Neurocomputing 120 (2013) 355–364

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Feature integration analysis of bag-of-features model for image retrieval

Jing Yu [a], Zengchang Qin [a,*], Tao Wan [b], Xi Zhang [a]

[a] Intelligent Computing and Machine Learning Lab, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
[b] School of Medicine, Boston University, Boston, USA

ARTICLE INFO

Article history:
Received 25 January 2012
Received in revised form
17 July 2012
Accepted 24 August 2012
Available online 29 March 2013

Keywords:
Bag-of-features (BoF)
Image retrieval
Weighted K-means
SIFT-LBP
HOG-LBP
Histogram intersection

ABSTRACT

One of the biggest challenges in content based image retrieval is to solve the problem of "semantic gaps" between low-level features and high-level semantic concepts. In this paper, we aim to investigate various combinations of mid-level features to build an effective image retrieval system based on the bag-of-features (BoF) model. Specifically, we study two ways of integrating the SIFT and LBP descriptors, HOG and LBP descriptors, respectively. Based on the qualitative and quantitative evaluations on two benchmark datasets, we show that the integrations of these features yield complementary and substantial improvement on image retrieval even with noisy background and ambiguous objects. Two integration models are proposed: the patch-based integration and image-based integration. By using a weighted K-means clustering algorithm, the image-based SIFT-LBP integration achieves the best performance on the given benchmark problems comparing to the existing algorithms.

1. Introduction

Content-based image retrieval (CBIR) is a technique to search for the most visually similar images to a given query image from a large image database. It has received increasing attentions in recent years and become a hot topic in computer vision [1,2], image processing [3,4], and multimedia [5,6]. The main idea of image retrieval is to extract low-level features from the images and measure the degree of similarity between them to find the most similar ones in terms of visual contents. Colors, textures, and shapes have been used to describe the image contents [4]. There are advantages and disadvantages of using these low-level features on CBIR systems. Color features have high computational efficiency and are invariant to rotation and scale. However, they do not consider image content and spacial distribution of colors. The incomplete color space describing the color characteristics in human imperfect visual perception has yet to be solved. Texture features, describing the spatial variation in pixel intensities, can reflect the surface attributes of an object.

In [7], a number of different methods are proposed to describe image textures. However, texture segmentation still remains a difficult problem to meet human perception [8]. Shape-based features are relatively consistent with the intuitive feeling but lacking perfect mathematical foundations to deal with the target deformation [4]. Therefore, only using low-level visual features can hardly capture the semantic concepts of images. In recent years, mid-level features have attracted more attentions. Among them, the scale-invariant feature transform (SIFT) operator [9] and its improved versions, such as PCA-SIFT [10], SURF [11], affine invariant SURF [12] are invariant to rotation, scaling, translation and small distortions. Nevertheless, perfect retrieval results cannot be achieved in practice because of complex background in image data. The histogram of oriented gradients (HOG) [13] descriptor is similar to the SIFT and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. HOG descriptor counts occurrences of gradient orientation in localized portions of an image. The basic idea behind the HOG descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. To increase the applicability of HOG, an extension of it has been proposed in [6] that makes use of a gradient decomposition and combination strategy to improve the human detection accuracy. The local binary pattern (LBP) descriptor [14] is considered as one of the most popular texture features due to its discriminative power and computational simplicity. Moreover, the most important property of the LPB operator is its robustness to monotonic gray-scale changes caused, for example, by illumination variations, making it is desirable to analyze images in challenging real-world applications.

All these three types of features have been proven to be very powerful descriptors to detect and describe local features in images. Both the SIFT and HOG features are capable of capturing local object shape or edge with the distributions of intensity gradients. For an image with simple background, the SIFT and HOG features are able to accurately represent the foreground

* Corresponding author. Tel.: +86 10 82316875.
E-mail addresses: zcqin@buaa.edu.cn, zengchang.qin@gmail.com (Z. Qin).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.08.061

objects without noise interference. However, the SIFT and HOG will perform poorly when the image contains complex background due to the fact that a portion of extracted features may come from the noisy background. On the contrary, the LBP descriptor does not take into account shape information in images. In addition, it can filter out background noise through local binary texture patterns [15], in which uniform patterns with consistent changes are labeled with separate labels while all the non-uniform patterns are considered as a single label. In this fashion, noisy background does not significantly affect the similarity measure between images. We believe that combining the shape information, such as obtained from the SIFT or HOG, with the texture information captured by the LBP operator provides a more robust and precise representation of images compared to a single feature.

Recently, there are many feature fusion algorithms published in literature. For example, Wang et al. [16] proposed an integrated HOG-LBP feature on the cell-level as a human detector. It has been reported that the method achieves good performance in the INRIA dataset. Heikkilä et al. [17] derived a new descriptor combining the strengths of the SIFT and LBP, in which center-symmetric local binary patterns were used to replace the gradient operator used by the SIFT operator for region detection and image matching applications. Zheng et al. [18] combined the SIFT and rotation-invariant LBP for an image matching problem, where the LBP descriptor was used to describe the local region centered at the keypoints which were found by the SIFT feature. In addition, Schwartz et al. [19] introduced a human detection method by integrating edge-based features with texture and color information to form a relative large descriptor set. Hiremath and Pujari [4] considered the combination of color, texture, and shape information to build an efficient framework for image retrieval, which is also used as a reference method to be compared to our proposed feature integration models.

In this paper, we design two local semantic descriptors by integrating the SIFT, HOG with the LBP descriptor. The new features do not rely on the image segmentation method and are able to automatically detect interest points and regions within an image. The proposed integration model is based on the bag-of-features (BoF) representations [20]. Features that are computed for each image are combined to form high-dimensional descriptors. These descriptors are clustered into several key points which are referred to as visual words. Each image is then represented by a distribution on the visual words. Given a query image, we are able to find a list of similar images ranked by similarity scores based on visual word distributions.

The remaining paper is organized as follows: Section 2 describes the framework of bag-of-features model and the similarity matching method. Section 3 provides a brief review of the SIFT, LBP and HOG features. We introduce two integration methods and apply them to combine the SIFT and LBP, HOG and LBP, respectively. The experimental results are presented in Section 4. The conclusions are drawn in Section 5.

