PAPER

Integrating Multiple Global and Local Features by Product Sparse Coding for Image Retrieval

Li TIAN[†a)], ***Member***, **Qi JIA**[†b)], ***Nonmember***, ***and*** **Sei-ichiro KAMATA**[††c)], ***Member***

SUMMARY In this study, we propose a simple, yet general and powerful framework of integrating multiple global and local features by Product Sparse Coding (PSC) for image retrieval. In our framework, multiple global and local features are extracted from images and then are transformed to Trimmed-Root (TR)-features. After that, the features are encoded into compact codes by PSC. Finally, a two-stage ranking strategy is proposed for indexing in retrieval. We make three major contributions in this study. First, we propose TR representation of multiple image features and show that the TR representation offers better performance than the original features. Second, the integrated features by PSC is very compact and effective with lower complexity than by the standard sparse coding. Finally, the two-stage ranking strategy can balance the efficiency and memory usage in storage. Experiments demonstrate that our compact image representation is superior to the state-of-the-art alternatives for large-scale image retrieval.

key words: *image retrieval, image representation, Trimmed-Root (TR)-feature, Product Sparse Coding (PSC), ranking strategy*

1. Introduction

An essential issue in content-based image retrieval (CBIR), object recognition and image classification is how to represent images by numeric values, called features or descriptors. Many image features have been developed for the applications in image retrieval and computer vision fields. Roughly speaking, image features can be grouped into global types and local types based on whether they use global or local description for image representation [1]. Nowadays, large-scale image-retrieval systems require a strong image representation and efficient storage systems capable of storing billions of images. There exists a trade-off between the precision of image representation and its size. Local features store multiple local invariant features in an image and often offer better performance in representation, but they need larger storage than global ones. Global features usually have more compact representations and smaller storage requirements than local ones. Thus, both global and local features have their advantages and drawbacks in image representation.

Manuscript received August 22, 2015.
Manuscript revised November 12, 2015.
Manuscript publicized December 21, 2015.

[†]The authors are with School of Foreign Studies, Xi'an Jiaotong University, Xi'an, Shaanxi Province, 710049, China.
[††]The author is with Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan.
a) E-mail: tian.li@mail.xjtu.edu.cn
b) E-mail: jiqqikk@mail.xjtu.edu.cn
c) E-mail: kam@waseda.jp
DOI: 10.1587/transinf.2015EDP7337

Commonly used global features include color histograms [2]–[4], texture descriptors [5] and recently a gradient descriptor named GIST [6]. GIST has widely been used for image retrieval for years and small codes based on GIST have also been proposed for efficient retrieval in billion image database [7].

Existing local features include SIFT [8] and many others [9]–[11]. Because image representation by local features often suffers from the efficiency and memory usage for large-scale image retrieval, some alternative approaches aggregating local features in one image into a single vector have been developed. Among them, the bag-of-features (BOF) [12]–[15] is the most popular one. Fisher Vector [16] or Vector of Local Aggregated Descriptor (VLAD) [17] are two alternative approaches to BOF. Various approaches have also been proposed to address the retrieval efficiency and the memory usage problem for local features [18]–[21].

The performance of global features and local features for large-scale image retrieval have been compared [22]. It is reported that local features obtain significantly better results for object and location recognition. However, global features provide higher accuracy than local features in near-duplicate detection and global features also have much higher efficiency and smaller memory usage, allowing very large data sets to be processed.

Given that global features and local features are complementary in image retrieval, it is reasonable to consider integrating them to obtain higher efficiency and better retrieval results. In this paper, we propose to use Product Sparse Coding (PSC) to integrate global and local features for image retrieval to achieve better performance under three joint constraints: search accuracy, efficiency and memory usage. In our framework, multiple kinds of global and local features are extracted from an image and are transformed to Trimmed-Root (TR)-features. Then, they are encoded into compact codes by PSC. Finally, a two-stage ranking strategy is used: global part is first used as the filtering step to reduce the number of images assumed to be relevant and the remainder is used to select the final retrievals. Our contributions are:

1. multiple global and local feature integration catches image features both in entire image level and local object level;
2. TR-features have better performance than original ones in image retrieval. Because they are computed as an element wise trimmed square root, they do not require

any additional storage space;
3. concatenation approach by PSC obtains a compact yet discriminative image representation that significantly outperforms the state-of-the-art and PSC also has lower complexity than standard Sparse Coding (SC);
4. two-stage ranking strategy makes the retrieval more efficient and flexible.

