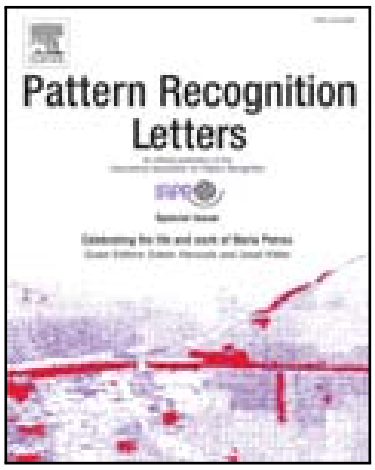


Image Retrieval based on Image-to-Class Similarity

Jun Chen, Yong Wang, Linbo Luo, Jin-Gang Yu, Jiayi Ma

PII: S0167-8655(16)00029-5
DOI: 10.1016/j.patrec.2016.01.017
Reference: PATREC 6440

To appear in: *Pattern Recognition Letters*

Received date: 5 November 2015
Accepted date: 22 January 2016

Please cite this article as: Jun Chen, Yong Wang, Linbo Luo, Jin-Gang Yu, Jiayi Ma, Image Retrieval based on Image-to-Class Similarity, *Pattern Recognition Letters* (2016), doi: 10.1016/j.patrec.2016.01.017

Research Highlights (Required)

To create your highlights, please type the highlights against each `\item` command.

It should be short collection of bullet points that convey the core findings of the article. It should include 3 to 5 bullet points (maximum 85 characters, including spaces, per bullet point.)

- We introduce the concept of image-to-class similarity.
- The new similarity is able to exploit intrinsic properties of the query class.
- We use the similarity to develop a retrieval method.
- The method has linear complexity and can outperform the state-of-the-art.
- We extend the retrieval method based on max-pooling used in visual recognition.

Pattern Recognition Letters
journal homepage: www.elsevier.com

Image Retrieval based on Image-to-Class Similarity

Jun Chen[a], Yong Wang[b], Linbo Luo[b], Jin-Gang Yu[c,**], Jiayi Ma[d]

[a]*School of Automation, China University of Geosciences, Wuhan 430074, China*
[b]*Faculty of Mechanical & Electronic Information, China University of Geosciences, Wuhan 430074, China*
[c]*Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA*
[d]*Electronic Information School, Wuhan University, Wuhan, 430072, China*

ABSTRACT

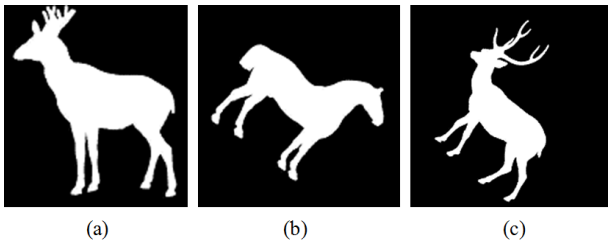
Similar image/shape retrieval has attracted increasing interests in recent years. A typical strategy of existing retrieval algorithms is to rank the images according to the image-to-image similarities, e.g., the similarities between the query image and the images in the database. This strategy ignores the inherent information of the class that the query image belongs to (we call it query class). To address this issue, rather than using image-to-image similarity, we propose a simple yet effective retrieval method based on exploring the image-to-class similarity. The method uses an iterative framework, where the size of the query class is progressively enlarged according to the previous retrieval results, and the ranked list is generated according to the similarities between the images in the database and the query class. This framework enables us to explore the inherent information of the query class, and hence helps to improve the retrieval accuracy. Experimental results on various datasets demonstrate that our method is able to effectively improve the image and shape retrieval accuracy compared to state-of-the-art methods.

Keywords: Image retrieval, shape retrieval, matching, image-to-class similarity

1. Introduction

This paper considers the task of similar image/shape retrieval, which is a critical problem in many computer vision and pattern recognition tasks [1, 29, 26, 12, 34, 20]. Given a query image, the goal is to retrieve the images of the same object or scene from a large database and return a ranked list.

The most common types of visual information are color, texture and shape information, and they are widely studied and applied to many application [4, 3, 37, 21, 13, 14, 24, 18], especially for measuring the similarity between images in retrieval systems. Classical retrieval techniques use two types of visual features: local and global features. Local features, such as SIFT [22] for images and shape context [6] for shapes, typically focus on local texture and shape. While global features such as global geometrical constraints constrain that the local features satisfy spatial coherence [42, 33, 25, 27].



(a) (b) (c)

Fig. 1. Three shape examples coming from the MPEG-7 database [5]: (a) a shape of deer; (b) a shape of horse; (c) a shape of deer. Existing shape similarity methods which are based on image-to-image similarity typically rank shape (b) as more similar to the input shape (a) than shape (c).

Existing image/shape retrieval methods are typically based on sorting of the similarities between the query image and other images, which just consider the pairwise image similarities, i.e., the *image-to-image similarity* [35, 16, 17, 32, 19]. This seems

**Corresponding author
e-mail: junchen@cug.edu.cn (Jun Chen), wycug2011@gmail.com (Yong Wang), luolb@ige-live.com (Linbo Luo), jgang.yu@gmail.com (Jin-Gang Yu), jyma2010@gmail.com (Jiayi Ma)

to be reasonable, since two similar images in general will have small difference which can be measured by some distance function [5]. However, it ignores a fact that some differences are more relevant while other differences are less relevant for image/shape similarity. For example, two images of different objects with the same background are likely to be classified into the same class, because the background probably plays a important role in calculating similarity if only the image-to-image similarity is considered. In this case, a similarity that can capture the intrinsic property of the query class is desired, and this is the very goal in this paper.