2. Bag-of-features model

The bag-of-features method is largely inspired by the concept of bag-of-words (BoW) [21] which has been used in text mining. In the BoW model, each word is assumed to be independent. Though it is counterintuitive, the BoW has been well known in spam filtering and topic modeling [22] with outstanding performance. In the BoF model, each image is described by a set of orderless local features [23]. It includes four key concepts: (1) local features: the essential aspect of the BoF concept is to extract global image descriptors and represent images as a collection of local properties calculated from a set of small sub-images called patches. For

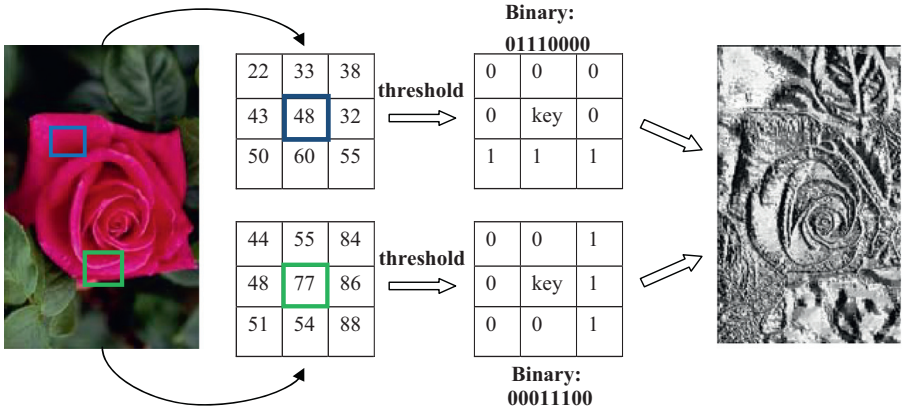


Fig. 1. The classic LBP descriptor. For each pixel, compare the pixel to each of its 8 neighbors. Where the center pixel's value is greater than the neighbor, write "1". Otherwise, write "0". This gives an 8-digit binary number.

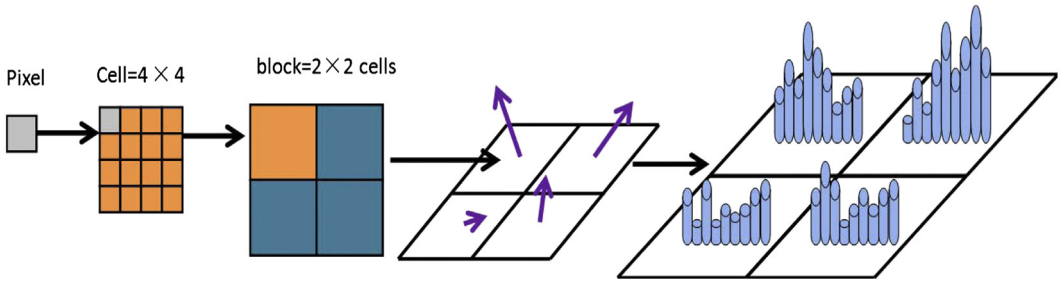


Fig. 2. The HOG descriptor. An image is divided into small connected regions with 4×4 pixels, called cells. For each cell, compile a histogram of gradient directions within the cell. The combination of this histograms in a block represents a HOG descriptor.

example, the SIFT patches are small rectangular regions centered on interest points, while the LBP patches are small round regions with the desired radius and a number of sampling points. (2) Codebook representation: codebook is a way that images can be represented as a set of local features [20]. The idea is to group the feature descriptors of all patches, and the representatives of resulting clusters are then used as entries of an unified codebook [21]. The entries are called "visual words". (3) Feature quantization: after obtaining the codebook, each local feature is quantized to one "visual word" using unsupervised learning methods, such as K-nearest neighbor (KNN) classifier. (4) Image representation: as all the features are mapped to the codebook, an image can be represented by the BoF frequency histogram of the "visual words" in the codebook. The similarity measure between two images can be computed by comparing these two BoF histograms. A histogram intersection is used to compute the similarity between two histograms of given images A and B, which is defined by

$$d(A,B) = 1 - \sum_{i=1}^{n} \min(a_i, b_i) \quad (1)$$

where a_i and b_i represent the probabilities of visual words of image A and B, respectively. For a given query image Q, the distance between Q and each image in the database will be calculated. A set of images are selected and sorted in ascending order based on their distance values.

3. Feature integration

The scale-invariant feature transform has been proven to be one of the most robust among the local invariant feature descriptors with respect to different geometrical changes [24]. It represents blurred image gradients in multiple orientation planes and at multiple scales. The SIFT has showed great success in object recognition and detection due to its invariance in translation, scaling, rotation, and small distortions. The basic idea is to identify the extreme points in the scale space, and filter these extreme points to find the stable feature points known as *keypoints*. The local attributes of orientation gradient and describe are computed, and the keypoints are described by $4 \times 4 \times 8$ matrix.

The LBP operator [15] is a texture descriptor that has been widely used in object recognition and achieved good results in face recognition problems. Its key advantages, namely its invariance to monotonic ray level changes and rotation, make it suitable for demanding image analysis tasks [16] such as object retrieval. The LBP method is schematically shown in Fig. 1. Previous research which used uniform patterns representing the most essential texture information showed a strong discriminative ability [15]. Uniform pattern LBP considers only those LBP codes that have U value of no more than 2 (U refers to the measure of uniformity, that is the number of 0/1 and 1/0 transitions in the circular binary code pattern). In case of $LBP^{u}_{8,1}$ (8 and 1 are respectively the number of neighboring sample points and the radius of $LBP_{P,R}$,

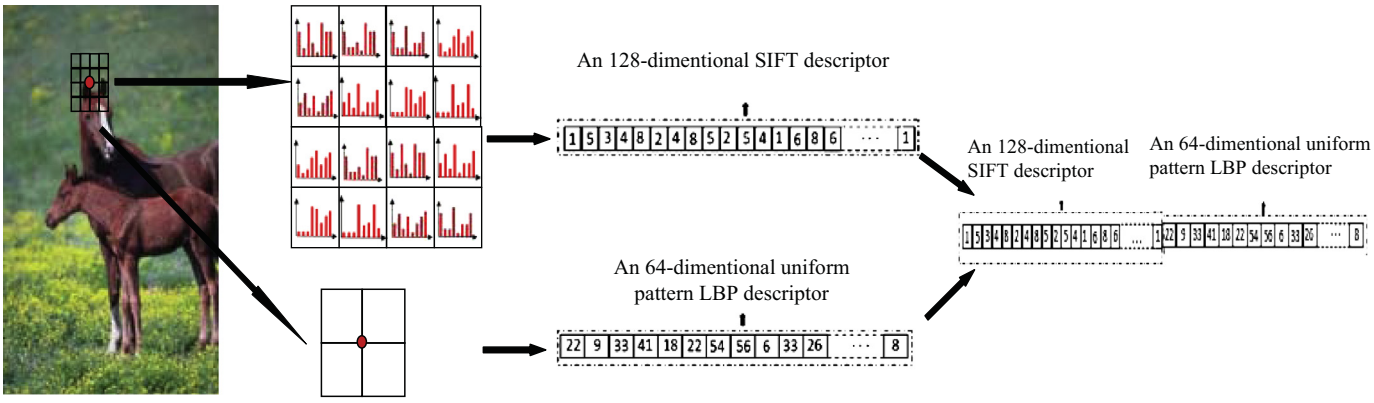


Fig. 3. The integration of the patch-based SIFT-LBP. First, an 128-dimensional SIFT descriptor is computed for a patch; An 8×8 region around the keypoint is chosen and an uniform pattern LBP is then computed on each pixel in this patch; the SIFT and LBP features are finally concatenated to form a patch-based SIFT-LBP descriptor.