Several other deep learning approaches [23] have been proposed in image retrieval, but we mainly target at improving hand-crafted features rather than fully-learnable approaches and focus on SC approaches in this study. There are already some works devoted to image categorization and retrieval through SC on raw image patches [24]–[26]. And some works use both local and global features for image retrieval or class classification [27]–[29]. However, our framework is different from them in many ways. In some works [24]–[26], they still focus on local features and do not take account into global ones. In other works [27]–[29], although both local and global features are used, our framework differs in using TR-features and PSC for coding and also the two-stage ranking strategy is more flexible.

The rest of our paper is organized as follows. We begin by reviewing related work in Sect. 2. Then, we describe the proposed framework in detail in Sect. 3. Different experiments of image retrieval demonstrate the performance of our framework in Sect. 4. The last section concludes our study.

2. Related Work

In this section, we first review some classic global features including GIST and others. Then we discuss some classic local features including SIFT SIFT, its variants, and their aggregation approaches. Finally, we give a brief introduction to SC and PSC.

2.1 Global Features

GIST was originally proposed [6] to represent a scene by a low dimensional vector for real world scene recognition. The idea is to develop a low dimensional representation of the scene. An image is first decomposed by a bank of multi-scale oriented filters (tuned to 8 orientations and 4 scales) to catch the scene structure in image. An image is simply divided by a 4-by-4 grid and orientation histograms are extracted. The resulting image representation is a $4 \times 8 \times 16 = 512$ dimensional vector. This representation can be thought of as using a single SIFT descriptor [8] to describe the entire image. This approach has recently shown good results for landmark classification [30], scene parsing [31], image completion [32], and image searching [22], [33]. A typical GIST has 512 dimensions and different strategies [7], [34], [35] have been proposed to further compress the size.

Color histograms [2]–[4] and texture descriptors [5] are also commonly used global features in image retrieval. A color histogram is a representation of the distribution of colors in an image and it is simple but useful. The main drawback of histograms is that the representation is dependent of the color of the image being studied, ignoring its shape and texture. Texture descriptors use image texture which is one important characteristics used in identifying objects or regions of interest in an image, but it does not work well for natural images without texture.

2.2 Local Features and Aggregation

SIFT feature was first developed by Lowe [8] and has been approved as the most useful local image features in computer vision fields. Original SIFT is computed on a small patch, i.e., 32-by-32 pixels, 8 orientations and 4-by-4 grid are used to compute orientation histogram, which results in a $8 \times 4 \times 4 = 128$ dimensional vector. Many variants including PCA-SIFT [36], GLOH [9], SURF [11] and DAISY [10] been proposed based on SIFT. They match small patches of images and is robust to image transformations. Because more than hundreds of local features may be extracted from a single image to represent it, it is not suitable for object recognition and image retrieval in large-scale image database. Thus, Bag-of-Features (BOF) [12] is proposed to solve the problem.

The BOF representation is based on local descriptors such as SIFT extracted at invariant regions of interest. First, interest regions are detected by some detectors such as Hessian-Affine, and SIFT descriptors for those interest regions are computed. Then, each local descriptor is assigned to the closest "visual words" by using a codebook of k "visual words" pre-constructed by k-means clustering. Finally, an image can be represented as a histogram of the assignment of all image descriptors to visual words in the image. The codebook often contains a large number of visual words. Therefore, it produces a k-dimensional vector and is very sparse, making queries in the inverted file efficient.

Fisher Vector [16] or Vector of Local Aggregated Descriptor (VLAD) [17] are two alternatives to BOF. For Fisher Vector, local descriptors are coded as the average of probabilities that feature belongs to the each Gaussian component of a codebook which is a pre-learned GMM model. And for VLAD, the differential (residual) of vector and its k-means centroid is used.

2.3 SC and PSC

Given a potentially large set of input patterns, SC attempts to automatically find a small number of representative patterns which, when combined in the right proportions, can reproduce the original input patterns. The sparse coding for the input then consists of those representative patterns. Most models of sparse coding are based on the linear generative model [37], in which the symbols are combined in a linear fashion to approximate the input. Sparse coding of image patches has been successfully applied to tasks such as image and video denoising [38], restoration [39], super-resolution [40], segmentation [41] and face recognition [42].

Because SC is computational expensive, PSC is proposed to solve the complexity issue [43]. PSC shares the same encoding model as SC, but requires the codebook to be a Cartesian product of two smaller subcodebooks. PSC can reduce the time complexity of normal sparse coding from $O(K)$ to $O(\sqrt{K})$ in the codebook size K. We will give more details of PSC in introducing our framework later.