We take a shape retrieval task for further explanation, as shown in Fig. 1. Using an image-to-image similarity based method such as Inner Distance Shape Context [19] (IDSC), shape (b) has larger similarity with the query shape (a) compared to shape (c). However, shapes (a) and (c) are both a deer, while shape (b) is a horse. The reason of such incorrect result is that the posture of horse (b) is much more similar to that of deer (a) and hence largely increases the similarity. Unlike human beings, such retrieval algorithm views each shape as a single pattern and cannot capture the intrinsic property of a deer, for example, the antler. To address this issue, in the proposed approach, we introduce the concept of *image-to-class similarity*, which is measured based on the similarity between an image and a class, and the class does not need to be known in advance. A typical example is given in Fig. 2. Given a query shape in the first column, we first test the Coherent Spatial Relations [8, 7] (CSR) matching method and the IDSC combined with Dynamic Programming [19] (IDSC + DP) method for shape retrieval, as shown in the first two rows. We see that these methods using an image-to-image similarity can find a few of the most similar shapes, while the latter retrieved shapes tend to be wrong, as marked by the red boxes. In our approach, we retrieve the query shape using the proposed image-to-class similarity, and the ten retrieved shapes all come from the same class as the query shape, as shown in the last row. Therefore, the proposed method can effectively improve the retrieval results.

The basic idea of our approach is as follows. When we need to retrieve multiple examples of a query image, we can progressively generate a set of retrieved images based on the similarity between the images in the database and the query class, i.e., the image-to-class similarity. The query class is composed of the query image and the current retrieved images, while the image-to-class similarity is defined as the sum of the similarities between an image in the database and each image in the query class. The method uses an iterative framework, which alternatively updates the query class and computes the image-to-class similarity. In the first iteration, we have little information about the query class, and hence the query class only contains the query image. Once we have found the most similar image, we then merge it into the query class, and find the next most similar image. As the iteration proceeds, the query class will contain more samples, and a new image regarded as a positive sample should be similar to all the samples in the query class, which means that the new image should contain some intrinsic properties of the query class. Therefore, this framework enables us to explore the inherent information of the query class, and hence helps to improve the retrieval accuracy.

Our contribution in this paper includes the following two aspects. (i) We introduce the concept of image-to-class similarity, where the similarity not just relates to the query image, but relies on the whole query class that the query image belongs to. (ii) We propose a simple yet effective retrieval algorithm based on the image-to-class similarity, which can significantly improve the retrieval accuracy. (iii) We give an extension of the proposed method based on the max-pooling used in visual recognition. (iv) The proposed method has linear time complexity which can be used in large scale database.

2. Related Work

Our proposed method exploits the intrinsic properties of the query class by a post-processing procedure. Recently, there exists a growing interest in providing post-processing procedures by exploiting the contextual information to improve retrieval performance. These methods in general aim at capturing the geometry of the underlying manifold among all elements of the database.

In [5], a typical algorithm called Graph Transduction (GT) was proposed, which can learn context-sensitive similarity in semi-supervised settings by considering the query itself as the only labeled data. It further improves GT in [38] by using shortest path propagation such that redundant context is removed. As one of the most popular branch, diffusion process performs random walk in the feature manifold to spread the similarities in an iterative manner. The generic framework of diffusion process is summarized in [9], including some widely-used variants, such as locally constrained diffusion process (LCDP) [40]. Meanwhile, some methods mange to refine the similarity measure without iterative diffusion. Kontschieder *et al*. proposed a modified mutual kNN graph as the underlying representation and demonstrated its performance for the task of shape retrieval [15]. Egozi *et al*. introduced a shape meta-similarity measure [10], and it is able to agglomerate pairwise shape similarities and improve the retrieval accuracy. Moreover, Bai *et al*. proposed an effective retrieval method called Neighbor Set Similarity, and they have presented promising retrieval performances which are superior to diffusion process [2], but at a much lower time complexity.

In this paper, rather than capturing the manifold geometry among all elements of the database, we focus on the query class and exploit its intrinsic properties. Nevertheless, our method can use all the above mentioned methods for initialization, for example, using the pairwise similarities generated by these methods as inputs, and then further improves the retrieval performance.

3. The Proposed Algorithm

In this section, we first review the classical image-to-image similarity, and then describe the mathematic formulation and solution of our proposed method, followed by an extension of our method, and we finally analyze the computational complexity.