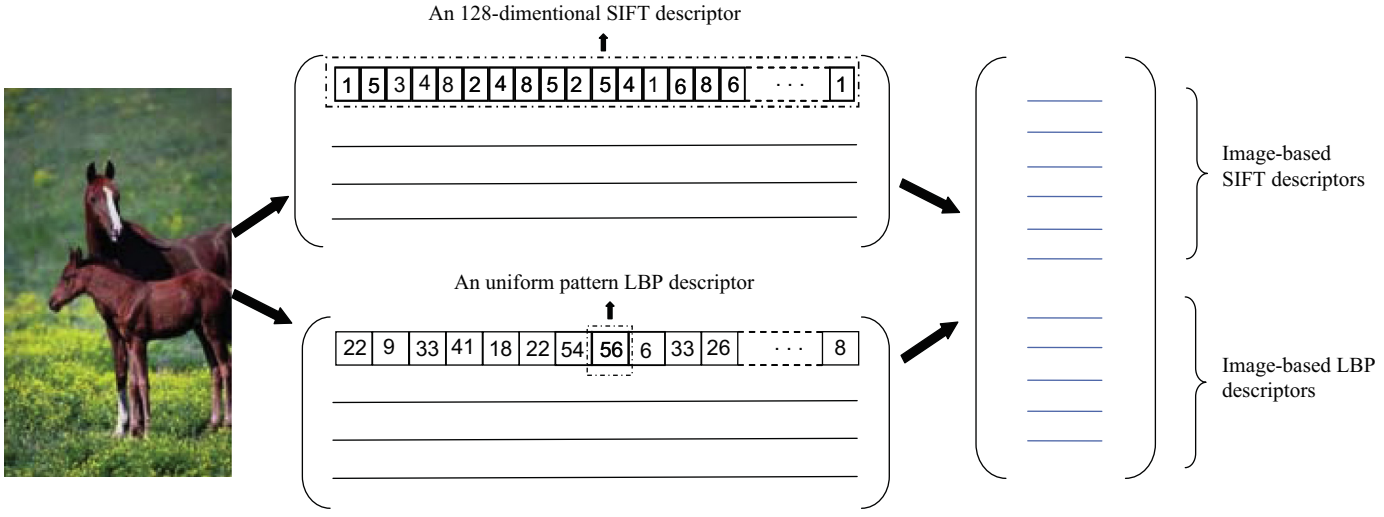


Fig. 4. The integration of the image-based SIFT-LBP. First, an 128-dimensional SIFT feature and an uniform pattern LBP feature are computed on the entire image; they are then concatenated to from an image-based SIFT-LBP descriptor; individual codebooks are built for the SIFT and LBP features by a weighted K-means clustering.

u means uniform pattern), we get a feature vector of 58 bins instead of original 256 bins. As the remaining un-uniform patterns are accumulated to a single bin, the uniform LBP histogram actually contains 59 bins. That is only a fraction (59/256) of the classic LBP descriptor.

As a dense version of SIFT feature, histograms of oriented gradients (HOG) feature has shown great performance in object presentation and recognition [25–29]. HOG is accepted to be one of the best features to describe the edge and shape information. Basic procedure of extracting HOG features is presented in Fig. 2.

3.1. SIFT-LBP features integration

The SIFT descriptor perform poorly when the background is lack of texture or corrupted with noise due to the fact the SIFT fails to find stable keypoints in these cases. The LBP with uniform patterns has been proven to be a very robust texture feature to supplement the SIFT by filtering out the noises [15]. We believe that the characteristics of an object in an image can be better captured by combining these two features. Thereby, two SIFT-LBP integrations methods are proposed at patch level and image level.

3.1.1. Patch-based SIFT-LBP feature integration

We define $p_i(x, y, \sigma, \theta)$ as a keypoint detected by SIFT approach, where (x, y) is the location of pixel p_i in the original image, σ and θ are the scale and main direction of p_i respectively. σ refers to the certain level of p_i in Gaussian Pyramid. Take a region with size of 16×16 as a patch where p_i is the center of the patch. The SIFT-LBP descriptor as shown in Fig. 3 is built as follows:

*Step*1. Use an 128-dimensional SIFT descriptor to describe each keypoint p_i in a patch, denoted as $SIFT_i$ in the image.

*Step*2. Choose an 8×8 region around p_i and compute the uniform pattern $LBP_{8,1}^{u_j}$ of each pixel. These descriptors are composed as a 64-dimensional vector, i.e.

$$LBP_i = [LBP_{8,1}^{u_1} \;\; LBP_{8,1}^{u_2} \;\; \cdots \;\; LBP_{8,1}^{u_{64}}]$$

*Step*3. LBP_i is directly connected to the end of $SIFT_i$, thus a patch can be described as an 192-dimensional integrated vector

$$SIFT\text{–}LBP_i = [SIFT_i \;\; LBP_i]$$

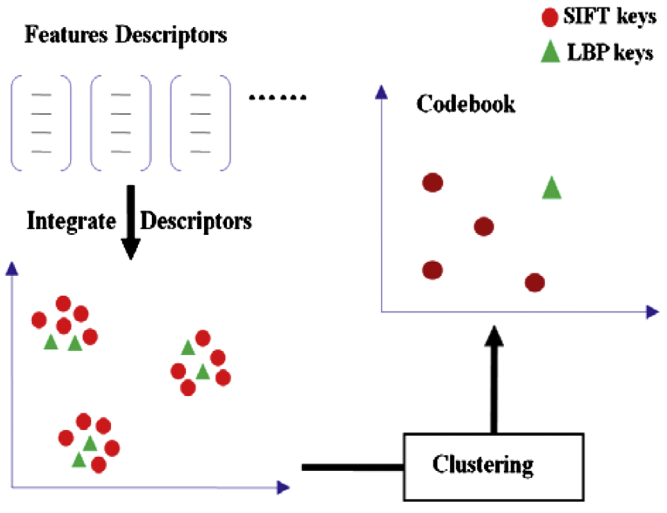


Fig. 5. The schematic diagram of BoF with SIFT-LBP using the K-means clustering. The vectors represent the independent SIFT and LBP descriptors. In the left bottom coordinate system, red circles and green triangles are SIFT and LBP keys, respectively. The circles in the right top coordinate system show the clustering results where the SIFT feature takes up majority of the cluster centers. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

3.1.2. Image-based SIFT-LBP feature integration

Patch-based SIFT-LBP may have a serious disadvantage that patches around keypoints are sparse and then much texture information is missing. Therefore, we propose to compute all the SIFT and LBP descriptors in an image independently and link them directly at the image level. The LBP-SIFT descriptor as shown in Fig. 4 is built as follows:

*Step*1. The same as *Step*1 in Section 3.1.1.

*Step*2. Compute the uniform pattern $LBP_{8,1}^{u}$ for each image pixel.