3. Proposed Framework

In this section, we describe how to transform features to Trimmed Root (TR)-features by using GIST and SIFT as representatives at first. Then, we introduce how to use PSC to integrate multiple global and local features to obtain compact image representation. After that, we demonstrate a two-stage ranking strategy and show how to use it to balance the accuracy and efficiency in image retrieval. Finally, we give some explanations why we integrate global and local features for large-scale image retrieval.

3.1 TR-Features

It is shown that using a square root (Hellinger) kernel instead of the standard Euclidean distance to measure the similarity between SIFT descriptors leads to a dramatic performance boost in image retrieval [18]. In this study, we transform SIFT to TR-SIFT. TR-SIFT is computed as an element wise square root of the L_1 normalized SIFT vectors. If the value of the feature is larger than a predefined threshold we set it to zero. Then we replace SIFT with the proposed TR-SIFT at every point in retrieval pipeline. Thus, we can obtain features including more zero values and it is preferred in PSC.

Because GIST can be reviewed as computing SIFT descriptor on the entire image, it is expected that TR-GIST, which is computed as an element wise trimmed square root of GIST, can give a performance boost comparing to GIST.

The improvement is simple but powerful. We apply it to different global and local features in image retrieval in this study. We show that our TR-features makes a dramatic performance improvement in the experiments.

3.2 Integrating Features by PSC

Figure 1 illustrates the flowchart of how to integrate multiple global and local features. It usually contains two steps: encoding multiple global and local features by PSC, and normalization and weighting. PSC encodes multiple kinds of global and local features from an image into a sparse vector. Our approach consists of the following major parts:

Part 1: Encoding Global Features. If we have multiple kinds of global features from an image, for each kind of global feature $\mathbf{x}$, it can be encoded into a d-dimensional vector $\mathbf{y} = [y^1, y^2, \ldots, y^d]$ by fitting a linear model with sparsity (L_1) constraint:

$$\min_{\mathbf{y}} \|\mathbf{x} - \mathbf{A}\mathbf{y}\|^2 + \lambda|\mathbf{y}|, \tag{1}$$

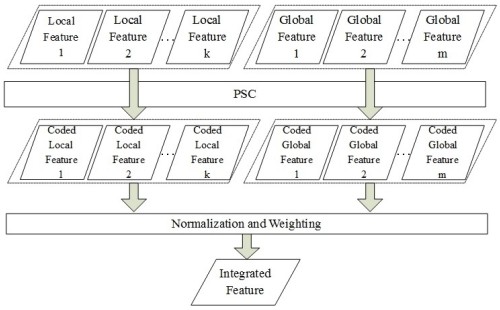

Fig. 1 Flowchart of integrating multiple global and local features by PSC.

$$\text{subject to } \mathbf{y} \geq 0$$

$$\text{and} A = A_1 \times A_2$$

where $\times$ denotes the Cartesian product. A_1 and A_2 are two subcodebooks of a size $1/2d \times k$ learned in advance by PSC. Any codeword in A is the concatenation of a subcodeword in A_1 and a subcodeword in A_2. So A is a $d \times K$ matrix with $K = k^2$. Here $\mathbf{y} \geq 0$ means that all the elements of $\mathbf{y}$ are nonnegative. Thus, the time complexity of each subproblem becomes linear in $\sqrt{K}$. Finally, a coded feature is normalized by

$$\mathbf{y} := \frac{\mathbf{y}}{\| \mathbf{y} \|_2}. \tag{2}$$

Part 2: Encoding Local Features. If we have multiple kinds of local features from an image, for each kind of local features $\mathbf{X}'$, let $\mathbf{X}'$ be a set of m dimensional local descriptors with n local features, i.e. $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_n]^T$. We can obtain their corresponding sparse codes $[\mathbf{u}'_1, \mathbf{u}'_2, \ldots, \mathbf{u}'_n]^T$ as in encoding global features previously for each descriptor by PSC.

Then, we pool them into a single m dimensional vector $\mathbf{u} = [u^1, u^2, \ldots, u^m]$. Two pooling methods may be used in this step–average pooling and max pooling:

$$\text{average pooling: } u^i = \sum_{t=1}^{n} u'^i_t; \tag{3}$$

$$\text{max pooling: } u^i = \max \{u'^i_t \mid t = 1, 2, \ldots, n\}. \tag{4}$$

The pooled vector $\mathbf{u}$ is normalized by

$$\mathbf{u} := \frac{\mathbf{u}}{\| \mathbf{u} \|_2}. \tag{5}$$

In most cases, average pooling is better than max pooling and we choose it in this study.