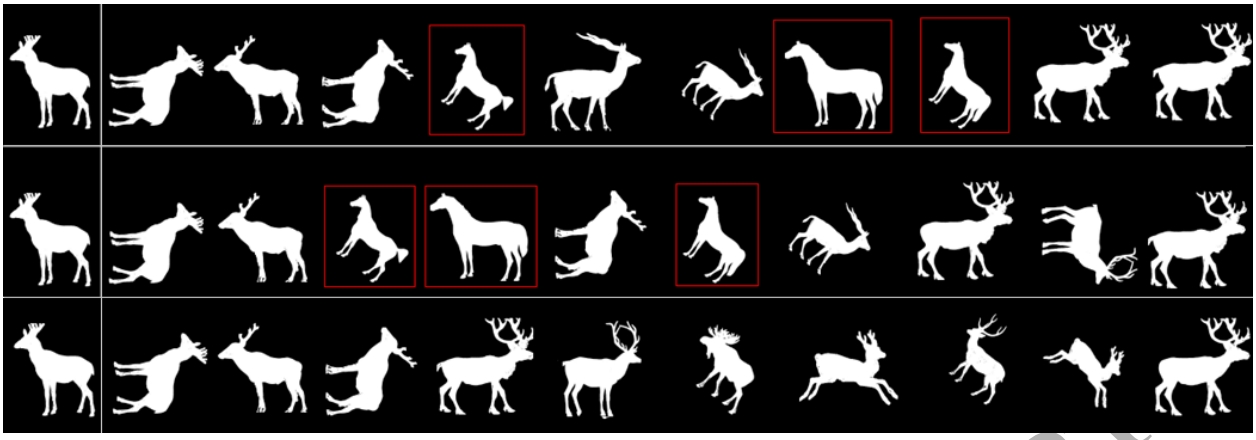


Fig. 2. In each row, the first shape is the query shape, and the rest ten shapes are the ten most similar shapes retrieved by a certain algorithm which is listed in descending order. The first two rows are results of CSR [7] and IDSC + DP [19] based on image-to-image similarity, respectively. The third row shows the results of our method based on image-to-class similarity. Red boxes indicate false results.

3.1. Image-to-Image Similarity Measure

We first describe the classical image-to-image similarity retrieval. It has been applied to many retrieval scenarios like key word, document, image, and shape retrieval. The given is an image database $\mathcal{I} = \{I_i : i \in \mathbb{N}_N\}$ and a similarity function S: $I \times I \to \mathbb{R}^+$ that assigns a positive similarity value to each pair of images based on, for example, image matching [19, 5, 23]. Therefore, we obtain a similarity matrix $\mathbf{P}$, where $\mathbf{P}_{ij} = S(I_i, I_j)$ is the similarity between I_i and I_j.

We assume that I_1 is a query image, and $\{I_2, I_3, \ldots, I_N\}$ is a set of known database images. Then, by sorting the values $\mathbf{P}_{1i}$ in decreasing order for $i = 2, 3, \ldots, N$, we obtain a ranking of database images according to their similarities to the query, i.e., the most similar database image has the highest value and is listed first. Usually, the first M ($M \ll N$) images are returned as the most similar to the query I_1.

3.2. Problem Formulation

In general, the similarity function S is not perfect, and for many pairs of images, it returns wrong results, although it may return correct scores for many pairs. This motivates us to investigate the image-to-class similarity: the retrieved images should not just be similar to the query images, but be similar to all the images in the unknown class that the query image belongs to, i.e., the query class (to be determined). This is equivalent to solving the following problem: given I_t as the query image, the goal is to find the M images that are not only similar to the query image I_t, but also similar to each other. Let $\{x_i : i \in \mathbb{N}_N\} \in \{0, 1\}$ be a set of indicator variables. Thus the goal is to maximize the following optimization problem:

$$\{x_i^* : i \in \mathbb{N}_N\} = \arg \max_{x_1, \cdots, x_N} \sum_{i=1}^{N} x_i \mathbf{P}_{ti} + \lambda \sum_{i=1, i\neq t}^{N} \sum_{j=i+1, j\neq t}^{N} x_i x_j \mathbf{P}_{ij}, \tag{1}$$

$$\text{s.t. } \{x_i : i \in \mathbb{N}_N\} \in \{0, 1\}, \text{ and } \sum_{i=1}^{N} x_i = M,$$

where the first term in Eq. (1) is the similarity between the query image and the retrieved images, the second term is the similarity of the retrieved images themselves, λ is a positive number controlling the trade-off between the two terms. Clearly, smaller value of λ (e.g., $\lambda < 1$) indicates greater weight to the query image I_t. We denote L_M the index set of the optimal solution, i.e., $L_M = \{i | x_i^* = 1, i \in \mathbb{N}_N\}$. Note that there is a constraint that should be added to ensure objectivity: $L_M \subset L_{M+1}$, which means that as the retrieval number increases, the later results should contain all the images in the former results. Therefore, our objective function becomes

$$\{x_i^* : i \in \mathbb{N}_N\} = \arg \max_{x_1, \cdots, x_N} \sum_{i=1}^{N} x_i \mathbf{P}_{ti} + \lambda \sum_{i=1, i\neq t}^{N} \sum_{j=i+1, j\neq t}^{N} x_i x_j \mathbf{P}_{ij}, \tag{2}$$

$$\text{s.t. } \{x_i : i \in \mathbb{N}_N\} \in \{0, 1\}, \ \sum_{i=1}^{N} x_i = M, \text{ and } L_m \subset L_{m+1}, \ \forall m < M.$$

3.3. Optimization

The Optimization problem in Eq. (2) can be solved using a recursive method.

For $M = 1$, the second term in the objective function is gone, and the solution is obvious, i.e., $x_t = 1$. Therefore, we have $L_1 = \{t\}$.

For $M = 2$, the second term is also gone, and solution is $L_2 = \{t, i\}$, where i satisfies that P_{ti} has the largest value, for $i \in \mathbb{N}_N, i \neq t$.