*Step*3. For each kind of feature, we independently build codebooks for SIFT and LBP features by weighted K-means clustering algorithms introduced below.

3.2. Weighted K-means clustering

K-means is one of the simplest unsupervised algorithm and has been widely used in image processing [5]. It is also used to cluster the SIFT descriptors to form codebook in the bag-of-feature model [20]. Fig. 5 schematic illustrated the BoF model with SIFT-LBP. The vectors represent independent SIFT and LBP descriptors. Given the above two integration methods, K-means can be applied directly to the patch-based approach. In the image-based integration, the number of LBP keys is much smaller than that of SIFT keys (e.g., given an image of 384×256. It contains 1457 SIFT keys and 384 LBP keys in a descriptor. It is quite possible that LBP features only take up small ratio of the total cluster centers. In this case, the SIFT features may play a key role in the codebook whereas the LBP features are less effective. We then use a weight parameter w $(0 \le w \le 1)$ to balance the importance between these two set of features

$$N_{SIFT} = w \cdot N \tag{2}$$

$$N_{LBP} = (1-w) \cdot N \tag{3}$$

where N is the desired number of cluster centers (the size of codebook). N_{SIFT} and N_{LBP} represent the number of cluster centers selected from SIFT keys and LBP keys, respectively. Then, we can control the weight of each feature in the codebook and develop the more effective integrated features through experiments. A schematic illustration of clustering with weighted K-means is shown in Fig. 6.

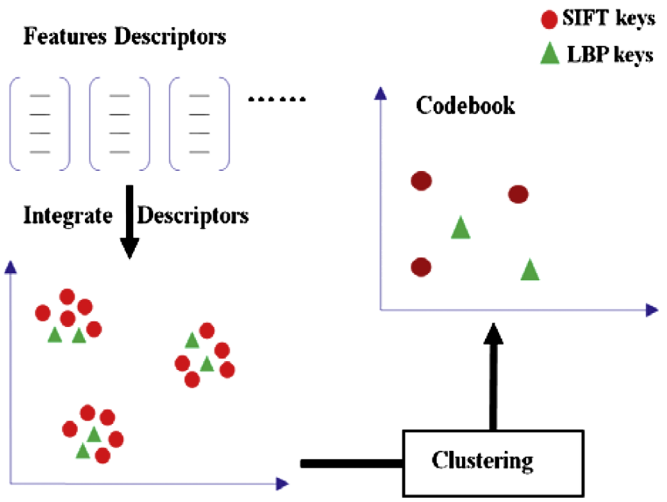


Fig. 6. The illustration of using a weighted K-means for codebook clustering. The LBP feature has a higher weights compared to Fig. 5.

3.3. HOG-LBP feature integration

As a dense version of SIFT feature, histograms of oriented gradients (HOG) feature have shown great performance in object presentation and recognition [25–29]. HOG is accepted to be one of the best features to describe the edge and shape information. While LBP, a texture feature, has the key advantages of invariance to monotonic grey level change and computational efficiency. An

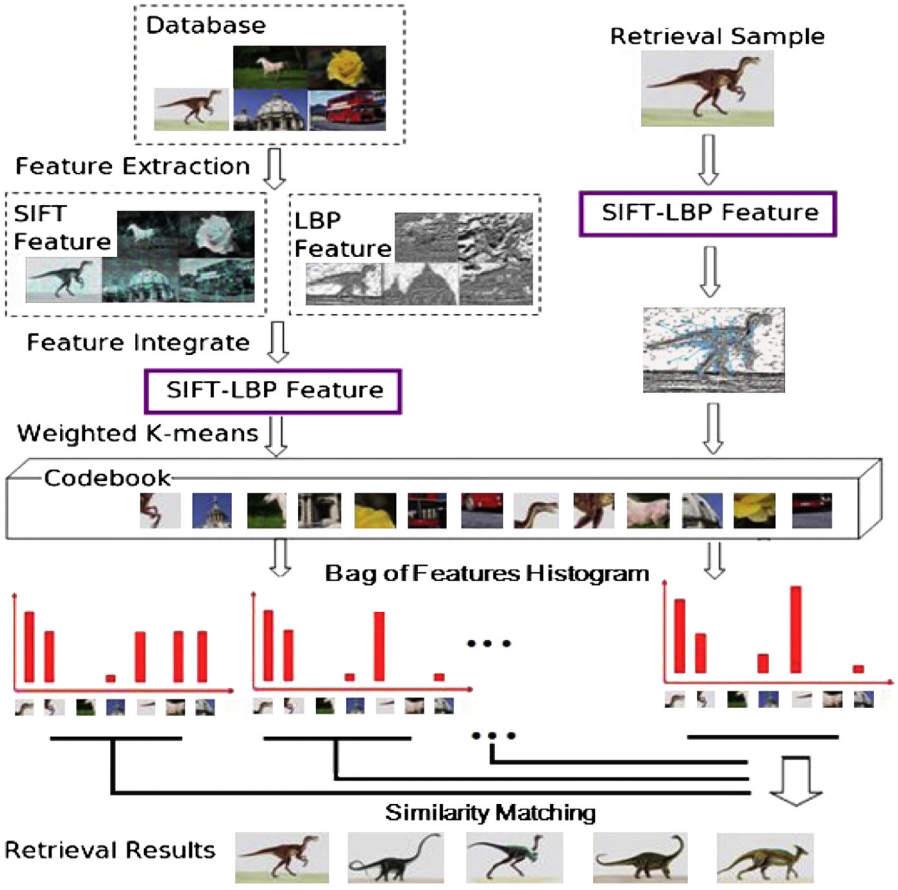


Fig. 7. The proposed framework for image retrieval based on the bag-of-features model using the SIFT-LBP integration. The SIFT and LBP features are extracted independently and combined by the proposed methods. A bag-of-features model is trained on these features. Given a new query image, the similarity between the BoF histograms of the query image and the ones in database is calculated to find the most similar images from the database.