Part 3: Integration. After encoding p kinds of global features and q kinds of local features, we can obtain a set of coded global features $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p]$ with $d \times p$-dimensional vector and a set of coded local feature $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q]$ with $m \times q$-dimensional vector. The final image representation $d \times p + m \times q$ dimensional vector $\mathbf{Z}$ is a combination of $\mathbf{Y}$ and $\mathbf{U}$ by a weighting parameter w:

$$\mathbf{Z} = [w\mathbf{Y}, (1-w)\mathbf{U}]. \quad (6)$$

In two extreme cases,$w = 1$ or $w = 0$, $\mathbf{Z}$ equals to using only global features or local features, respectively. Note that the features may be concatenated before PSC, but it is necessary to constructing concatenation feature codebooks in learning phase. Thus, we choose to code it by PSC before concatenation in this study for convenience and flexibility.

3.3 Two-Stage Ranking Strategy

For large-scale image database, i.e., a billion image dataset, we have to scan dozens of millions of images in the database by directly using the coded features above. Hence, we propose a two-stage ranking strategy in this study. It can be used to refine the ranking result both in accuracy and efficiency.

Figure 2 shows the process of the two-stage ranking strategy. We use the global part in the first ranking stage to filter a subset of the full retrieval set and use the local part in the second ranking stage to find the final similar images. Without loss of generality, we also use TR-GIST and TR-SIFT as the global and local feature representatives to demonstrate the strategy here.

The first ranking is based on the comparison of the encoded TR-GIST descriptors to produce a list of images ranked according to the Euclidean distance. That means only a part of images whose similarity S are smaller than a threshold S_0 are ranked and enter into the second stage. Alternatively, we can specify a percentage of selected images that can enter the next stage here, i.e., 50% top ranked images are selected into the second stage. Then, the selected images are compared again by using the local part of the descriptors $\mathbf{y}$ in the second stage. In the second stage, images whose similarity $S1$ are smaller than a threshold $S1_0$ are treated as the final similar images. We also can specify a percentage of selected images as the final similar images in this stage.

For different retrieval purposes, we may use local part in first ranking stage and use global part in the second stage or use global part in first ranking stage and use the whole feature in the second stage. Because global descriptors such as TR-GIST can capture the spatial structure better, we will choose it when we want to select similar images in global view; and we will use local descriptors such as TR-SIFT when we want to select similar images conclude similar objects since it can capture local features better. Based on many experimental results, we find that global-local strategy is more effective.

3.4 Reason for Integrating Global and Local Features

Figure 3 shows three groups of similar images from the INRIA Holidays dataset [22], which is usually used as a benchmark database in similar image retrieval. Probably everyone would agree that the images in the first row, are similar because they look nearly the same. Many would probably say that the images in the second row are also similar since they contain the same object taken from different views. The third row containing images relating to sail is the most ambiguous one, and it is not easy to give a yes or no answer as to whether they are similar. This illustrates our main concern: what people consider to be similar images.

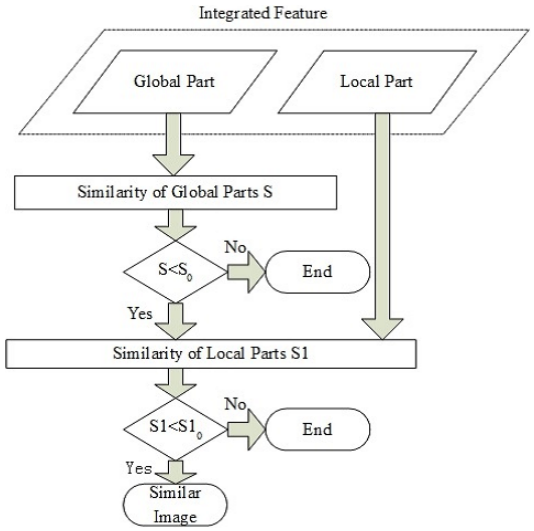

Fig. 2 Flowchart of two-stage ranking strategy.

