a transductive ranking algorithm called Manifold Ranking (MR) by diffusing the label information among data neighborhoods via a heat equation [6]. It is significantly different from distance-based ranking algorithms which only consider pairwise distances or inner products as shown in [18, 7]. Recently, more algorithms that directly make use of the data manifold assumption [12, 13] and address specific issues of the original MR have been proposed. For example, [10] and [11] solve for linear or nonlinear projections of the original image features before comparing the Euclidean distance among data. The LRG algorithm proposed in [8] learns the Laplacian matrix for ranking data using local linear regression rather than directly computing it using Gaussian kernel. [9] addresses multi-modality retrieval tasks by harmonizing hierarchical data manifolds, and thus can be will applied in scenarios like multi-task and cross-media retrieval. Still in other frameworks, extra label information can be incorporated as explicit constraints into the projection process [19] and more powerful and robust techniques such as the iterated graph Laplacian [20],  $k$ -regular nearest neighbor graph [21], anchor graph [22] and parallel vector field [15] are adopted for better performance and scalability of ranking on data manifolds. Compared with inductive learning algorithms, those frameworks can make use of both unlabelled and labelled data for ranking and thus yield more stable and accurate ranking results.

Relevance feedback has been shown helpful in many image retrieval systems [23, 10, 7, 11, 9, 24, 2, 3, 4, 5]. Short-term relevance feedback algorithms that only consider feedback information provided by the current user are usually derived directly from some manifold-based ranking algorithms with carefully constructed queries [7, 9], or from classification algorithms with ranking scores computed from the decision values [4, 5]. To make use of the feedback information provided by more users, various long-term relevance feedback algorithms have also been proposed [23, 9, 10]. Moreover, other algorithms like [25, 26] apply active learning to achieve better understanding

of user preferences at the cost of less flexible interactive processes.

Our algorithm addresses short-term relevance feedback image retrieval as a multi-query ranking problem and exploits the intrinsic distribution of whole data to improve ranking results, which is generally similar to MR and LRGGA [7, 9]. But unlike them, we employ the parallel vector field to ensure the linearity of our ranking function and we adopt the anchor graph to speed up the optimization processes.

### 3. Multi-Query Parallel Field Ranking

In this section, we begin with the motivation of our algorithm and then introduce the objective function which learns a multi-query ranking function on the data manifold.

#### 3.1. From Single Query to Multiple Queries

The generic problem of image retrieval can be described as following. Given an image database  $\{x_i\}_{i=1}^n \subset \mathbb{R}^m$  and an initial query image  $q_1 \in \mathbb{R}^m$ , learn a ranking function  $f$  such that  $f(x_i)$  reflects the semantic relationship between  $x_i$  and  $q_1$ . However, in many real applications, only one query is not enough to convey useful information and relevance feedback is an effective way of enhancing the learning process. The typical relevance feedback based retrieval process can be outlined as follows:

1. The system presents the top ranked images to the user by using a pre-defined ranking function  $f$ , such as Euclidean distance function.
2. The user provides his relevance feedback to the system by labeling some interested images as ‘positive’.
3. The system learns a new ranking function  $f$  by using the feedback information and re-ranks the images in the database.

Since each image marked by the user as positive represents the user’s information need, they should all be equally treated as multiple queries. Moreover, there is usually no negative example available, so the image retrieval problem here can not be considered as classification. Given these queries, most of the traditional methods essentially merge them into a single query. Consider a ranking problem in a 2-dimensional space, Figure 1(a) shows a result obtained by one of the state-of-the-art approach [15]. In their approach, the obtained ranking function can vary linearly along the geodesics on the data manifold and reaches the highest value at a single point. Therefore, when dealing

with more than one queries, for example, the three queries marked by ‘●’ in Figure 1, they can only merge them into a single one, which is marked by ‘▲’. However, there is no guarantee that this single query represents the most relevant information due to the complexity of large databases, where relevant data points are not necessarily clustered tightly around a single prototype [27]. Therefore, important information may get lost during the merging process and the ranking function may lack predictive power as a result. Unlike most of the traditional approaches, our approach explicitly considers the local geometrical structure of each query and yields highest ranking score at the multiple queries simultaneously, as shown in Figure 1(b). In this way, we make better use of the user submitted information and enhance the chances of reliably learning user’s favored patterns.

### 3.2. Our Algorithm

In order to find a ranking function which effectively makes use of the multiple queries and the geometrical structure of the data manifold, we use techniques of parallel vector field [14]. A vector field  $V$  on the manifold  $\mathcal{M}$  is said to be parallel if  $\nabla V \equiv 0$ , where  $\nabla$  is the covariant derivative on  $\mathcal{M}$ . According to [28], if the gradient field  $\nabla f$  of the ranking function  $f$  is a parallel vector field,  $f$  will vary linearly along any geodesics on the manifold. Therefore, a linear function on the manifold can be obtained by requiring its gradient field to be a parallel vector field. Therefore, we construct constraints in the form of regularization terms to learn a ranking function  $f$  and a parallel vector field  $V$  simultaneously as follows [14, 15]:

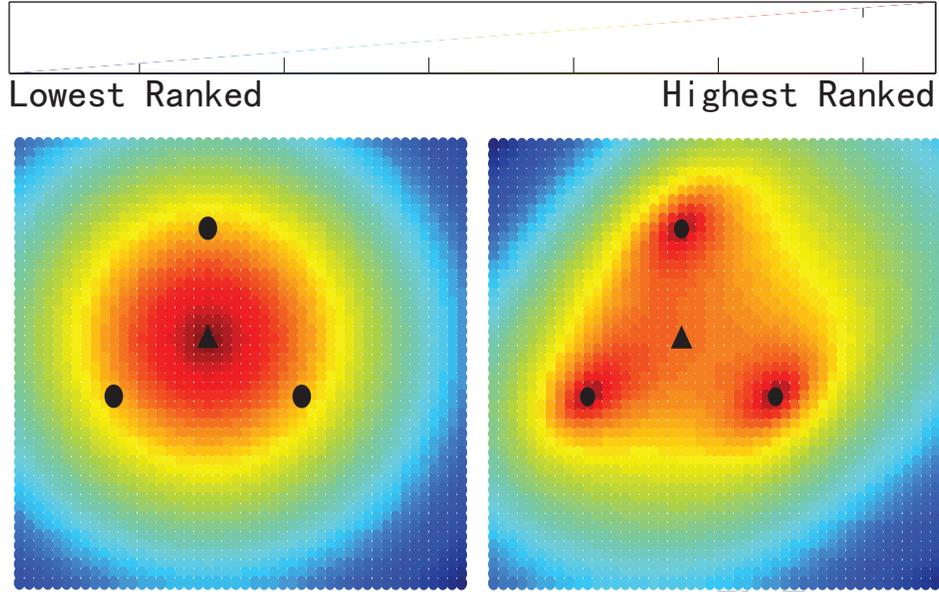
- The gradient field  $\nabla f$  of the ranking function should be close to the vector field  $V$ , so we have the first constraint as

$$\min_{f \in C^\infty, V} R_1(f, V) = \int_{\mathcal{M}} \|\nabla f - V\|^2; \quad (1)$$

- The vector field  $V$  should be as parallel as possible, so we have the second constraint as

$$\min_V R_2(V) = \int_{\mathcal{M}} \|\nabla V\|_F^2, \quad (2)$$

where  $C^\infty(\mathcal{M})$  denotes smooth functions on  $\mathcal{M}$  and  $\|\cdot\|_F$  denotes the Frobenius norm.  $\nabla$  measures the change of the vector field  $V$ .  $V$  becomes a parallel vector field when  $\nabla$  vanishes.