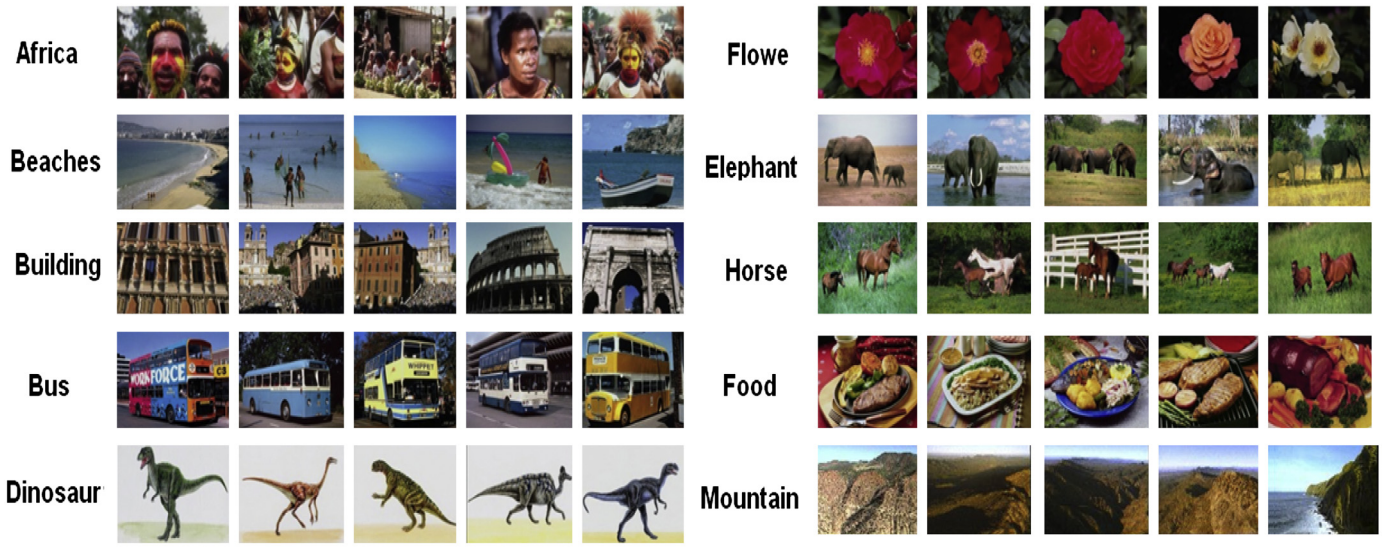


Fig. 8. Five samples from 10 categories of the Corel database [30].

augment feature vector, which integrating HOG and cell-based LBP, outperforms other state-of-the-art descriptors on human detection [16]. Inspired by this idea, we propose two kinds of HOG-LBP integration approaches at the image level and the patch level, and apply these features to image retrieval.

We follow the procedure of [25] to extract HOG features. Each image is divided into small cells of 4×4 pixel and for each cell we accumulate a 1–9 histogram of gradient directions. Each block is constructed by 2×2 cells. Then we refer to the normalized $2 \times 2 \times 9 = 36$ dimensional block descriptors as the HOG descriptors.

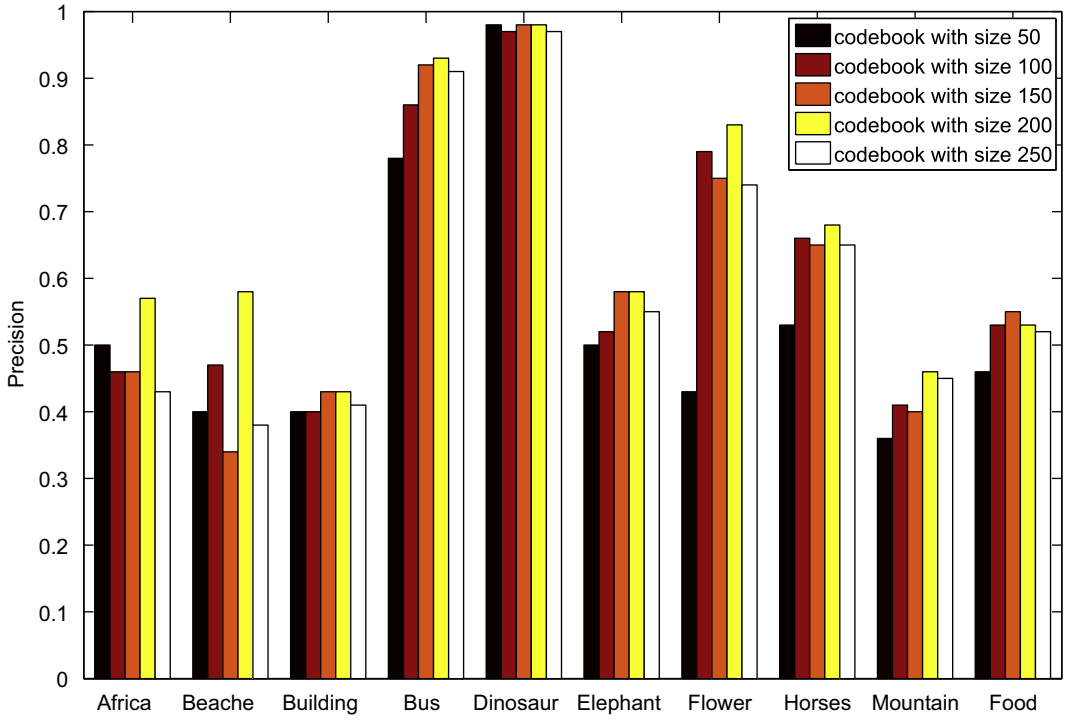


Fig. 9. Comparison of retrieval results with different codebook sizes. The ARP values for 10 categories with different codebook sizes are presented, and size 200 gives the best performance in most of the 10 categories except for *Food*.

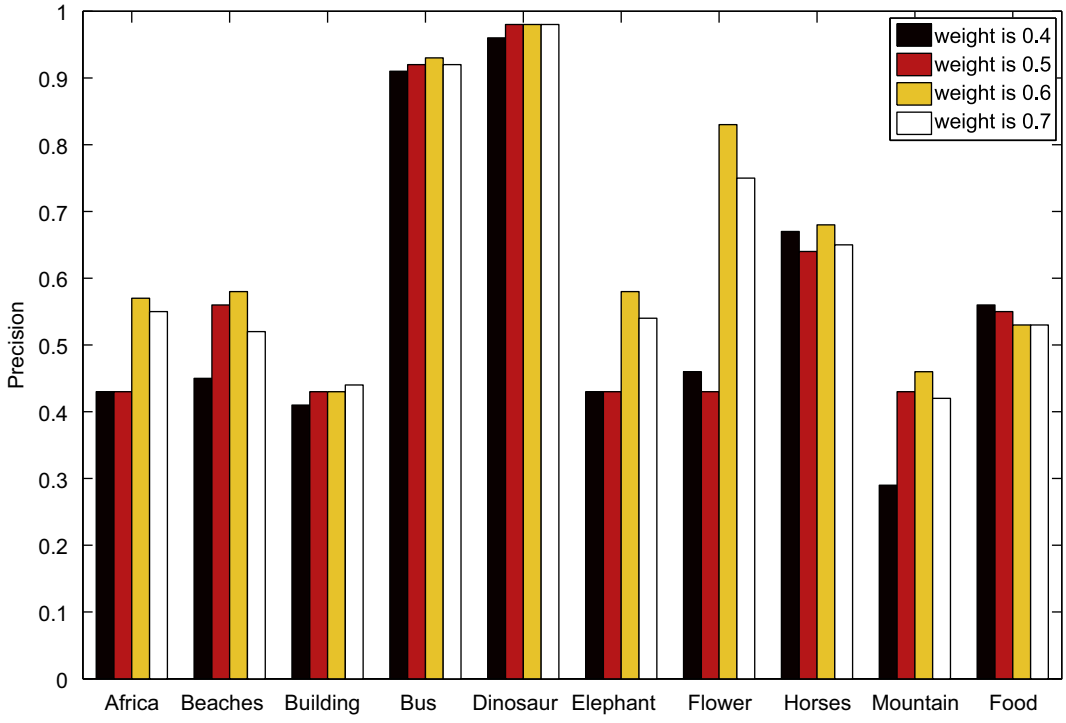


Fig. 10. Comparison of retrieval results with different choices of K-means weight in the image-based integration. The ARP values for 10 categories with different weights are presented by a histogram plotting. The performance differs in: *Flower*, the weight plays an important role where the ARP could vary as big as 30% by using different weights; however, in categories like *Bus* and *Dinosaur*, the difference is as small as 3%.