For $M > 2$, since $L_{M-1} \subset L_M$, the former $M - 1$ images to be retrieved have been determined, and hence we only need to find the optimal M-th image. Therefore, the objective function in Eq. (2) becomes

$$j^* = \arg \max_{j \notin L_{M-1}} \mathbf{P}_{tj} + \lambda \sum_{i \in L_{M-1}, i\neq t} \mathbf{P}_{ij}. \tag{3}$$

The optimal value of j, i.e. j^*, is the index of the image that has the largest similarity with respect to the query image, i.e. $\mathbf{P}_{tj}$, plus the images already retrieved with a weight λ, i.e.,

Algorithm 1: The RICS Algorithm

Input: Similarity matrix $\mathbf{P}$, index of query image t, parameter λ, required number of retrieved images M

Output: Index set of M retrieved images L_M

1 Initialize $L_M = \{t\}$, $\mathbf{p} = \mathbf{P}_{\cdot,t}$, $k = 1$;
2 **repeat**
3 Find the index j, where $j \notin L_M$ and $\mathbf{p}_j$ has the maximum value;
4 Update $L_M \leftarrow L_M \cup j$;
5 Update $\mathbf{p} \leftarrow \mathbf{p} + \lambda \mathbf{P}_{\cdot,j}$;
6 $k = k + 1$;
7 **until** $k = M$;
8 The index set L_M is obtained after the iteration stops.

$\lambda \sum_{i \in L_{M-1}} \mathbf{P}_{ij}$. Therefore, $I_M = I_{M-1} \cup j^*$. This process proceeds until M reaches the predefined value.

Now, we analyze the solving process. Denote $\mathbf{p}$ an $N \times 1$ dimensional vector, $\{i_n : n \in \mathbb{N}_M\}$ a set of indexes corresponding to the retrieved images, i.e. $L_m = \{i_1, \ldots, i_m\}$. For $M = 1$, the retrieved image is obviously the query image I_t itself, i.e., $i_1 = t$. We then assign $\mathbf{p} = \mathbf{P}_{\cdot,i_1}$ with $\mathbf{P}_{\cdot,k}$ being the k-th column of $\mathbf{P}$. For $M = 2$, the index i_2 of the retrieved image should be j, where $j \notin L_1$ and $\mathbf{p}_j$ has the maximum value. We then assign $\mathbf{p} = \mathbf{p} + \lambda \mathbf{P}_{\cdot,i_2}$. For $M = 3$, the index i_3 of the retrieved image should be j, where $j \notin L_2$ and $\mathbf{p}_j$ has the maximum value. We then assign $\mathbf{p} = \mathbf{p} + \lambda \mathbf{P}_{\cdot,i_3}$. As this process proceeds, we finally get the indexes of all the retrieved images $L_M = \{i_1, \ldots, i_M\}$. Note that this is an iteration process. By defining a *query class* as the set of current retrieved images, in the k-th iteration, we first find the k-th image based on the vector $\mathbf{p}$ and update the query class, and then calculate the vector $\mathbf{p}$. Here the element of $\mathbf{p}$, e.g. $\mathbf{p}_j$ with $j \notin L_{k-1}$, can be seen as the similarity of image I_j and the query class L_{k-1}. We define this similarity as the *image-to-class* similarity.

Since our Retrieval algorithm is based on Image-to-Class Similarity, we name it *RICS* for short. We summarize our RICS algorithm in Alg. 1.

3.4. Diagram Model of The Proposed Method

We next give an intuitive explanation of our algorithm. We use a state transition diagram to represent the relation of images in the database, as shown in Fig. 3. The database contains five images, where the first image is the query image. Each image (e.g., *1, 2, 3, 4, 5*) is a node in the diagram, and the probability of transit from node i to node j can be expressed as the similarity of the corresponding two images, e.g. $\mathbf{P}_{ij}$, here $\mathbf{P}_{ij} = \mathbf{P}_{ji}$ and the normalized similarity matrix is adopted $\mathbf{P}'_{ij} = \frac{\mathbf{P}_{ij}}{\sum_{k=1}^{N} \mathbf{P}_{ik}}$. The fully connected state diagram can then be used to indicate similarity relations of all the images in the database.

The left figure of Fig. 3 is the initial status of the state transition diagram, where node 1 itself is the first retrieved images. Then we aim to find the maximum value of the set $\{\mathbf{P}_{21}, \mathbf{P}_{31}, \mathbf{P}_{41}, \mathbf{P}_{51}\}$. Suppose it is $\mathbf{P}_{12}$, then node 2 is merged into the query class, as shown in the middle figure of Fig. 3. Next we update the image-to-class similarity, such as $\mathbf{P}_{3,12} = \mathbf{P}_{31} + \lambda \mathbf{P}_{32}$. After that, we again find the maximum value of the set $\{\mathbf{P}_{3,12}, \mathbf{P}_{4,12}, \mathbf{P}_{5,12}\}$. Suppose it is $\mathbf{P}_{5,12}$, then node 5 is merged into the query class, as shown in the right figure of Fig. 3. As this iteration proceeds, the sizes of the query class increases, and we gradually obtain all the required images.

3.5. Extension: Sum-Pooling and Max-Pooling for Similarity

In our RICS algorithm, the image-to-class similarity is defined using the sum-pooling rule, i.e., Eq. (3), which is the sum of the similarity between an image in the database and all the retrieved images in the query class. Alternatively, it can also be defined using the max-pooling rule proposed in visual recognition [39], such as the maximum similarity between an image in the database and one of the retrieved images in the query class. Then Eq. (3) becomes

$$j^* = \arg \max_{j \notin L_{M-1}} \{\mathbf{P}_{tj}, \lambda \mathbf{P}_{ij} : i \in L_{M-1}, i \neq t\}. \quad (4)$$

In general terms, the objective of similarity pooling is to transform the joint image similarity into a new, more usable one that preserves important information.