(a) Rank by merging three queries into a single one (b) Rank with three queries simultaneously

Figure 1: **A toy example illustrating the ranking results with single query and multiple queries. In this example, three query points (marked by ‘•’) in a two-dimensional space are given. The color represents the ranking score from the highest (red) to the lowest (blue). (a) Ranking result obtained by the Parallel Field Ranking algorithm. The three queries are merged into a single query, which is marked by ‘▲’. As it can be seen, the ranking function fails to achieve the highest score at the three queries which represent user needs. (b) Ranking result obtained by our approach. We can see that the highest ranking score is achieved at the three queries simultaneously.**

Secondly, we need to ensure that the ranking function  $f$  achieves the highest score at the multiple queries. Let  $\mathbf{Q} = (q_1, q_2, \dots, q_s)$  denote the queries. Without loss of generality, we assume that  $x_i = q_i$ ,  $i = 1, \dots, s$ . In this work, we consider the case that the images are sampled from some underlying  $d$ -dimensional manifold  $\mathcal{M}$  which is embedded in the ambient Euclidean space  $\mathbb{R}^m$ . That is,  $x_i \in \mathcal{M}$ ,  $i = 1 \dots, n$ . Thus, for each query  $q_i$ , we require that the tangent vectors of its neighboring points should all point to  $q_i$ . Note that, the vector field  $V$  is as parallel as possible. That is, the change between two neighboring tangent vectors is sufficiently small. Therefore, for any point  $x \in \mathcal{M}$ , there is a geodesic curve connecting  $x$  and

$q_i$  such that the ranking function increases linearly from  $x$  to  $q_i$ . In other words,  $f$  achieves the highest value at the multiple queries  $q_i$  simultaneously and then decreases linearly along the geodesics passing through those queries to nearby data points. Specifically, we have the third constraint as

$$\min_V R_3(V) = \sum_{t=1}^s \int_{z \in U_{q_t}} \|V_z - P_z(q_t - z)\|^2, \quad (3)$$

where  $U_{q_t}$  is a neighborhood of  $q_t$ ,  $V_z$  denotes the tangent vector at point  $z$  and  $P_z$  is the projection operator which projects a vector to the tangent space  $T_z\mathcal{M}$ . In practice, we can choose  $k$  nearest neighbors or  $\epsilon$  nearest neighbors of the query as neighborhood points. With the three regularization terms  $R_1$ ,  $R_2$ , and  $R_3$ , we finally get the following objective function:

$$\begin{aligned} & \arg \min_{f \in C^\infty(\mathcal{M}), V} J(f, V) \\ &= R_0(y, f) + \lambda_1 R_1(f, V) + \lambda_2 R_2(V) + \lambda_3 R_3(V), \end{aligned} \quad (4)$$

where  $R_0$  is a simple loss function that forces the ranking scores of the queries to be close to a positive number, thus ensuring the solution to be unique. The trade-off among the other three regularization terms is controlled by the coefficients  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  in the range of  $(0, +\infty)$ .

### 3.3. Scalable Graph Construction

Like many other graph-based algorithms, before actually conducting the label inference, we need to firstly construct a weight graph which represents the data adjacency. A common resort is to employ the  $k$ -nearest neighbor ( $k$ NN) graph due to its ability to capture local structure of the data as well as its simplicity. However, the  $O(kn^2)$  construction cost and  $O(n^2)$  memory cost of  $k$ NN are quite expensive, especially in large scale situations. Therefore, to handle scalable data sets, we resort to the recently proposed anchor graph [29] and a new design of adjacency matrix  $W$  proposed in [9].

Given the input samples  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ , the key idea of anchor graph is to introduce a subset  $\mathcal{U} = \{\mathbf{u}_k\}_{k=1}^p \subset \mathbb{R}^m$  called anchors and try to represent the semantic properties  $f$  of  $\mathcal{X}$ , such as label prediction and ranking score as

$$f(\mathbf{x}_i) = \sum_{k=1}^p Z_{ik} f(\mathbf{u}_k), \quad (5)$$

where each  $Z_{ik}$  is a sample-adaptive value which represents the weight between data point  $x_i$  and anchor  $u_k$  and therefore can be used directly to compute the adjacency matrix  $W$ . In this paper, following [9], we firstly use k-means algorithm and select the centers of each cluster as the anchors. Then we use the well known Nadarata-Watson kernel regression to assign the weights smoothly as

$$z_{ik} = \frac{K\left(\frac{|x_i - u_k|}{\lambda}\right)}{\sum_{l=1}^s K\left(\frac{|x_i - u_l|}{\lambda}\right)}, \quad (6)$$

with the *Epanechnikov* quadratic kernel

$$K_\lambda(t) = \begin{cases} \frac{3}{4}(1 - t^2) & \text{if } |t| \leq 1; \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The smoothing parameter  $\lambda$  determining the size of the local region in which anchors can affect the target point is set using the nearest neighborhood size as

$$\lambda(x_i) = |x_i - u_{\{r\}}|, \quad (8)$$

where  $u_{\{r\}}$  is the  $r$ th closest anchor of  $x_i$ .

Then the adjacency matrix  $W$  is computed in a low-rank form

$$W = Z^T Z, \quad (9)$$

which means that if two data points are ‘close’ to each other ( $W_{ij} > 0$ , and we denote it as  $x_i \sim x_j$ ), they share at least one common anchor point. The  $W$  designed this way preserves the good properties of sparseness and non-negativeness of  $Z$ . Moreover, we do not need to compute and save  $W$  in memory but only save the  $p \times n$  matrix  $Z$ , where  $p$  is often far smaller than  $n$ .

Overall, to build the adjacency matrix, we first use k-means clustering algorithm to select informative anchors. Then we connect each data point to its  $r$  nearest anchors and assign weights to each connection by the kernel function. Finally, we use the weight matrix  $Z$  to compute the adjacency matrix  $W$  directly. Note that, by employing anchor graph, we are able to separate the graph construction into an off-line anchor selection stage and an on-line graph construction stage, which might further improve the efficiency because the second stage always has linear complexity and the carefully selected anchors for a large data sets are relatively stable when new samples are added [9].