3.3.1. Patch-based HOG-LBP integration

Here we define each block of the HOG feature as a patch. For the construction of block-based LBP, we build uniform pattern LBP histogram in each block. The integration is to directly link the HOG descriptor and uniform LBP descriptor in the same block. Detailed procedure is as follows:

*Step*1. Extract a 36-dimensional HOG descriptor in each patch, denoted as HOG_i. The building procedure is shown in Fig. 2.

*Step*2. Choose an 8×8 region around the center of each patch and compute the uniform pattern $LBP_{8,1}^{u_j}$ of each pixel. These descriptors are composed as a 64-dimensional vector, i.e.

$$LBP_i = [LBP_{8,1}^{u_1} \; LBP_{8,1}^{u_2} \; \cdots \; LBP_{8,1}^{u_{64}}]$$

*Step*3. In each patch, LBP_i is directly linked to the end of HOG_i. So far, a patch-based 100-dimensional descriptive vector is built

$$[HOG\text{–}LBP_i] = [HOG_i \; LBP_i]$$

Table 1
Comparison of the ARP values obtained by the proposed methods with the standard image retrieval systems.

Class	Color, texture shape based [4]	Block based LBP [8]	BoF based SIFT [21]	SPM based SIFT [33]	Patch-based SIFT-LBP	Image-based SIFT-LBP	Patch-based HOG-LBP	Image-based HOG-LBP
Africa	0.48	0.23	0.55	0.61	0.54	0.57	0.55	0.29
Beaches	0.34	0.23	0.47	0.49	0.39	0.58	0.47	0.43
Building	0.36	0.23	0.44	0.46	0.45	0.43	0.56	0.30
Bus	0.61	0.23	0.93	0.93	0.80	0.93	0.91	0.66
Dinosaur	0.95	0.23	0.98	0.99	0.93	0.98	0.94	0.97
Elephant	0.48	0.23	0.52	0.58	0.30	0.58	0.49	0.36
Flower	0.61	0.23	0.77	0.83	0.79	0.83	0.85	0.52
Horses	0.74	0.23	0.65	0.65	0.54	0.68	0.52	0.55
Mountain	0.42	0.23	0.34	0.36	0.35	0.46	0.37	0.29
Food	0.50	0.23	0.52	0.51	0.52	0.53	0.55	0.22
Total ARP	0.549	0.230	0.617	0.641	0.561	0.657	0.621	0.459

3.3.2. Image-based HOG-LBP integration

It is also possible to combine HOG and LBP features at the image level. In this case, HOG and LBP features are extracted independently of the whole image, and integration is finished within our newly proposed weighted K-means clustering. The detailed procedure is as follows:

*Step*1. The same as *Step*1. in patch-based HOG-LBP integration.

*Step*2. Compute the uniform pattern LBP descriptors for each image pixel in the whole image.

*Step*3. For each image, we build codebook for above HOG and LBP features using weighted K-means clustering whose principle idea is the same as the method introduced in Section 3.2.

4. Experimental studies

A schematic diagram of the proposed framework for image retrieval is shown in Fig. 7. In the training process (shown on the left-hand side of Fig. 7), the SIFT and LBP features are extracted and integrated to construct a SIFT-LBP descriptor. After weighted K-means clustering, the codebook is then generated using the integrated descriptor. Each image is then mapped to the codebook in order to obtain its BoF histogram. In the retrieval process (shown on the right-hand side of Fig. 7), we select a query image from the given database. By comparing its BoF histogram to the other BoF histograms in the database, we can yield a ranked set of the most similar images. For the HOG-LBP integration, we simply replace the SIFT feature with the HOG feature and the remaining process is untouched.

We first conduct experiments using the Corel database [30]. The sample images are displayed in Fig. 8. We analyze the performance by considering the different codebook sizes and K-means weights. The subsequent subsections focus on comparing our new algorithms with a number of state-of-the-art image retrieval methods on two different databases. For the purpose of fair comparison, we adopt the same evaluation criteria reported in [8,32].

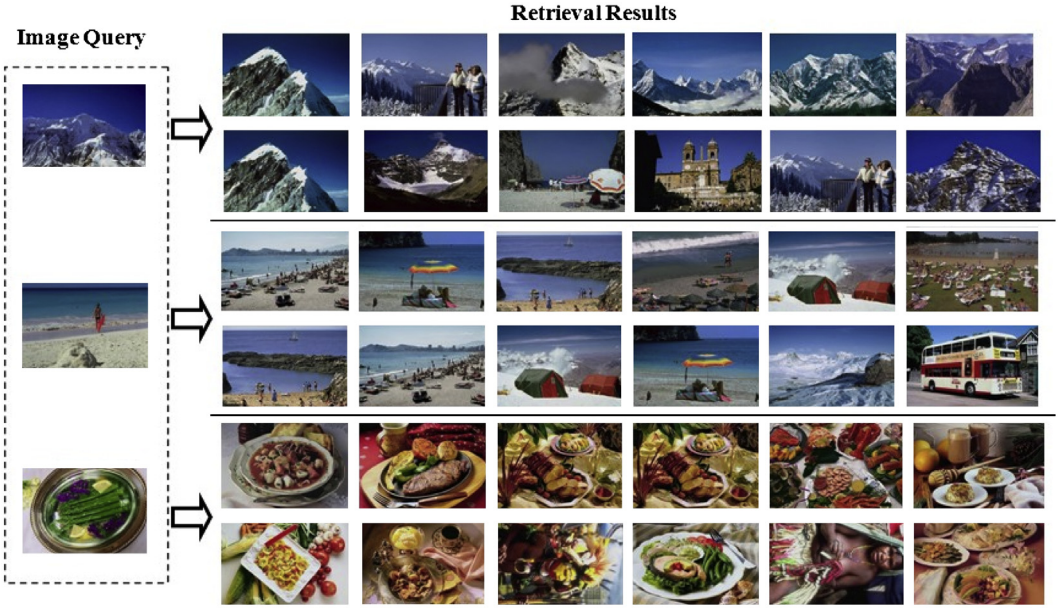


Fig. 11. Comparison of retrieval results obtained by: (1) the image-based SIFT-LBP integration method, and (2) the SPM based SIFT method. The left side lists three query images. The right side shows two rows of retrieval results corresponding to each query image, the first and the second rows are the results from method (1) and method (2), respectively.