3.6. Computational Complexity

As can be seen in Alg. 1, for a query image with a given pairwise similarity matrix $\mathbf{P}$, our RICS algorithm requires M iterations to retrieve M images. In each iteration, it requires to find the maximum value of a vector in Line 3 which at most has N elements. Its time complexity is $O(N)$. Clearly, update of the image-to-class similarity $\mathbf{p}$ in Line 5 also has time complexity $O(N)$. Therefore, the total time complexity of our RICS is $O(MN)$, which is linear with respect to the scale of the database. Note that to compute and store the similarity matrix $\mathbf{P}$, it requires $O(N^2)$ both in time and in space. However, for retrieving a certain image, it only requires to compute M columns of $\mathbf{P}$[1] and hence, the complexity of computing the pairwise similarities decreases to $O(MN)$ both in time and in space. Therefore, the complexity of the whole procedure is $O(MN)$ both in time and in space. For our RICS with max-pooling, the time and space complexities remain the same, since Eq. (4) also has time complexity $O(N)$.

4. Experimental Results

In this section, we evaluate our proposed RICS on various databases for image and shape retrieval. Throughout the experiments, the initial image-to-image similarity matrix $\mathbf{P}$ is obtained by the IDSC method [19]. In the following, we first discuss the databases considered in this paper, and then test the performance of our method on each database and compare it to several state-of-the-art algorithms.

[1]The similarity matrix $\mathbf{P}$ here is computed based on some off-the-shelf method. If the pairwise similarities are not computed independently, for example, the chosen method itself involves a post-processing procedure to rescore the pairwise similarities based on all elements of the database, then it probably cannot obtain the M required columns of $\mathbf{P}$ in $O(MN)$ time space.

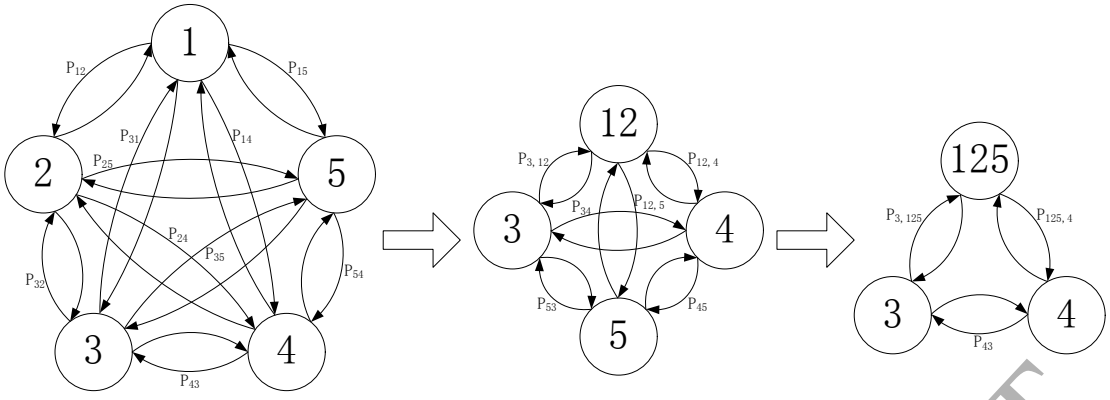


Fig. 3. Iterative update process of our RICS method. The database contains five images, and the query image is node 1. Left: initial similarity relations represented by fully connected state transition diagram. Middle: new state transition diagram after the first iteration, node 12 represents the current query class. Right: result of the second iteration, where node 5 has the largest image-to-class similarity with node 12, and hence is merged into the query class.

4.1. Databases

We test our RICS on three databases: MPEG-7 shape database [5], Nistér and Stewénius (N-S) database [30], and AT&T face database [31].

MPEG-7 consists of 1400 shapes, which are divided into 70 classes and each class has 20 shapes. The retrieval rate is measured by the so-called bulls-eye score [5]. Every shape in the database is compared to all other shapes, and the number of shapes from the same class among the 40 most similar shapes is reported. The bulls-eye retrieval rate is defined as the ratio of the total number of shapes from the same class to the highest possible number, i.e., 20. The best possible rate is 100 percent.

N-S database contains 10200 natural images which are collected by Henrik Stewénius and David Nistér. The database has 2550 classes, and each class has 4 images which are the picture of the same scene from the different angle.

AT&T face database contains 400 face images. There are 40 groups, each group has 10 images os the same people's face. The faces are different in the photo time, light condition, expression (open eyes, close eyes, laugh, cry, angry), details(glasses or no glasses), and direction.

4.2. Results on MPEG-7 Database

To compute the similarity matrix by using the CSR shape matching method, we first estimate the pairwise shape distance as the weighted sum of three terms, e.g., shape context distance, bending energy, and outlier ratio. The first two terms can be referred to [6] for details, and we also added an additional outlier ratio term to discourage the unmatched parts of shapes. The pairwise distances are then converted into a similarity matrix by using the strategy in [5].