### 3.4. Discrete Objective Function

With the anchor graph constructed, we are now able to compute the data manifold  $\mathcal{M}$  and discretize the continuous objective function in Eq. (4).

We begin at estimating the local tangent spaces for all data points, which can be efficiently performed using the anchor graph. Firstly, for each  $u_i$  in the anchor set, we estimate its tangent space  $T_{u_i}\mathcal{M}$  by performing PCA on its local neighborhood. The local neighborhood of an anchor  $u_i$  is formed by the  $k$  nearest neighbors of it. We choose the eigenvectors corresponding to the  $d$  largest eigenvalues since we assume that the data manifold  $T_i\mathcal{M}$  is  $d$ -dimensional. Let  $T_{u_i} \in \mathbb{R}^{m \times d}$  be the matrix whose columns constitute an orthonormal basis for  $T_{u_i}\mathcal{M}$ . Then  $P_{u_i} = T_{u_i}T_{u_i}^T$  is the unique orthogonal projection from  $\mathbb{R}^m$  to the tangent space  $T_{u_i}\mathcal{M}$ . This is because for any vector  $a \in \mathbb{R}^m$ , we have  $P_{u_i}a \in T_{u_i}\mathcal{M}$  and  $(a - P_{u_i}a) \perp P_{u_i}a$  [30, 14]. Then we just use  $Z$  to assign the local tangent spaces of all data points  $x_i$  by

$$T_i\mathcal{M} = \sum_{k=1}^p Z_{ik}T_{u_k}\mathcal{M}, \quad (10)$$

where  $Z$  is the weight matrix computed while constructing the anchor graph. It works well because each data point on the manifold can be locally approximated by a linear combination of its nearby anchor points, and the linear weights become its local coordinate coding [31].

Then we denote the discrete ranking function on the data manifold as  $f = [f(x_1), \dots, f(x_n)] = [f_1, \dots, f_n]^T$  and the value of vector field  $V$  at each data point  $x_i$  as  $V_{x_i}$ . Similar to that in [15],  $V_{x_i}$  is a vector in tangent space  $T_{x_i}\mathcal{M}$  and therefore can be represented by the local coordinates of the tangent space. That is,  $V_{x_i} = T_i v_i$ , where  $v_i \in \mathbb{R}^d$ . We define  $\mathbb{V} = [v_1^T, \dots, v_n^T] \in \mathbb{R}^{dn}$  as a  $dn$ -dimensional column vector concatenating all the  $v_i$ 's.

Then  $R_1$ ,  $R_2$  and  $R_3$  reduce to

$$R_1(f, \mathbb{V}) = \sum_{i,j=1}^n \omega_{ij} \left( (x_j - x_i)^T T_i v_i - f_j + f_i \right)^2, \quad (11)$$

$$R_2(\mathbb{V}) = \sum_{i,j=1}^n \omega_{ij} \|P_i T_j v_j - T_i v_i\|^2, \quad (12)$$

$$R_3(\mathbb{V}) = \sum_{t=1}^s \sum_{x_j \sim q_t} \|T_j v_j - P_j(q_t - x_j)\|^2. \quad (13)$$

In order to remove an arbitrary scaling factor and make the solution unique, we let  $y \in \mathbb{R}^n$  be a column vector where the first  $s$  entries (corresponding to the queries) are 1, and all other entries are 0. Thus, we have

$$R_0(y, f) = \sum_{i=1}^s (f_i - y_i)^2. \quad (14)$$

Combining  $R_0$  in Eq. (14),  $R_1$  in Eq. (11),  $R_2$  in Eq. (12) and  $R_3$  in Eq. (13), we get our discrete objective function as follows:

$$\begin{aligned} J(f, \mathbb{V}) &= R_0(y, f) + \lambda_1 R_1(f, \mathbb{V}) + \lambda_2 R_2(\mathbb{V}) + \lambda_3 R_3(\mathbb{V}) \\ &= \sum_{i=1}^s (f_i - y_i)^2 \\ &\quad + \lambda_1 \sum_{i,j=1}^n \omega_{ij} \left( (x_j - x_i)^T T_i v_i - f_j + f_i \right)^2 \\ &\quad + \lambda_2 \sum_{i,j=1}^n \omega_{ij} \|P_i T_j v_j - T_i v_i\|^2 \\ &\quad + \lambda_3 \sum_{t=1}^s \sum_{j \sim q_t} \|T_j v_j - P_j(q_t - x_j)\|^2. \end{aligned} \quad (15)$$

#### 4. Optimization Approach

In this section, we briefly discuss how to solve the objective function Eq. (15). We first rewrite  $R_1$  and  $R_2$  as follows:

$$R_1(f, \mathbb{V}) = 2f^T Lf + \mathbb{V}^T G\mathbb{V} - 2\mathbb{V}^T Cf, \quad (16)$$

$$R_2(\mathbb{V}) = \mathbb{V}^T B\mathbb{V}, \quad (17)$$

where  $L = D - W$  denotes the Laplacian matrix of the graph with  $W_{ij} = \omega_{ij}$  and  $D_{ii} = \sum_{j=1}^n \omega_{ij}$ ,  $G$  is a  $dn \times dn$  block diagonal matrix with  $G_{ii} = \sum_{j \sim i} \omega_{ij} T_i^T (x_j - x_i)(x_j - x_i)^T T_i$ ,  $C$  is a  $dn \times n$  block matrix with  $C_i = \sum_{j \sim i} \omega_{ij} T_i^T (x_j - x_i) s_{ij}^T$ ,  $s_{ij} \in \mathbb{R}^n$  is a selection vector of all zero elements except for the  $i$ -th element being -1 and the  $j$ -th element being 1,  $B$  is a

$dn \times dn$  sparse block matrix with  $B_{ii} = \sum_{j \sim i} \omega_{ij} (Q_{ij} Q_{ij}^T + I)$  and  $B_{ij} = -2\omega_{ij} Q_{ij}$ , and  $Q$  is a  $dn \times dn$  block matrix with  $Q_{ij} = T_i^T T_j$ .