4.1. Effectiveness of the codebook size and K-means weight

The Corel database is utilized in this experiment which comprises 1000 images from 10 categories [30]. The images are with the size of 256×384 or 384×256 pixels. The average retrieval precision (ARP) [4] is used to quantify the performance. We define $A(i)$ as a set of images having the same category index with the query image i in the database, and $B(i)$ is a collection of retrieved images. The precision is defined as the percentage of the retrieved images belonging to the same category as the query image

$$P(i) = \frac{|A(i) \bigcap B(i)|}{|B(i)|} \tag{4}$$

If we choose a dinosaur image from the database as a query, and 70 of the first 100 (we totally have 100 dinosaur images) retrieved images belong to the category of dinosaur. The retrieval precision is 0.7. The ARP of a specific category is defined by

$$ARP(ID_m) = \frac{1}{N} \sum_{id(i) = ID_m} P(i) \tag{5}$$

where N is the size of the category ID_m in the testing database and $id(i)$ is the category index for the query image i.

We study the effectiveness of different sizes of codebook and K-means weights on the performance for the image-based SIFT-LBP descriptor. We choose the size of codebooks from $\{50, 100, 150, 200, 250\}$. The comparison results are shown in Fig. 9. By examining the figure, we can see that size 200 gains the best results on nine out of 10 categories. Given the best size of codebook, we test different values of weight $w \in \{0.4, 0.5, 0.6, 0.7\}$. The results shown in Fig. 10 indicate that $w = 0.6$ outperforms all the other weights. The ARP values for each of the 10 categories are listed in Table 1. For the patch-based SIFT-LBP integration, we have tested patch region with size of 16×16 and 8×8 with different codebook sizes. We found that the best results are given by the patch size of 8×8 and $N = 200$, which are shown in Table 1.

4.2. Image retrieval performance

We compare the retrieval precision of the proposed methods to the existing methods using the Corel database [30]. An integrated matching scheme combining color, texture, and shape features [4], which has been reported to be a robust retrieval system, is included for comparison. A block-based LBP method [8] and a BoF-based SIFT approach [21] are also used as reference methods. The spatial pyramid matching (SPM) [33] is one of the most effective extensions of BoF. The SPM divides an image into small sub-regions, and counts the number of features locating in each region. Here we compare our proposed feature integration schemes with the SPM-based SIFT retrieval system.

Experimental results shown in Table 1 indicate that SPM using the SIFT feature improves the retrieval performance on almost all the categories compared with the BoF model. We note that the SPM gains the best performance on "Africa" which contains more human figures than other categories except "Food" because the spatial information is effective in discriminating human shapes. The poor performance from the block-based LBP is due to the fact that the method only takes into account the texture information.

Our proposed patch-based SIFT-LBP scheme outperforms the color, texture, and shape integration [4] and block-based LBP [8], but is worse than the BoF and SPM. This indicates that the patch-based LBP has a less discriminant capability. The patch-based HOG-LBP integration significantly improves the results compared to the image-based HOG-LBP. The image-based SIFT-LBP achieves the best retrieval result among all these methods.

As shown in Fig. 11, we can see that the image-based SIFT-LBP method yields more meaningful matching results than the SPM method. It is clear to see that by integrating SIFT and LBP features at image level allows us to achieve superior retrieval results in

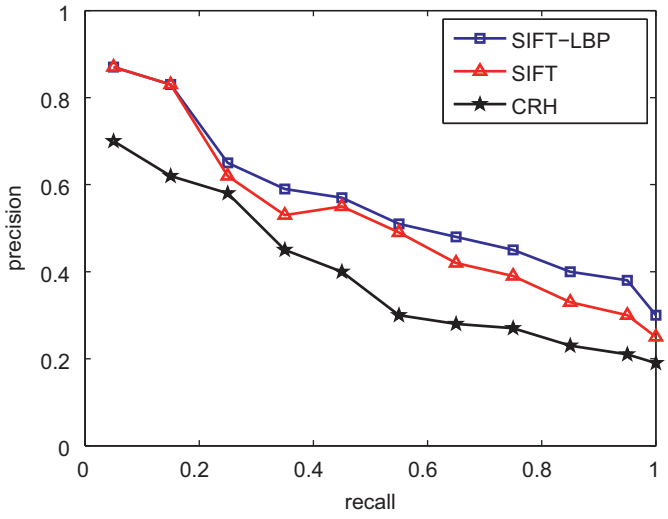


Fig. 13. Comparison of the precision and recall rate of the CRH [32], the BoF model based on SIFT [21], and the BoF model based on the image-based SIFT-LBP integration.

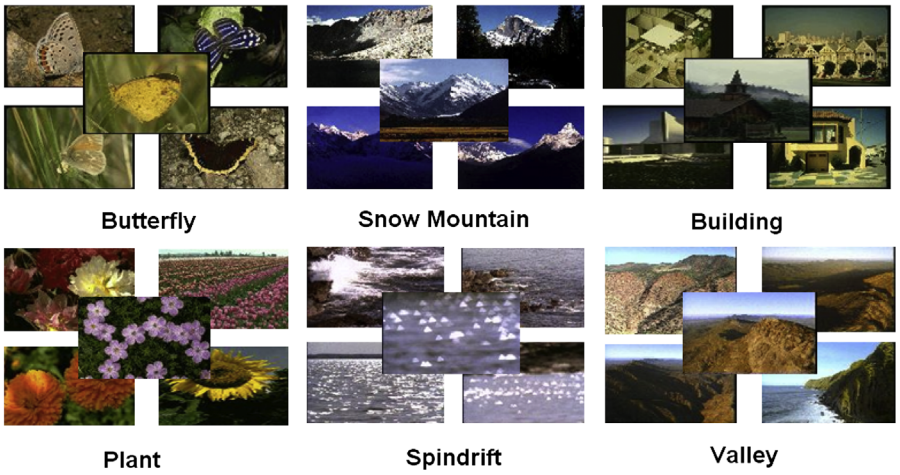


Fig. 12. Five samplers from six categories of the SIMPLIcity database [31].

terms of visual and quantitative evaluations. To further verify the performance of the integrated SIFT-LBP feature, we test on the SIMPLIcity database [31]. It contains 500 images with size 128×85 from six classes based on their semantic contents. Five sample images for six categories are shown in Fig. 12. We use the precision-recall (PR) curve [32] to evaluate the retrieval results. The precision is defined in Eq. 4. The recall represents the percentage of relevant images that are retrieved, which is expressed as

$$R(i) = \frac{|A(i) \bigcap B(i)|}{|A(i)|} \tag{6}$$

Fig. 13 shows the precision-recall curve on the three models: the circular ring histogram (CRH) [32], the BoF model based on SIFT [21], and the BoF model based on SIFT-LBP. The CRH is a histogram-based method that is robust to the image rotation and scaling [32]. The BoF model using the SIFT feature outperforms the CRH method. Again, the image-based SIFT-LBP has shown the best performance for image retrieval.