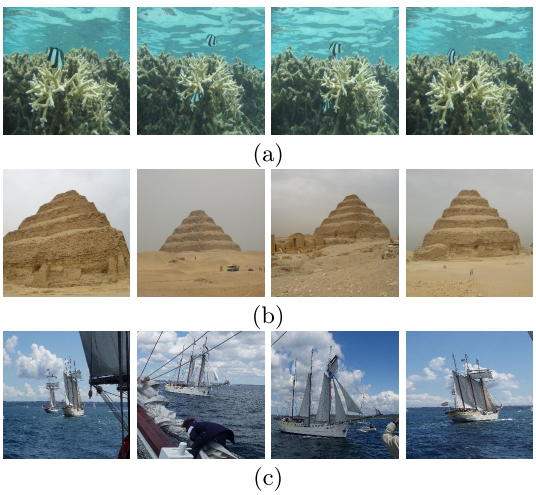

Fig. 3 Similar images in different meaning levels. (a) Duplicate level. (b) Near-duplicate level. (c) Near-semantic level.

We now return to our concern what people consider to be "similar" images in similar image retrieval. In general, similar image retrieval is only meaningful in its service to people [44], so we prepared a questionnaire about the definition of similar images for several objects and used the answers to it to determine similarity at three meaning levels:

Duplicate level Duplicate is used here to distinguish it from the near-duplicate approach [45] commonly used in image retrieval. Unlike near-duplicate images, duplicate similar images here are considered as those

taken from the same location with small transformations as shown in Fig. 3 (a).

Near-duplicate level As used here, near-duplicate mostly refers to the images containing the same object taken from different viewpoints with certain transformations as shown in Fig. 3 (b).

Near-semantic level Images are defined as "similar" based on semantically-meaningful information from the image content as shown in Fig. 3 (c).

As mentioned previously, both global and local approaches have their advantages and drawbacks in image retrieval since similar image has different meaning levels, it is reasonable to integrating global and local features for image retrieval to obtain flexibility in this study. By adjusting the weights of local and global feature parts, our integrated feature can handle with different meaning levels. It is better to use larger weight of global part if the images are near duplicate level while larger weight of local part should be used when the images are near semantic level.

4. Experiments

In order to illustrate the performance of our framework, we set up several different experiments in this study. We first evaluate TR-features; We then provide comparisons to show the effect of integrating global and local features by PSC; We finally illustrate the two-stage ranking strategy.

4.1 Experiments on TR-Features

Without loss of generality, we choose TR-GIST and TR-SIFT as the global and local feature representatives to demonstrate the improvement in this experiment. Because the dramatic improvement in performance by using a square root (Hellinger) kernel in SIFT named RootSIFT on Oxford 105k dataset (tf-idf only) is shown in [18], we choose to use Oxford 105k dataset to compare them with our TR-SIFT. It contains 5062 Oxford building images and defines 55 queries with another 100k Flickr images to test large-scale retrieval. The first image of each group is the query image and the correct retrieval results are the other images of the group. The accuracy is measured by mean Average Precision (mAP), where the mean is taken over all queries [14]. The threshold for trimming in TR-SIFT is 0.0005. The retrieval results are as in Table 1. We can see that our TR-SIFT obtain better performance than two others in Table 1. The improvement is from 0.515 to 0.581 by RootSIFT and further to 0.593 by our TR-SIFT.

Also, because the performance of GIST in image retrieval on standard datasets INRIA Holiday has been reported in [22], we use the standard datasets INRIA Holiday to evaluate the performances for fair. The dataset contains 500 image groups, each of which represents a distinct scene or object and 991 corresponding relevant images. We compared GIST, RootGIST and our TR-GIST in the retrieval. The accuracy is also measured by mAP. The threshold for trimming in TR-GIST is 0.0002. The retrieval result is as in Table 2. The improvement is from 0.376 to 0.414 by RootGIST and further to 0.428 by our TR-GIST.

Table 1 Retrieval results by SIFT, RootSIFT and TR-SIFT.

	mAP
SIFT	0.515
RootSIFT	0.581
TR-SIFT	0.593

Table 2 Retrieval results by GIST, RootGIST and TR-GIST.

	mAP
GIST	0.376
RootGIST	0.414
TR-GIST	0.428

It is shown that the performance improvements of our TR-features are obtained by root and trimming modifications comparing to original SIFT and GIST. As mentioned previously, a square root modification can change the similarity kernel. Trimming modification can ignore the features with small gradient in the image, which can be viewed as a filter may help to improve the performance. These improvements come at virtually no additional cost, and no additional storage since GIST and SIFT can be converted online to TR-GIST and TR-SIFT with a negligible processing overhead.