Then we rewrite  $R_0$  as follows:

$$R_0(y, f) = (f - y)^T \mathbb{I} (f - y), \quad (18)$$

where  $\mathbb{I}$  is an  $n \times n$  diagonal matrix with only entries at the  $i$ -th row and  $i$ -th column being 1,  $i = 1, 2, \dots, s$ , and all other entries being 0. Similarly, we rewrite  $R_3$  as follows:

$$R_3(\mathbb{V}) = \mathbb{V}^T D \mathbb{V} - 2H^T \mathbb{V} + \sum_{t=1}^s \sum_{j \sim q_t} \|P_j(q_t - x_j)\|^2, \quad (19)$$

where  $D$  is a  $dn \times dn$  block diagonal matrix with the  $j$ -th  $d \times d$  diagonal block defined as follows:

$$D_{jj} = \begin{cases} I_d, & \text{if } x_j \sim \text{any } q_t \text{ in } \mathbf{Q} \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

and  $H$  is a  $dn \times 1$  block vector with the  $j$ -th  $d \times 1$  block as follows:

$$H_j = \begin{cases} T_j^T (q_t - x_j), & \text{if } x_j \sim \text{any } q_t \text{ in } \mathbf{Q} \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

Finally, combining  $R_0$  in Eq. (18),  $R_1$  in Eq. (16),  $R_2$  in Eq. (17) and  $R_3$  in Eq. (19), we get the simplified matrix form of the objective function in Eq. (15) as follows:

$$\begin{aligned} J(f, \mathbb{V}) = & f^T (\mathbb{I} + 2\lambda_1 L) f - 2y^T \mathbb{I} f - 2\lambda_1 \mathbb{V}^T C f \\ & + \mathbb{V}^T (\lambda_1 G + \lambda_2 B + \lambda_3 D) \mathbb{V} - 2\lambda_3 H^T \mathbb{V} \\ & + y^T \mathbb{I} y + \lambda_3 \sum_{t=1}^s \sum_{x_j \sim q_t} \|P_j(q_t - x_j)\|^2. \end{aligned} \quad (22)$$

Thus the derivatives of  $J(f, \mathbb{V})$  are

$$\frac{\partial J(f, \mathbb{V})}{\partial f} = 2(\mathbb{I} + 2\lambda_1 L) f - 2y - 2\lambda_1 C^T \mathbb{V}, \quad (23)$$

$$\frac{\partial J(f, \mathbb{V})}{\partial \mathbb{V}} = -2\lambda_1 C f + (\lambda_1 G + \lambda_2 B + \lambda_3 D) \mathbb{V} - 2\lambda_3 H. \quad (24)$$

To make the derivatives vanish, we solve the following linear equation system:

$$(\mathbb{I} + 2\lambda_1 L)f - \lambda_1 C^T \mathbb{V} = y, \quad (25)$$

$$-\lambda_1 C f + (\lambda_1 G + \lambda_2 B + \lambda_3 D)\mathbb{V} = \lambda_3 H, \quad (26)$$

which can be simplified to the following linear system:

$$\begin{pmatrix} \mathbb{I} + 2\lambda_1 L & -\lambda_1 C^T \\ -\lambda_1 C & \lambda_1 G + \lambda_2 B + \lambda_3 D \end{pmatrix} \begin{pmatrix} f \\ \mathbb{V} \end{pmatrix} = \begin{pmatrix} y \\ \lambda_3 H \end{pmatrix}. \quad (27)$$

Since the linear system is a well-studied problem and the systems derived in our algorithm are especially sparse, we can solve the ranking problem efficiently using means of decomposition and iteration [30].

## 5. Computational Complexity Analysis

The computational complexity of our proposed MQPFR algorithm is dominated by three parts: constructing the anchor graph, computing local tangent space and solving a sparse linear system. For the anchor graph construction, the overall complexity is  $O(pmn + prmn)$ , where  $n$  is the data size,  $m$  is the dimension of raw data,  $p$  is the number of anchors and  $r$  is the number of nearest anchors around a data point. Specifically, the k-means algorithm for selecting anchors takes  $O(pmn)$ 's time, and the computing of  $Z$  takes  $O(prmn)$ 's time. The complexity of local PCA is  $O(mk^2)$ , where  $k$  is the size of largest local neighborhood. Therefore, the complexity of computing the local tangent space for all data points is  $O(mp k^2 + pn)$ . Solving the final sparse linear system has the complexity of  $O(kd^2 n)$ , where  $d$  is the dimension of underlying data manifold. In this way, the total computational complexity of our MQPFR method is  $O(pmn + prmn + mp k^2 + pn + kd^2 n)$ . Empirically,  $p$ ,  $r$ ,  $k$  and  $d$  can all be fixed as small constants that are irrelative to the size of specific data sets. Therefore, the total computational complexity could be  $O(mn)$ .

It is worth noting that, although the sparse linear system can be solved in linear time using iterations theoretically, it often takes relatively long time when the matrices are extremely huge. Therefore, solving a large sparse linear system to compute the parallel vector field might not be the best way. Since the vector field are only required to be parallel along the geodesics passing

through the queries, a more efficient way to learn the vector field might be propagating the tangent vectors from around the queries to other regions parallelly on the data manifold. In this way, the accuracy of the learned vector field will be well preserved around the queries, which is desirable for the ranking problem.

## 6. Experimental Results

In this section, we investigate the performance of our proposed algorithm for multi-query ranking problem. Several illustrative examples and experimental evaluations are provided. We begin with two synthetic examples to give some intuition about how our algorithm works.

### 6.1. Two Synthetic Examples

Two synthetic examples are given in Figure 2. In the first example, we randomly sampled 2,000 data points from a punctured sphere and selected three points as queries. The ranking result obtained by using our proposed algorithm is shown in Figure 2(a). The color represents the ranking scores from the highest (red) to the lowest (blue). In the second example, we randomly sampled 2,000 data points from a twin-peak surface and selected two points as queries. Figure 2(c) shows the ranking result. As it can be seen, in both examples, the obtained ranking functions achieve their highest values at the multiple queries simultaneously.

In order to show how the obtained ranking functions vary on the manifolds, we also present their corresponding vector fields learned by our algorithm in Figure 2(b) and 2(d). As shown in each example, the vector field remains constant along the geodesics passing through the queries on the manifold. This indicates that the obtained ranking functions vary linearly along the geodesics and consequently well preserve the order of the data points with respect to the multiple queries.

### 6.2. Image Retrieval Experiments

In this subsection, we evaluate the performance of our proposed algorithm for multi-query image retrieval in two real world image databases. The results of both relevance feedback experiments and pure multi-query experiments are shown and we begin with a description of the data preparation.