It is noted that our feature extraction module takes a longer time than computing one single feature since there are two features (SIFT-LBP, HOG-LBP) to be computed. However, all the features are extracted before the feature integration and image retrieval tasks. In fact, the integration and similarity measure modules are performed separately and run fast.

5. Conclusions

In this paper, we proposed a feature integration framework for image retrieval based on the bag of features model and weighted K-means clustering. Patch-based and image-based integration of SIFT, LBP and HOG features are proposed. Based on comprehensive experimental studies on benchmark image retrieval problems, the image-based integration achieves the best performance compared to the existing models when adopting codebooks size $N=200$ and K-means weight $w=0.6$. The future work will focus on evaluation of more real-world image retrieval problems and improvement on the ranking quality by incorporating the relations between these features.

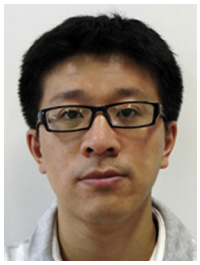
Acknowledgment

This work is partially funded by the NCET Program of MOE, China, the SRF for ROCS and the China Scholar Council.

References

[1] Y. Pang, M. Shang, Y. Yuan, J. Pan, Scale invariant image matching using triplewise constraint and weighted voting, Neurocomputing 83 (2012) 64–71.
[2] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Comput. Surv. 40 (2008) 1–60.
[3] X. Yuan, J. Yu, Z. Qin, T. Wan, A SIFT-LBP image retrieval model based on bag-of-features, in: ICIP, 2011, pp. 1161–1164.
[4] J. Pujari, P.S. Hiremath, Content based image retrieval using color, texture and shape features, in: Proceedings of the International Conference on Advanced Computing and Communications, 2007, pp. 780–784.
[5] T. Wan, Z. Qin, A new technique for summarizing video sequences through histogram evolution, SPCOM (2010) 1–5.
[6] H. Yan, Y. Yuan, K. Wang, Robust CoHOG feature extraction in human-centered image/video management system, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 42 (2012) 458–468.
[7] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Trans. Syst. Man Cybernet. 8 (1978) 460–473.
[8] M. Pietikäinen, T. Ahonen, V. Takala, Block-based methods for image retrieval using local binary patterns, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, 2005, pp. 882–891.
[9] D.G. Lowe, Object recognition from local scale-invariant features, in: ICCV, 1999, pp. 1150–1157.
[10] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: CVPR 85, 2004, pp. 506–513.
[11] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, SURF: speeded-up robust features, Comput. Vis. Image Understand. 110 (2008) 346–359.
[12] Y. Pang, W. Li, Y. Yuan, J. Pan, Fully affine invariant SURF for image matching, Neurocomputing 85 (2012) 6–10.
[13] Y. Pang, Y. Yuan, X. Li, J. Pan, Efficient HOG human detection, Signal Process. 91 (2011) 773–781.
[14] D. Ojala, M. Pietikäine, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 19 (1996) 51–59.
[15] D. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987.
[16] X. Wang, T. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, in: ICCV, 2009, pp. 32–39.
[17] M. Heikkilä, M. Pietikäinen, C. Schmid, Description of interest regions with local binary patterns, Pattern Recognition 42 (2009) 425–436.
[18] Y. Zheng, X. Huang, S. Feng, An image matching algorithm based on combination of SIFT and rotation invariant LBP, J. Computer-Aided Design Comput. Graph. 2 (2010) 286–292.
[19] W.R. Schwartz, A. Kembhavi, D. Harwood, L.S. Davis, Huam detection using partial least squares analysis, in: ICCV, 2009, pp. 24–31.
[20] J. Fehr, A. Streicher, H. Burkhardt, A bag of features approach for 3D shape retrieval, in: Proceedings of the Fifth International Symposium on Advances in Visual Computing: Part I, 2009, pp. 34–43.
[21] B. Triggs, F. Jurie, Creating efficient codebooks for visual recognition, in: ICCV, vol. 1, 2005, pp. 604–610.
[22] Z. Qin, M. Thint, Z. Huang, Ranking answers by hierarchical topic models, in: Proceedings of IEA/AIE LNCS, vol. 5579, 2009, pp. 103–112.
[23] Q. Tian, S. Zhang, Descriptive visual words and visual phrases for image applications, ACM Multimedia (2009) 19–24.
[24] C. Schmid, K. Mikolajczyk, A performance evaluation of local descriptors, in: ICPR, vol. 2, 2003, pp. 257–263.
[25] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, vol. 1, 2005, pp. 886–893.
[26] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: ECCV, 2006, pp. 428–441.
[27] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627–1645.
[28] P. Sabzmeydani, G. Mori, Detecting pedestrians by learning shapelet features, in: CVPR, 2007, pp. 1–8.
[29] Q. Zhu, M.C. Yeh, K.T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, in: CVPR, 2006, pp. 1491–1498.
[30] J.A. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 947–963.
[31] J.Z. Wang Group: Modeling Objects, Concepts, and Aesthetics in Images, website: ⟨http://wang.ist.psu.edu/⟩.
[32] X. Wang, A novel circular ring histogram for content-based image retrieval, ETCS, 2009, pp. 1–4.
[33] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: CVPR, 2009, pp. 1794–1801.

Jing Yu received her B.S. degree in Automation Science from MinZu University, Beijing, China, in 2011. She is currently working toward her M.S. degree in Pattern Recognition and Intelligent Systems from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. She used to work as an intern in Samsung Electronics (China) R&D Center and Microsoft Research Asia. Her research interests include multimedia information retrieval, computer vision and digital image processing.



Zengchang Qin is an associate professor in Beihang University, Beijing, China. He obtained his MSc and PhD from University of Bristol, UK, and did his postdoc research with Lotfi Zadeh in UC Berkeley, US. He used to work (or intern) in HP, BT, Optimor Labs and worked as a visiting scholar in University of Oxford and Carnegie Mellon University. His research interests are agent-based modeling, machine learning, computational intelligence and multimedia retrieval.

Tao Wan is working as a researcher associate at the Case Western Reserve University. She was postdoctoral associate at Boston University's Section of Radiology in the School of Medicine. She received her Master degree in Global Computing and Multimedia from the University of Bristol, UK in 2004 and her Ph.D. in Computer Science from the same university in 2009. She spent one year working as a senior researcher in the Samsung Advanced Institute of Technology (SAIT) China before becoming a visiting scholar in the Visualization and Image Analysis Lab in the Robotics Institute, Carnegie Mellon University. Her research interests are statistical models for image segmentation, fusion, and denoising, machine learning, computer-aided diagnosis system, medical image analysis on prostate and breast cancer.

Zhang Xi was born in Langfang, Hebei Province, in 1990. She was enrolled by Beihang University, Beijing, China, in 2009. Currently she is a senior student in School of Automation Science and Electronic Engineering, Beihang University. Her research interests include image retrieval, guidance, optimal control system and fuzzy control system.