4.2 Experiments on Integrated Features by PSC

Here, we will compare our integrated feature with the-state-of-art such as Fisher Kernel, VLAD in image retrieval. We use the INRIA Holidays+ 1m dataset for large-scale experiment. The dataset contains the Holiday dataset with 1 million distracter images downloaded from Flickr which is called Holidays+1m dataset. The accuracy is also measured by mAP.

For the dimensions of integrated TR-GIST and TR-SIFT, we choose $d = 4096$ and $m = 4096$ in a 8192 dimensional vector. We set the weighting parameter $w = 0$ and $w = 0.5$. Note that when $w = 0$, it means that we only use TRD-SIFT in the integrated feature. Thresholds for trimming in TR-SIFT and TR-GIST are 0.0005 and 0.0002, respectively. The features are detected by the Harris detector and a 128-D SIFT descriptor is computed for each feature for Fisher Kernel and VLAD. Then the descriptors are aggregated by Fisher Kernel and VLAD, respectively. The codebook sizes are k = 64 for both Fisher and VLAD, resulting in a 128×64 = 8192 dimensional aggregated vectors for comparison. In order to show the effectiveness of PSC, we added SIFT+PSC for comparison. All the above features are also compressed from 8192 dimension to 128, 64 and 32 dimensions by PCA here.

The retrieval results are shown in Table 3. As we can see, PSC performs better than Fisher Kernel and VLAD. Comparing with Fisher Kernel and VLAD, the improvement is from 0.492 and 0.525 to 0.534 by using PSC at 8192 dimension. And the result is further improved to 0.542 ($w = 0$) and 0.546 ($w = 0.5$) by integrating TR-SIFT and TR-GIST.

Table 3 Retrieval results by different features.

strain	mAP			
	D=8192	D=128	D=64	D=32
Fisher	0.492	0.490	0.460	0.424
VLAD	0.525	0.511	0.473	0.422
original SIFT+PSC	0.534	0.526	0.483	0.428
PSC(w=0)	0.542	0.530	0.484	0.432
PSC(w=0.5)	0.546	0.542	0.496	0.434

Table 4 Retrieval results by SC and PSC.

strain	mAP			
	D=8192	D=128	D=64	D=32
SC(w=0.5)	0.552	0.548	0.492	0.428
SC(w=0)	0.534	0.515	0.467	0.410
PSC(w=0.5)	0.546	0.542	0.496	0.434

Table 5 Retrieval results by different weights.

the weighting parameter w	mAP
w=0.1	0.542
w=0.3	0.548
w=0.5	0.546
w=0.7	0.534
w=0.9	0.512

Note that when $w = 0$, it equals to use TR-SIFT only. And PSC also performs better than others at reduced dimensions by PCA.

Then, we compare the performances of the integrated features by SC and PSC in Table 4. $w = 0$ means using only local features as [26]. From the table, we can see that SC is slightly better than PSC at 8192 dimension but performs equally or worse than PSC at reduced dimensions by PCA. On a computer with 3.4GHz CPU and 8 GB memories, the computational times for SC, PSC, VLAD are 9.5, 0.55, 0.98 seconds for encoding one image, respectively. Considering the computational complexity, we believe that PSC is more suitable in large-scale image retrieval.

Third, we evaluate how the weighting parameter w works here. We set the w from 0.1 to 0.9 at 0.2 interval and repeat the retrieval experiment above. The results are shown in Table 5. From the table, we find that the weighting parameter w affect the retrieval results slightly and all the results are better than Fisher Kernel and VLAD methods excepting $w = 0.9$.

Finally, we tried to integrate more different kinds of features. We add DAISY [10], or color histogram [2], or both of them in local and global parts. The weighting parameters are 1/3 for each part when integrating three features and 1/4 for each part when integrating four features. We make the dimensions are 128 for all cases by PCA for fair comparison. The retrieval results are shown in Table 6. We can conclude that using more features can further improve the retrieval performance. Considering the computational cost, we may choose TR-GIST and TR-SIFT as the two representatives in most cases.

Table 6 Retrieval results by integrating different kinds of features.

	mAP
TR-GIST+TR-SIFT	0.542
TR-GIST+TR-SIFT+DAISY	0.548
TR-GIST+TR-SIFT+color histogram	0.534
TR-GIST+TR-SIFT+DAISY+color histogram	0.560

Table 7 Retrieval results using different percentages in the first ranking stage.