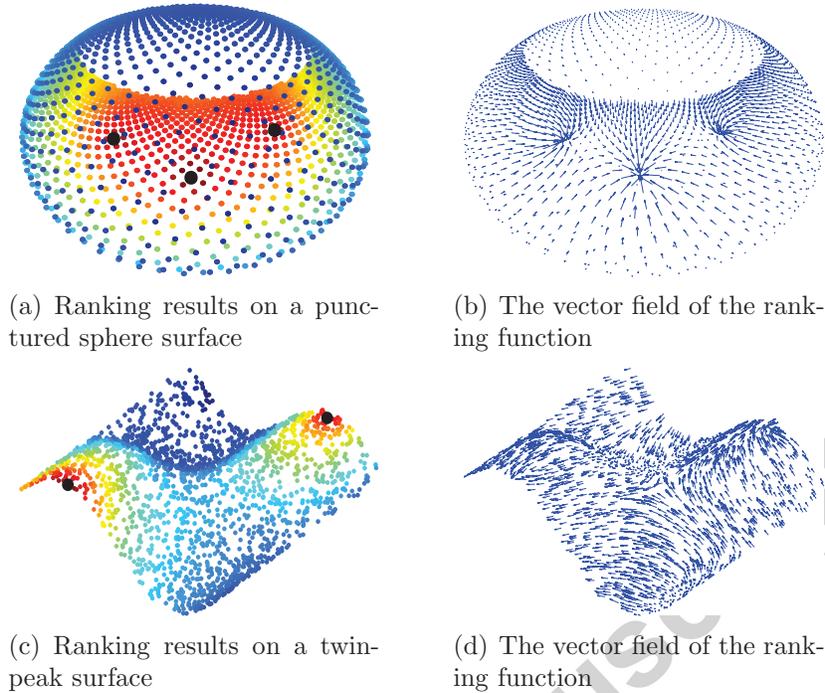


Figure 2: Two simple synthetic examples. The ranking results are obtained by using our proposed MQPFR algorithm and the color represents the ranking scores from the highest (red) to the lowest (blue). (a) Ranking results on a punctured sphere. The three query points are marked by ‘•’. (b) The obtained vector field on the punctured sphere. (c) Ranking results on a twin-peak surface. The two query points are marked by ‘•’. (d) The obtained vector field on the twin-peak surface. As it can be seen, the obtained ranking functions achieve their highest scores at the multiple queries simultaneously. Moreover, the vector fields show how the ranking functions increase from the distant points to the query points and they are indeed quite parallel.

### 6.2.1. Data Preparation

Two real world data sets are used in our experiments. The first one consists of 5,000 images of 50 semantic categories, from the widely used Corel image database. Each of the 50 different categories has 100 images, such as *cat*, *lion*, *lighthouse*, *couple* and so on. Images within the same category are judged relevant and otherwise irrelevant. Figure 3 shows some sample images from the Corel database. For each image, we extract a 297-dimensional feature vector which combines the following information:

- Grid Color Moment: Each image is partitioned into  $3 \times 3$  grids. For

each grid, three color moments: mean, variance and skewness are extracted in each color channel (R, G, and B) respectively. Thus, an 81-dimensional grid color moment vector is adopted.

- Edge: The Canny Edge detector [32] is used to obtain the edge map for the edge orientation histogram, which is quantized into 36 bins of 10 degrees each. An additional bin is to count the number of pixels without edge information. Hence, a 37-dimensional vector is used.
- Gabor Wavelets Texture: Each image is first scaled to  $64 \times 64$  pixels. The Gabor wavelet transform [33] is then applied on the scaled image with 5 levels and 8 orientations, which results in 40 subimages. For each subimage, 3 moments are calculated, i.e., mean, variance and skewness. Thus, a 120-dimensional vector is used.
- Local Binary Pattern: The LBP [34] is a gray-scale texture measure derived from a general texture definition in a local neighborhood. A 59-dimensional LBP histogram vector is adopted.

The second data set is from the CMU PIE face database [35]. This database contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying poses, illuminations and expressions. In this experiment, we choose the frontal pose (C27) with varying lighting conditions, which leaves us 42 images per subject. Preprocessing to locate the faces were applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then the facial areas were cropped into the final image for matching. The size of each cropped image in all the experiments is  $32 \times 32$ , with 256 gray levels per pixel. Therefore, each image can be represented by a 1024-dimensional feature vector in the image space. No further preprocessing is done. Figure 4 shows some sample faces of two different people in the CMU PIE database.

### 6.2.2. Experimental Settings

Firstly, we designed an automatic feedback scheme to simulate the real relevance feedback image retrieval process. We begin with a single query randomly picked out from the data set. Since the user usually only cares about the most relevant images and the irrelevant images are usually very diverse, we only require the user to provide positive examples. At each iteration,



Figure 3: 16 randomly selected sample images from Corel image database of semantic concept *cat*, *lion*, *lighthouse* and *couple*.



Figure 4: Some sample face images of two different people from CMU PIE dataset. The First two rows belong to the one subject while the next two rows belong to another.

the user is required to provide one positive example, which is simulated by randomly picking out one relevant image from the top ranked images. Then different algorithms are performed to re-rank the images in the database. Ten iterations are performed until there are ten queries.

Secondly, since in some cases the user may have more than one query images at the beginning or during a certain feedback iteration, we also compare different algorithms with multiple query images right from the start. We call it pure multi-query image retrieval because no feedback iterations are involved.

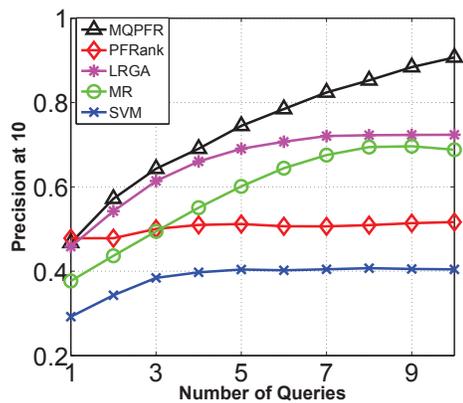
For each experiment of the two kinds, the evaluation metrics are computed by averaging the results from ten separate runs for each category of the data sets. It is worth noticing that most existing graph-based algorithms need to construct a nearest neighbor graph over the data and invert a matrix whose size is at least  $n \times n$ , which is time consuming. In order to perform the methods more efficiently, we first rank all the images according to the Euclidean distances to the original query image and then choose the top 500 images as candidates. We then perform different ranking algorithms on this small subset. The evaluation metrics used in our experiments include Precision, Recall, Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [36], which provide a comprehensive view of the ranking performance of different algorithms.

For our MQPFR algorithm, the dimensionality of the manifold ( $d$ ) in real data sets is unknown. We perform cross-validation and choose  $d = 2$  for Corel data set and  $d = 5$  for CMU PIE data set. The intrinsic parameters of our algorithm  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are all set to 0.01 except for model selection. For other parameters, we empirically set the number of anchors ( $p$ ) to be 300, the number of nearest anchors for representation ( $r$ ) to be 8 and the number of nearest neighbors to estimate the local tangent space of anchors ( $k$ ) to be 16.

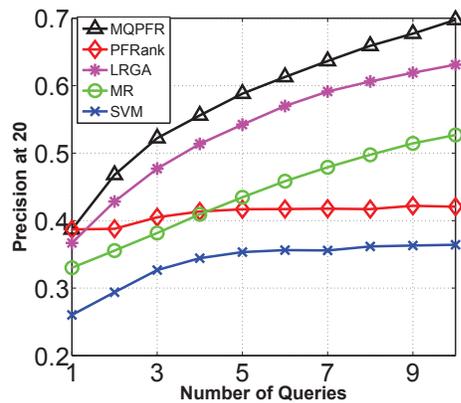
### 6.2.3. Compared Algorithms

To demonstrate how our algorithm improves the performance of multi-query image retrieval, we compared the following five algorithms.