Based on the similarity matrix of the 1400 shapes generated by CSR, we test the retrieval accuracy of our RICS algorithm. Fig. 4 demonstrates a part of the retrieval results presented in every two rows. In each group, the first row is the result obtained by directly sorting the image-to-image similarities, such as the CSR method, while the second row is the result of our RICS. From the results, we can see that our RICS algorithm is able to rectify many false retrieval results obtained by the original CSR method, especially for the apple shapes in the 15-th row. Although there are many false shapes by directly sorting the image-to-image similarities, we can get all the correct shapes after using our image-to-class similarity. This illustrates that the image-to-class similarity does make sense, and can promote the retrieval accuracy.

There is only one parameter in our method, i.e., λ. Here we test its influence on the retrieval accuracy on the database. The average accuracy statistics are given in Fig. 5, we see that our RICS with both sum-pooling (RICS-SP) and max-pooling (RICS-MP) can achieve the best performance around $\lambda = 0.7$. Thus we fix $\lambda = 0.7$ in the rest of our experiments.

To give a quantitative comparison on this database, we compare our RICS-SP and RICS-MP to several state-of-the-art shape matching methods, such as Curvature Scale Space (CSS) [28], SC+TPS [36], IDSC+DP [19], Shape Tree (ST) [11], and CSR [7]. Besides, we also use some post-processing methods such as LCDP [40], GT [5], Meta Descriptor (MD) [10], and Generic Diffusion Process (GDP) [9], where the result of IDSC is also used as input for these methods. The results are summarized in Table 1, we see that our RICS-SP and RICS-MP perform better than all the shape matching methods, and the result of RICS-SP is slightly better than that of RICS-MP. Among the post-processing methods, our RICS-SP can achieve comparable accuracy. However, our method has linear complexity with respect to the scale of the database. Here we present the average runtime for retrieving a certain query, as shown in the last column in Table 2. Clearly, the time cost of our RICS is much less than the other post-processing methods. The experiments are performed on a laptop with 3.3 GHz Intel Core CPU, 8 GB memory and Matlab Code.

Since our RICS is a post-processing method, it can uses the similarity matrix produced by any method, including a post-processing, as input to further improve the retrieval performance. To justify this issue, we use the result of GT [5] as input, and the retrieval accuracy is increased from 91.61% to 93.14%. That is to say, our RICS can be taken as a complement for existing methods to improve the retrieval performance.

Table 1. Comparison of different shape matching methods on the MPEG-7 database.

	CSS [28]	SC+TPS [36]	IDSC+DP [19]	ST [11]	CSR [7]	LCDP [40]	GT [5]	MD [10]	GDP [9]	RICS-SP	RICS-MP
Acc.	75.44%	76.51%	85.40%	87.70%	85.37%	92.36%	91.61%	91.46%	91.12%	91.98%	90.36%

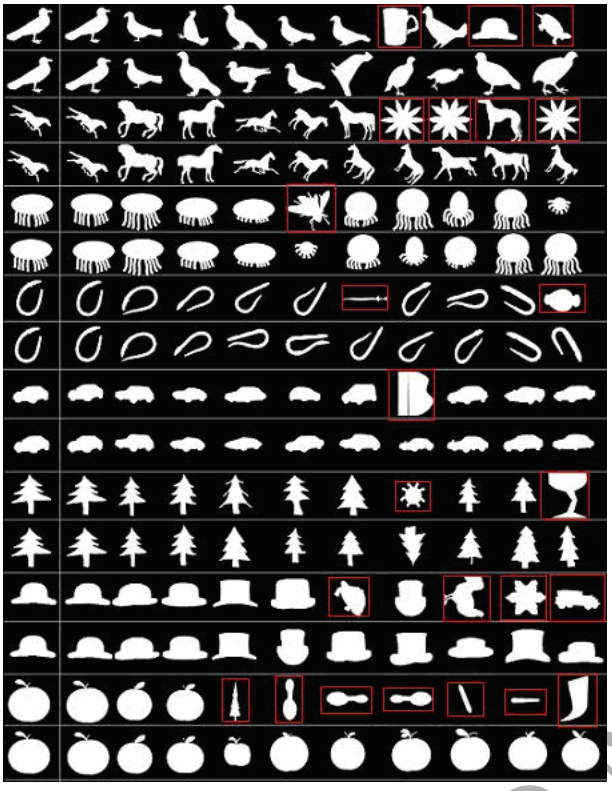


Fig. 4. Shape retrieval results on the MPEG-7 database presented presented in every two rows. In each group, the first row is the result of CSR obtained by directly sorting the image-to-image similarities, while the second row is the result of our RICS obtained based on the image-to-class similarity. In each row, the first image is the query image.

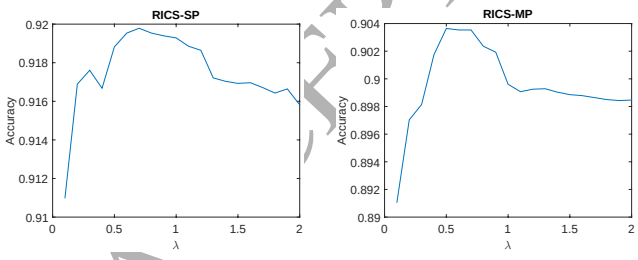


Fig. 5. Performance of our method RICS-SP (left) and RICS-MP (right) on the MPEG-7 database with different values of λ.

Table 2. Run time comparison of different post-processing methods on the MPEG-7 database.