Selected percentage in the first stage	mAP
10%	0.428
30%	0.506
50%	0.514
70%	0.544
90%	0.538

4.3 Experiments on Two-Stage Ranking Strategy

In this experiment, we will evaluate the proposed two-stage ranking method in our framework for image retrieval. As mentioned previously, we may choose a threshold or specify a percentage to produce a list of ranked images in the first ranking stage. In this experiment, we use a percentage from 10% to 90% at an interval 20% to evaluate the strategy. We used the TR-GIST+TR-SIFT as the integrated feature. The first ranking is based on the comparison of the encoded TR-GIST descriptors $\mathbf{u}$ to produce a list of ranked images according to the Euclidean distance. Then, the ranked images are compared again by using the local descriptor part TR-SIFT.

The retrieval results using different percentages in the first ranking stage are shown in Table 7. From the table, we find that the retrieval results changes slightly when using different percentage in the first ranking stage. Comparing with the $mAP = 0.542$ by using the same feature without ranking strategy, the results are comparable in most cases. Especially, when the percentage is 70%, the result is 0.544, which is even better than 0.542 which is without ranking strategy. That is because some confusing images which are similar globally but not locally are filtered in the first stage. The whole process times for different percentages are nearly proportional to the percentages. Considering the time saving in searching, the results are very satisfying and the two-stage ranking strategy gives flexibility between the retrieval accuracy and speed.

5. Conclusions and Future Work

This study presents a framework for integrating global and local features based on Product Sparse Coding (PSC) with a two-stage ranking strategy. We also transform features to Trimmed-Root (TR)-features and it is shown that TR-features offer better performance than original versions and do not require any additional storage space. Compared with other state-of-the-art systems, our framework shows its superiorities in large-scale image retrieval accuracy and PSC

has lower computational complexity than standard Sparse Coding. Moreover, our framework can give the flexibility in retrieval speed and accuracy by using a two-stage ranking strategy. Future work will aim at extending our framework to ingrate more different kinds of features, automatically optimize weighting parameter w against the dataset, and explore real applications in large-scale image retrieval system.

Acknowledgements

We would like to thank all people providing their free code and images for test and all the related people for their contributions to this study.

References

[1] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," Inf. Retr, vol.11, no.2, pp.77–107, 2008.
[2] M.J. Swain and D.H. Ballard, "Color indexing," IJCV, vol.7, no.1, pp.11–32, Nov. 1991.
[3] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," IEEE Trans. PAMI, vol.22, no.12, pp.1349–1380, 2000.
[4] J. Puzicha, J.M. Buhmann, Y. Rubner, and C. Tomasi, "Empirical evaluation of dissimilarity measures for color and texture," ICCV, pp.1165–1172, 1999.
[5] R.M. Haralick, K.S. Shanmugan, and I. Dunstein, "Textural features for image classification," IEEE Trans. SMC, vol.3, no.6, pp.610–621, 1973.
[6] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," IJCV, vol.42, no.3, pp.145–175, May 2001.
[7] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," CVPR, pp.1–8, 2008.
[8] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, vol.60, no.2, pp.91–110, 2004.
[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. PAMI, vol.27, no.10, pp.1615–1630, Oct. 2005.
[10] E. Tola, V. Lepetit, and P. Fua, "A Fast Local Descriptor for Dense Matching," CVPR, Alaska, USA, 2008.
[11] H. Bay, T. Tuytelaars, and L.J.V. Gool, "SURF: Speeded up robust features," ECCV, vol.3951, pp.404–417, Springer Berlin Heidelberg, 2006.
[12] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," ICCV, pp.1470–1477, 2003.
[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," CVPR, pp.II: 2161–2168, 2006.
[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," CVPR, pp.1–8, 2007.
[15] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," ECCV, pp.I: 304–317, 2008.
[16] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," CVPR, pp.3384–3391, IEEE, 2010.
[17] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," CVPR, pp.3304–3311, IEEE, 2010.
[18] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," CVPR, pp.2911–2918, IEEE, 2012.
[19] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," CVPR, pp.817–824, IEEE, 2011.
[20] L. Torresani, M. Szummer, and A. Fitzgibbon, "Learning query-dependent prefilters for scalable image retrieval," CVPR, pp.2615–2622, 2009.
[21] J.C. Yang, K. Yu, Y.H. Gong, and T.S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," CVPR, pp.1794–1801, 2009.
[22] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," Proc. the ACM International Conference on Image and Video Retrieval, CIVR '09, New York, NY, USA, pp.19:1–19:8, ACM, 2009.
[23] J. Wan, D. Wang, S.C.H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," Proceedings of the ACM International Conference on Multimedia, MM '14, New York, NY, USA, pp.157–166, ACM, 2014.
[24] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng, "Self-taught learning: transfer learning from unlabeled data," ICML, ed. Z. Ghahramani, ACM International Conference Proceeding Series, vol.227, pp.759–766, ACM, 2007.
[25] M. Ranzato, F.J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," CVPR, pp.1–8, 2007.
[26] T. Ge, Q. Ke, and J. Sun, "Sparse-coded features for image retrieval," Proceedings of the British Machine Vision Conference, BMVA Press, pp.132.1–132.11, 2013.
[27] Y. Gong, S. Kumar, H.A. Rowley, and S. Lazebnik, "Learning binary codes for high-dimensional data using bilinear projections," CVPR, pp.484–491, IEEE, 2013.
[28] D.A. Lisin, M.A. Mattar, M.B. Blaschko, M.C. Benfield, and E.G. Learned-Miller, "Combining local and global image features for object class recognition," p.47, Jan. 2005.
[29] J. Jeong, H. Jeon, C. Hwang, and B. Jeon, "Efficient image feature combination with hierarchical scheme for content-based image management system," MUE, pp.539–545, IEEE Computer Society, 2009.
[30] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," IEEE Trans. PAMI, vol.29, no.2, pp.300–312, Feb. 2007.
[31] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," CVPR, pp.1972–1979, 2009.
[32] J. Hays and A.A. Efros, "Scene completion using millions of photographs," Commun. ACM, vol.51, no.10, pp.87–94, 2008.
[33] X.W. Li, C.C. Wu, C. Zach, S. Lazebnik, and J.M. Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," ECCV, pp.I: 427–440, 2008.
[34] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," ICCV, pp.2130–2137, IEEE, 2009.
[35] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," NIPS, ed. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, pp.1753–1760, Curran Associates, Inc, 2008.
[36] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," CVPR (2), pp.506–513, 2004.
[37] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed in V1," Vision Research, vol.37, no.23, pp.3311–3325, 1997.
[38] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. Image Processing, vol.15, no.12, pp.3736–3745, Dec. 2006.
[39] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," IEEE Trans. Image Processing, vol.17, no.1, pp.53–69, Jan. 2008.
[40] J.C. Yang, J. Wright, T.S. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," CVPR, pp.1–8, 2008.