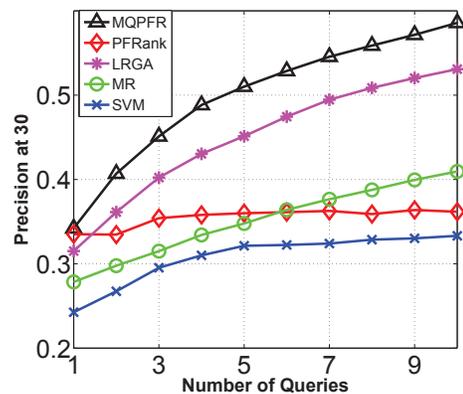
- Our proposed **MQPFR** algorithm.
- **PFRank** [15]: Parallel Field Ranking, which is the most related algorithm to ours. The major difference is that PFRank considers single-query ranking, while our algorithm considers multiple-query ranking.



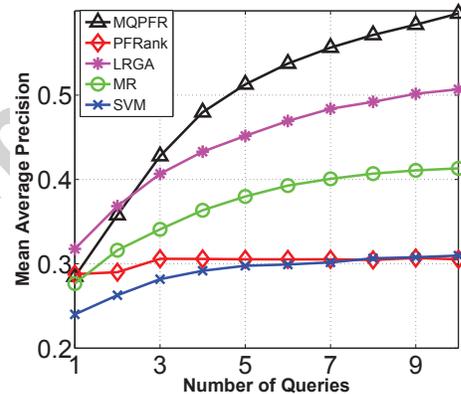
(a) Precision at top 10 results



(b) Precision at top 20 results

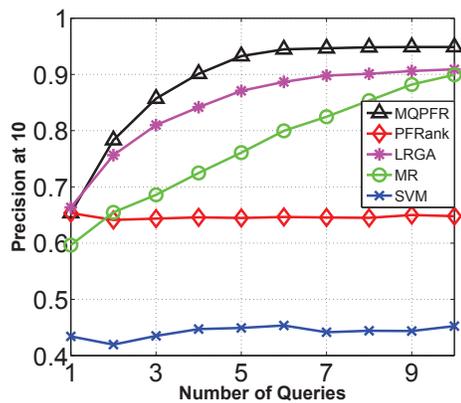


(c) Precision at top 30 results

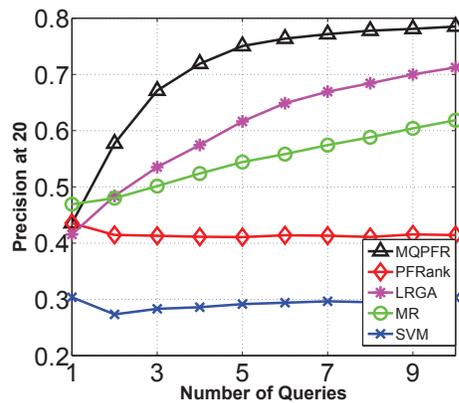


(d) Mean Average Precision

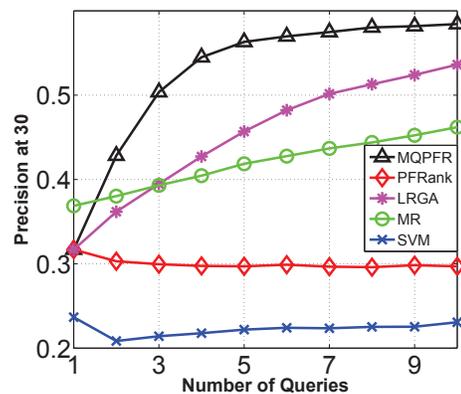
Figure 5: The retrieval precision and MAP versus the number of available queries on Corel dataset.



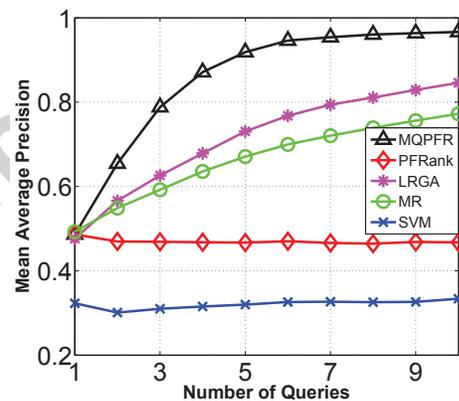
(a) Precision at top 10 results



(b) Precision at top 20 results

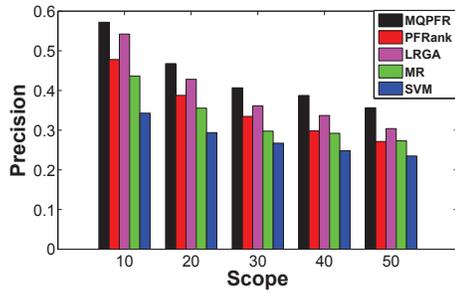


(c) Precision at top 30 results

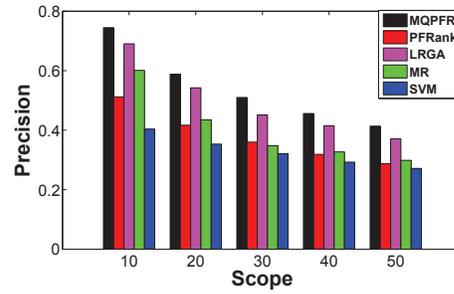


(d) Mean Average Precision

Figure 6: The retrieval precision and MAP versus the number of available queries on CMU PIE dataset.

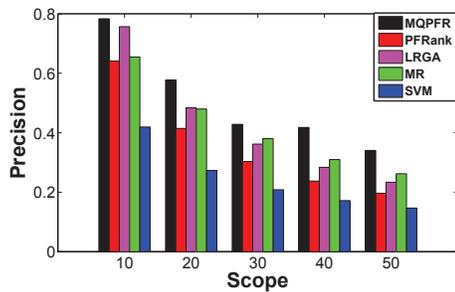


(a) Ranking with 2 queries

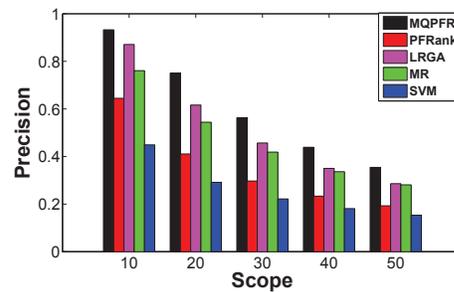


(b) Ranking with 5 queries

Figure 7: The average retrieval precision versus scope on Corel dataset. Note that, in this case all the queries are considered as initial queries and there is no relevance feedback involved.



(a) Ranking with 2 queries



(b) Ranking with 5 queries

Figure 8: The average retrieval precision versus scope on CMU PIE dataset. Note that, in this case all the queries are considered as initial queries and there is no relevance feedback involved.