	LCDP [40]	GT [5]	MD [10]	GDP [9]	RICS
Time (ms)	55.14	1600	1400	0.81	0.29

Fig. 6. Image retrieval results on the N-S database presented in every two rows. In each group, the first row is the result of CSR obtained by directly sorting the image-to-image similarities, while the second row is the result of our RICS-SP obtained based on the image-to-class similarity. In each row, the first image is the query image.

4.3. Results on N-S Database

Next, we consider image retrieval on the N-S database. There are 4 images in each class of the database. For each query image, we use a retrieval algorithm to find 4 images with the largest similarity to the query image. The retrieval precision is measured by the average number of correct images in the top 4 images returned. Thus, the best score is 4. There are only 4 images in each class, which makes the dataset very challenging for exploiting the intrinsic properties of a query class.

To compute the pairwise similarity, we first use the SIFT algorithm [22] to establish feature correspondences between the image pair, and then use the CSR method to remove false correspondences. Finally, the pairwise similarity is assigned by the number of preserved correspondences by CSR.

Fig. 6 present a part of the retrieval results presented in every two rows. In each group, the first row is the result obtained by directly sorting the image-to-image similarities, such as the CSR method, while the second row is the result of our RICS-SP. From the results, we see that the retrieved images of the CSR method contain several false images, such as the vase class and the car class. Since there are lots of images in the database, there may exist a false image with some parts matching with the query image which makes high similarity between them. Nevertheless, the probability of one false image having good match with all the images in the query class is in general very small. Therefore, based on the image-to-class similarity, the retrieval accuracy is increased by our RICS-SP method. As we can see in the figure, our RICS-SP method is able to retrieve all the correct images.

To provide a quantitative evaluation, we use three state-of-

Table 3. Comparison of accuracies on the N-S dataset.

	CSR [7]	LCDP [40]	TPG [41]	RICS
Acc.	3.45	3.58	3.61	3.65

the-art methods for comparison, such as CSR, LCDP, and Tensor Product Graph (TPG) [41]. The average retrieval accuracies are reported in Table 3. Clearly, our RICS has the best accuracy.

4.4. Results on AT&T Face Database

There are 10 images in each class in the AT&T face database. For a query image, we need to find all the 10 images in the same class. As on the N-S database, we also use the average number of correct images in the top 10 returned images to measure the retrieval accuracy, where the best score is 10. Meanwhile, the pairwise similarity between an image pair is also computed as the same as on the N-S database. Fig. 7 presents some of the retrieval results of the CSR (i.e. the odd rows) and our RICS-MP (i.e. the even rows) methods. The false images are highlighted with red (CSR) or blue (RICS-MP) boxes.

We see that in the first and fifth rows of Fig. 7, only three correct images are retrieved by using CSR which only consider the image-to-image similarity. This is no surprise, since in case of different expression, direction, light condition, and so on, it is hard to match all the ten images to the query image. However, using our RICS-MP based on the image-to-class similarity, nearly all the ten correct images are retrieved. This is appropriate, since it is possible to find one image in the query class that can match well to a correct image ready to be retrieved. Therefore, the max-pooling rule in our RICS-MP can generate a more appropriate similarity measure to improve the retrieval accuracy. The average retrieval accuracy of CSR and our RICS-MP are 7.24 and 8.76, respectively.

5. Conclusion

Within this paper, we proposed a simple yet effective retrieval method named RICS based on image-to-class similarity. The method alternatively updates the query class and computes the image-to-class similarity with a linear time complexity. We also provide an extension of RICS by using the max-pooling rule in visual recognition. The quantitative comparisons on both shape retrieval and image retrieval tasks demonstrate that the proposed method can yield results outperform many state-of-the-art algorithms.

Acknowledgements

The authors gratefully acknowledge the financial supports from the National Natural Science Foundation of China under Grant Nos. 61503288 and 41202232, and the China Postdoctoral Science Foundation under Grant No. 2015M570665.

Fig. 7. Image retrieval results on the AT&T face database presented in every two rows. In each group, the first row is the result of CSR obtained by directly sorting the image-to-image similarities, while the second row is the result of our RICS-MP obtained based on the image-to-class similarity. In each row, the first image is the query image.

References

[1] Bai, S., Bai, X., Liu, W., Roli, F., 2015a. Neural shape codes for 3d model retrieval. Pattern Recognition Letters 65, 15–21.

[2] Bai, X., Bai, S., Wang, X., 2015b. Beyond diffusion process: Neighbor set similarity for fast re-ranking. Information Sciences .

[3] Bai, X., Wang, B., Yao, C., Liu, W., Tu, Z., 2012. Co-transduction for shape retrieval. IEEE Transactions on Image Processing 21, 2747–2757.

[4] Bai, X., Yang, X., Latecki, L.J., 2008. Detection and recognition of contour parts based on shape similarity. Pattern Recognition 41, 2189–2199.

[5] Bai, X., Yang, X., Latecki, L.J., Liu, W., Tu, Z., 2010. Learning context-sensitive shape similarity by graph transduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 861–874.

[6] Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 509–522.

[7] Chen, J., Ma, J., Yang, C., Ma, L., Zheng, S., 2015. Non-rigid point set registration via coherent spatial mapping. Signal Processing 106, 62–72.

[8] Chen, J., Ma, J., Yang, C., Tian, J., 2014. Mismatch removal via coherent spatial relations. Journal of Electronic Imaging 23, 043012–043012.

[9] Donoser, M., Bischof, H., 2013. Diffusion processes for retrieval revisited, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE. pp. 1320–1327.