[41] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," CVPR, pp.1–8, 2008.
[42] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," CVPR, pp.625–632, IEEE, 2011.
[43] T. Ge, K. He, and J. Sun, "Product sparse coding," CVPR, pp.939–946, IEEE, 2014.
[44] N.V. Shirahatti and K. Barnard, "Evaluating image retrieval," CVPR, pp.I: 955–961, 2005.
[45] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system," ACM Multimedia, ed. H. Schulzrinne, N. Dimitrova, M.A. Sasse, S.B. Moon, and R. Lienhart, pp.869–876, ACM, 2004.

Li Tian is a lecturer at Xi'an Jiaotong University, where he received the bachelor degree in 2003. He received his M.S. degree and Ph.D degree from Waseda University in 2006 and 2009. From 2006.4 to 2007.3, he was a Research Fellow of the International Communications Foundation (ICF) for Graduate Students from Abroad, and was a Research Fellow of the Japan Society for the Promotion of Science (JSPS) from 2007.4 to 2009.3. From 2009.4 to 2011.3, he was a researcher in NTT Cyber Space laboratories. Since 2013.4, he joined Xi'an Jiaotong University. His current research interests are in data mining, image processing, pattern recognition and natural language processing. He is a member of the IEICE.

Qi Jia is an associated professor at Xi'an Jiaotong University, where she received the bachelor and master degree in 2003 and 2006. She received her Ph.D degree from Kyushu University in 2010. Since 2010, she joined Xi'an Jiaotong University. Her current research interests are in conversation analysis and natural language processing.

Sei-ichiro Kamata received the M.S. degree in computer science from Kyushu University, Fukuoka, Japan, in 1985, and the Doctor of Engineering degree from the Department of Computer Science, Kyushu Institute of Technology, Kitakyushu, Japan, in 1995. From 1985 to 1988, he was with NEC, Ltd., Kawasaki, Japan. In 1988, he joined the faculty at Kyushu Institute of Technology. From 1996 to 2001, he was an Associate Professor in the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been an Professor in Graduate School of Information, Production and Systems, Waseda University. In 1990 and 1994, he was a Visiting Researcher at the University of Maine, Orono. His research interests include image processing, pattern recognition, image compression, and space-filling curve applications. Dr. Kamata is a member of the IEEE and the IEICE.