Compared Algorithm	R@10	R@20	R@30	R@40	R@50
<b>MQPFR</b>	<b>0.1783</b>	<b>0.2587</b>	<b>0.3243</b>	<b>0.3727</b>	<b>0.4145</b>
PFRank	0.1271	0.1875	0.2363	0.2774	0.3111
LRGA	0.1583	0.2312	0.2822	0.3233	0.3565
MR	0.1328	0.1820	0.2253	0.2618	0.2948
SVM	0.1095	0.1697	0.2197	0.2587	0.2937

Table 1: Performance comparison using the metrics Recall (R), when 3 queries are available during the simulated feedback iterations, on Corel dataset.

Compared Algorithm	R@10	R@20	R@30	R@40	R@50
<b>MQPFR</b>	<b>0.7196</b>	<b>0.6110</b>	<b>0.5496</b>	<b>0.5011</b>	<b>0.4639</b>
PFRank	0.5668	0.4760	0.4244	0.3881	0.3586
LRGA	0.6634	0.5494	0.4809	0.4333	0.3956
MR	0.6104	0.4896	0.4323	0.3934	0.3648
SVM	0.4167	0.3648	0.3343	0.3094	0.2915

Table 2: Performance comparison using the metrics NDCG (N), when 3 queries are available during the simulated feedback iterations, on Corel dataset.

- **LRGA** [9]: Local Regression and Global Alignment, which is another state-of-the-art ranking algorithm.
- **MR** [6]: we also compare with the Manifold Ranking algorithm since we focus on learning a ranking function on the data manifold.
- **SVM** [27]: we also compare with the SVM classification method, which has been successfully applied to relevance feedback image retrieval. We use the well known LIBSVM toolbox [37] in our experiments and set data points with the lowest ranking scores as negative samples.

#### 6.2.4. Performance Evaluation

Figure 5 and Figure 6 show the retrieval precision, as well as mean average precision, as a function of the number of available queries for different algorithms on the two real datasets respectively. As it can be seen, our proposed MQPFR algorithm consistently outperforms the other four algorithms. Our algorithm performs particularly well as the number of available queries increases. This indicates that it has stronger generalization capability than

Compared Algorithm	R@10	R@20	R@30	R@40	R@50
<b>MQPFR</b>	<b>0.5066</b>	<b>0.7564</b>	<b>0.8303</b>	<b>0.8597</b>	<b>0.8698</b>
PFRank	0.3700	0.4646	0.5039	0.5290	0.5465
LRGA	0.4519	0.6013	0.6388	0.6608	0.6768
MR	0.3900	0.5354	0.6153	0.6623	0.6944
SVM	0.2372	0.3049	0.3442	0.3724	0.3992

Table 3: Performance comparison using the metrics Recall (R), when 3 queries are available during the simulated feedback iterations, on CMU PIE dataset.

Compared Algorithm	R@10	R@20	R@30	R@40	R@50
<b>MQPFR</b>	<b>0.8837</b>	<b>0.7452</b>	<b>0.6051</b>	<b>0.5051</b>	<b>0.4304</b>
PFRank	0.7239	0.5333	0.4267	0.3605	0.3147
LRGA	0.8235	0.6371	0.5022	0.4189	0.3818
MR	0.7653	0.6077	0.5080	0.4365	0.3845
SVM	0.5070	0.3743	0.3053	0.2617	0.2318

Table 4: Performance comparison using the metrics NDCG (N), when 3 queries are available during the simulated feedback iterations, on CMU PIE dataset.

the other four algorithms. Also, since PFRank and SVM essentially deal with multiple queries by simply merging them together, their performances increase very slowly when more queries are available.

Table 1, Table 2, Table 3 and Table 4 show the Recall and NDCG for different algorithms on the two real datasets when three queries are available. Again, our algorithm outperforms the other four algorithms on these metrics. Our approach performs particularly well when the scope is small. It would be important to note that, in practice the user is more interested in the top returned images. Thus, the precision with small scope is especially important.

Figure 7 and Figure 8 show the retrieval precision in the pure multi-query experiments on the two real datasets respectively. In these cases, there is no relevance feedback involved. The retrieval precision is measured within the top 10 to 50 returns for each compared algorithm. In all the cases, our algorithm performs better than the other four algorithms.

#### 6.2.5. Model Selection

Model Selection is a critical problem for most learning algorithms. Since the learning performance may vary dramatically due to different choices of the parameters in some situations. In this subsection, we explicitly carry out experiments to study the impact of the parameters on the performance of our MQPFR algorithm. The essential parameters in our algorithm are  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  which control the trade-off among the three regularization terms. In all the previous experiments, we empirically set  $\lambda_1 = \lambda_2 = \lambda_3 = 0.01$ , and in this subsection, we fix other parameters the same as before while let one of  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  vary in a wide range.

Since our algorithm mainly deals with multiple queries and more queries transmit richer user needs, we originally want to rank the data due to more queries. However, typical users will not submit too much feedback information and they easily get impatient during the long iterations. Therefore, three queries or so may be the most proper to collect major user needs while keeping the users interested. Moreover, it is appropriate to put about 20 images on a screen to show users the most relevant retrieval results in most cases. So in this subsection, we show the average retrieval precision at 20 of MQPFR with three randomly chosen queries in each category with respect to different values of  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$ . As we can see in Figure 9 and Figure 10, our algorithm is generally not very sensitive to different parameters and outperforms the other three methods over a wide range of parameters.

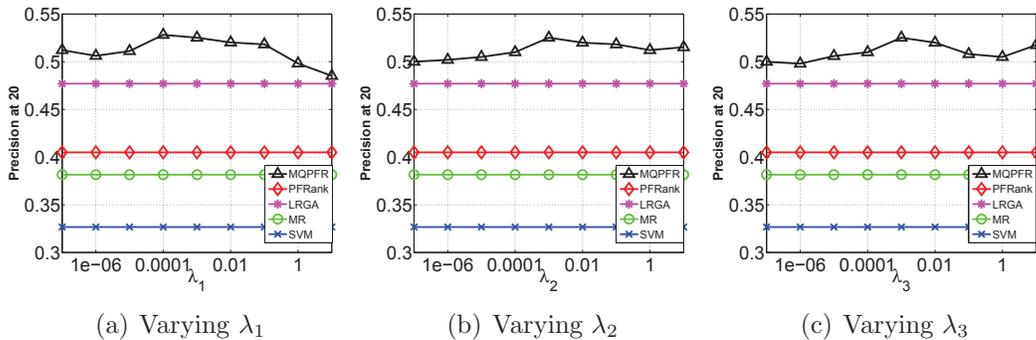


Figure 9: The average precision at top 20 images retrieved versus various parameters when 3 queries are available on Corel dataset.