[10] Egozi, A., Keller, Y., Guterman, H., 2010. Improving shape retrieval by spectral matching and meta similarity. IEEE Transactions on Image Processing 19, 1319–1327.

[11] Felzenszwalb, P.F., Schwartz, J.D., 2007. Hierarchical matching of deformable shapes, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, IEEE. pp. 1–8.

[12] Huang, W., Gao, Y., Chan, K.L., 2010. A review of region-based image retrieval. Journal of Signal Processing Systems 59, 143–161.

[13] Jiang, J., Hu, R., Wang, Z., Cai, Z., 2016. Cdmma: Coupled discriminant multi-manifold analysis for matching low-resolution face images. Signal Processing .

[14] Jiang, J., Hu, R., Wang, Z., Han, Z., Ma, J., 2015. Facial image hallucination through coupled-layer neighbor embedding. IEEE Trans. on Circuits and Systems for Video Technology PP, 1–1. doi:10.1109/TCSVT.2015.2433538.

[15] Kontschieder, P., Donoser, M., Bischof, H., 2010. Beyond pairwise shape similarity analysis, in: Asian Conference on Computer Vision. Springer, pp. 655–666.

[16] Latecki, L.J., Lakamper, R., 2000. Shape similarity measure based on correspondence of visual parts. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1185–1190.

[17] Leibe, B., Schiele, B., 2003. Analyzing appearance and contour based methods for object categorization, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, IEEE. pp. 409–415.

[18] Li, Y., Tao, C., Tan, Y., Shang, K., Tian, J., 2016. Unsupervised multilayer feature learning for satellite image scene classification. IEEE Geoscience and Remote Sensing Letters .

[19] Ling, H., Jacobs, D.W., 2007. Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 286–299.

[20] Liu, H., Liu, S., Zhang, Z., Sun, J., Shu, J., 2014. Adaptive total variation-based spectral deconvolution with the split bregman method. Applied optics 53, 8240–8248.

[21] Liu, H., Yan, L., Chang, Y., Fang, H., Zhang, T., 2013. Spectral deconvolution and feature extraction with robust adaptive tikhonov regularization. IEEE Transactions on Instrumentation and Measurement 62, 315–327.

[22] Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110.

[23] Ma, J., Qiu, W., Zhao, J., Ma, Y., Yuille, A.L., Tu, Z., 2015a. Robust L_2E estimation of transformation for non-rigid registration. IEEE Transactions on Signal Processing 63, 1115–1129.

[24] Ma, J., Zhao, J., Ma, Y., Tian, J., 2015b. Non-rigid visible and infrared face registration via regularized gaussian fields criterion. Pattern Recognition 48, 772–784.

[25] Ma, J., Zhao, J., Tian, J., Yuille, A.L., Tu, Z., 2014. Robust point matching via vector field consensus. IEEE Transactions on Image Processing 23, 1706–1721.

[26] Ma, J., Zhao, J., Yuille, A.L., 2016. Non-rigid point set registration by preserving global and local structures. IEEE Transactions on Image Processing 25, 53–64.

[27] Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., Tian, J., 2015c. Robust feature matching for remote sensing image registration via locally linear transforming. IEEE Transactions on Geoscience Remote Sensing 53, 6469–6481.

[28] Mokhtarian, F., Abbasi, S., Kittler, J., 1996. Efficient and robust retrieval by shape content through curvature scale space, in: Proceedings of British Machine Vision Computing, pp. 53–62.

[29] Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T., 2001. Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recognition Letters 22, 593–601.

[30] Nister, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, IEEE. pp. 2161–2168.

[31] Samaria, F.S., Harter, A.C., 1994. Parameterisation of a stochastic model for human face identification, in: Proceedings of IEEE Workshop on Applications of Computer Vision, IEEE. pp. 138–142.

[32] Sebastian, T.B., Klein, P.N., Kimia, B.B., 2004. Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 550–571.

[33] Shen, W., Bai, X., Hu, R., Wang, H., Latecki, L.J., 2011. Skeleton growing and pruning with bending potential ratio. Pattern Recognition 44, 196–209.

[34] Shen, W., Wang, Y., Bai, X., Wang, H., Latecki, L.J., 2013. Shape clustering: Common structure discovery. Pattern Recognition 46, 539–550.

[35] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1349–1380.

[36] Thayananthan, A., Stenger, B., Torr, P.H., Cipolla, R., 2003. Shape context and chamfer matching in cluttered scenes, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, IEEE. pp. 127–133.

[37] Wang, J., Bai, X., You, X., Liu, W., Latecki, L.J., 2012. Shape matching and classification using height functions. Pattern Recognition Letters 33, 134–143.

[38] Wang, J., Li, Y., Bai, X., Zhang, Y., Wang, C., Tang, N., 2011. Learning context-sensitive similarity by shortest path propagation. Pattern Recognition 44, 2367–2374.

[39] Yang, J., Yu, K., Gong, Y., Huang, T., 2009a. Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1794–1801.

[40] Yang, X., Koknar-Tezel, S., Latecki, L.J., 2009b. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE. pp. 357–364.

[41] Yang, X., Prasad, L., Latecki, L.J., 2013. Affinity learning with diffusion on tensor product graph. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 28–38.

[42] Zhao, J., Ma, J., Tian, J., Ma, J., Zhang, D., 2011. A robust method for vector field learning with application to mismatch removing, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, IEEE. pp. 2977–2984.