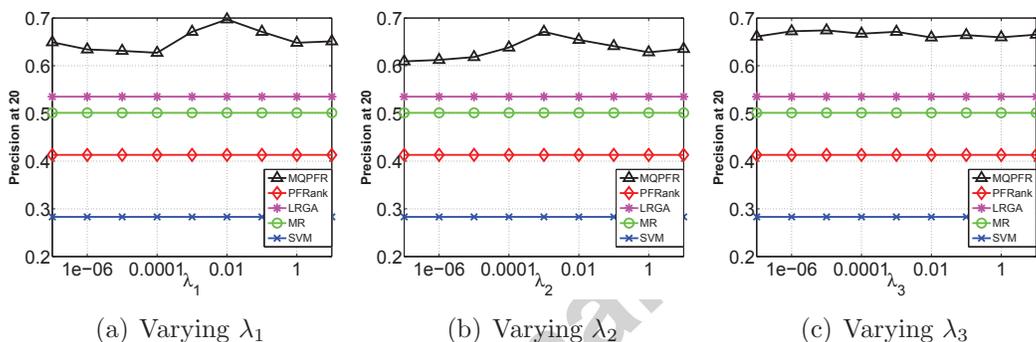


Figure 10: The average precision at top 20 images retrieved versus various parameters when 3 queries are available on CMU PIE dataset.

## 7. Conclusion

This paper introduces a novel ranking algorithm called Multi-Query Parallel Field Ranking, to enable more effective relevance feedback image retrieval. By using techniques from vector field theories, we are able to build multi-query information into a single objective function. By requiring the gradient field of the ranking function to be as parallel as possible and to point to the queries in their neighborhoods, the obtained ranking function varies linearly along the geodesics of the data manifold and achieves the highest ranking score at the multiple queries simultaneously. The experimental results on Corel and CMU PIE datasets show that our approach can significantly improve the image retrieval performance.

Our primary interest in this paper is focused on relevance feedback image

retrieval. However, our results may also be of interest to researchers in Web search and recommender systems, where the user may have multiple information needs.

## References

- [1] D. Luo, H. Huang, Ball ranking machines for content-based multimedia retrieval, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, volume 2, Barcelona, Spain, 2011, pp. 1390–1395.
- [2] Y. Rui, T. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Transactions on Circuits and Systems for Video Technology* 8 (1998) 644–655.
- [3] J. Su, W. Huang, P. Yu, V. Tseng, Efficient relevance feedback for content-based image retrieval by mining user navigation patterns, *IEEE Transactions on Knowledge and Data Engineering* 23 (2011) 360–372.
- [4] X. He, Laplacian regularized d-optimal design for active learning and its application to image retrieval, *IEEE Transactions on Image Processing* 19 (2010) 254–263.
- [5] L. Zhang, F. Lin, B. Zhang, Support vector machine learning for image retrieval, in: Proceedings of the 2001 International Conference on Image Processing, volume 2, IEEE, Thessaloniki, Greece, 2001, pp. 721–724.
- [6] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: Advances in Neural Information Processing Systems 17, Vancouver, B.C., Canada, 2003, pp. 169–176.
- [7] J. He, M. Li, H. Zhang, H. Tong, C. Zhang, Manifold-ranking based image retrieval, in: Proceedings of the 12th Annual ACM International Conference on Multimedia, ACM, New York, N.Y., USA, 2004, pp. 9–16.
- [8] Y. Yang, Y.-T. Zhuang, F. Wu, Y.-H. Pan, Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval, *IEEE Transactions on Multimedia* 10 (2008) 437–446.
- [9] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A multimedia retrieval framework based on semi-supervised ranking and relevance feedback, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 723–742.

- [10] X. He, W.-Y. Ma, H.-J. Zhang, Learning an image manifold for retrieval, in: Proceedings of the 12th Annual ACM International Conference on Multimedia, ACM, New York, N.Y., USA, 2004, pp. 17–23.
- [11] J. Yu, Q. Tian, Learning image manifolds by semantic subspace projection, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, ACM, Santa Barbara, CA, USA, 2006, pp. 297–306.
- [12] J. Tenenbaum, V. De Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (2000) 2319–2323.
- [13] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [14] B. Lin, C. Zhang, X. He, Semi-supervised regression via parallel field regularization, in: Advances in Neural Information Processing Systems 25, Granada, Spain, 2011, pp. 433–441.
- [15] M. Ji, B. Lin, X. He, D. Cai, J. Han, Parallel field ranking, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012, pp. 723–731.
- [16] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Puerto Rico, 1997, pp. 762–768.
- [17] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking, *The Journal of Machine Learning Research* 11 (2010) 1109–1135.
- [18] X. Bai, X. Yang, L. J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010) 861–874.
- [19] X. He, M. Ji, H. Bao, Graph embedding with constraints, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 2009, pp. 1065–1070.
- [20] X. Zhou, M. Belkin, N. Srebro, An iterated graph laplacian approach for ranking on manifolds, in: Proceedings of the 17th ACM SIGKDD

- International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 2011, pp. 877–885.
- [21] B. Wang, F. Pan, K. Hu, J. Paul, Manifold-ranking based retrieval using  $k$ -regular nearest neighbor graph, *Pattern Recognition* 45 (2012) 1569–1577.
  - [22] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, J. Luo, Efficient manifold ranking for image retrieval, in: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Beijing, China, 2011, pp. 525–534.
  - [23] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: *Proceedings of the IEEE 11th International Conference on Computer Vision*, IEEE, Rio de Janeiro, Brazil, 2007, pp. 1–7.
  - [24] Y.-T. Zhuang, Y. Yang, F. Wu, Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval, *IEEE Transactions on Multimedia* 10 (2008) 221–229.
  - [25] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: *Proceedings of the 9th ACM International Conference on Multimedia*, ACM, Ottawa, ON, Canada, 2001, pp. 107–118.
  - [26] X. S. Zhou, T. S. Huang, Relevance feedback in image retrieval: A comprehensive review, *Multimedia systems* 8 (2003) 536–544.
  - [27] C. Manning, P. Raghavan, H. Schütze, *Introduction to information retrieval*, volume 1, Cambridge University Press Cambridge, 2008.
  - [28] P. Petersen, *Riemannian geometry*, Springer New York, 2006.
  - [29] W. Liu, J. He, S. Chang, Large graph construction for scalable semi-supervised learning, in: *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 679–686.
  - [30] G. Golub, C. Van Loan, *Matrix computations*, volume 3, Johns Hopkins University Press, 1996.
  - [31] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: *Advances in Neural Information Processing Systems 23*, Vancouver, B.C., Canada, 2009, pp. 2223–2231.

- [32] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986) 679–698.
- [33] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Transactions on Computers* 42 (1993) 300–311.
- [34] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition* 29 (1996) 51–59.
- [35] T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression database, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003) 1615–1618.
- [36] G. Chowdhury, *Introduction to modern information retrieval*, Facet publishing, 2010.
- [37] C. Chang, C. Lin, Libsvm: a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* 2 (2011) 27.