A SIFT-LBP IMAGE RETRIEVAL MODEL BASED ON BAG-OF-FEATURES

Xiaoli Yuan [†], *Jing Yu* [♯], *Zengchang Qin*[†‡] , *and Tao Wan* [‡]

[†] Intelligent Computing and Machine Learning Lab, School of ASEE, Beihang University, China
[‡] Robotics Institute, Carnegie Mellon University, USA
[♯] School of Information Engineering, Minzu University, Beijing, China

ABSTRACT

Despite progress in image retrieval by using low-level features, such as colors, textures and shapes, the performance is still unsatisfied as there are existing gaps between low-level features and high-level semantic concepts (semantic gaps). In this research, we propose a novel image retrieval system based on bag-of-features (BoF) model by integrating scale invariant feature transform (SIFT) and local binary pattern (LBP). We show that SIFT and LBP features yield complementary and substantial improvement on image retrieval even in the case of noisy background and ambiguous objects. Two new integration models are proposed: patch-based integration and image-based integration. By using a weighted K-means clustering algorithm, the image-based SIFT-LBP integration achieves the superior performance on a given benchmark problem comparing to other existing algorithms.

Index Terms— bag-of-features (BoF), image retrieval, SIFT-LBP, weighted K-means, histogram intersection.

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 [2, 6, 13]. Low-level features such as colors, textures, and shapes have been used to describe the image content [4]. The main idea 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. 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 the image content and spatial distribution of colors. Texture features can describe spatial variations in pixel intensities and the surface characteristics of an object. But texture segmentation still remains a difficult problem to meet human perception [5]. Shape-based features are relatively consistent with the intuitive feeling but lacking perfect mathematical foundations to deal with the target deformation [6]. Therefore, only using low-level visual features can hardly describe the semantic concepts of images.

In recent years, mid-level features have attracted more attentions. Among them, SIFT [3] features are invariant to rotation, scaling, translation and small distortions. LBP features [4] are considered as one of the best texture features as they are invariant to monotonic changes in gray-scale, fast to calculate and also complementary for some disadvantages of the SIFT features. SIFT has been empirically proven to be one of the most robust among the local invariant feature descriptors with respect to different geometrical changes [8]. It represents blurred image gradients in multiple orientation planes and at multiple scales. SIFT has shown great success in object recognition and detection due to its invariance in translation, scaling, rotation, and small distortions. The basic idea is to look for the extreme points in the scale space, filter these extreme points to find the stable feature points known as *keypoints*, and finally compute local attribution of orientation gradient and describe the keypoints by $4 \times 4 \times 8$ vectors. The local binary pattern (LBP) operator [4] is a texture descriptor that has been widely used in object recognition and achieved good performance in face recognition. Previous research used uniform patterns representing the most essential texture information showed a strong discriminative ability [4]. Based on their advantages, Helkklla *et al.* [1] recently proposed a novel region descriptor by combining SIFT and LBP features. In this research, we will investigate how to use integrated SIFT-LBP features in image retrieval.

In this paper, we design the local semantic descriptors by integrating SIFT and LBP features. The new features do not rely on the image segmentation and are able to automatically detect interest points and regions in an image. The proposed image model is based on the bag-of-feature representations [9]. Features are computed for each image to form a high-dimensional descriptor. These descriptors are clustered into several key points which are referred to as visual words. Each image is then represented by a distribution on visual words. Given a query image, we aim to find a list of similar images which are ranked by similarity scores based on visual vocabulary distributions.

[‡] Corresponding author email: zcqin@buaa.edu.edu.cn. This work is partially funded by the NCET Program from MoE China and the China Scholar Council.

978-1-4577-1303-3/11/$26.00 ©2011 IEEE

2. BAG-OF-FEATURES MODEL

The bag-of-features (BoF) method is largely inspired by the bag-of-words (BoW) [11] concept which has been used in text mining. In the BoW model, each word is assumed to be independent, though it is very counterintuitive, it has been well-used in spam filtering and topic modeling [7] with outstanding performance. In the BoF model, each image is described by a set of orderless local features, recent research has demonstrated its effectiveness in image processing [10]. It has two key concepts: *local features* and *codebook*.

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 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.

Codebook Representation: Codebook is a way that an image can be represented by a set of local features [9]. The idea is to cluster the feature descriptors of all patches based on a given cluster number and each cluster represents a visual word that will be used to form the codebook [11]. After obtaining the codebook, each image can be represented by the BoF frequency histograms of the visual vocabulary of the codebook. The similarity of images can be measured by comparing between the BoF histograms.

In this research, histogram intersection is used to compute the similarity between two histograms of given images A and B. The histogram intersection is defined by:

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

where a_i and b_i represents the frequencies 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. Consequently, a set of images in database with small similarity distance is selected and ranked from the most to the least similar ones.

3. SIFT-LBP FEATURES INTEGRATION

SIFT may perform poorly when the background is complex or corrupted with noise, LBP with uniform patterns is complementary to SIFT by filtering out these noises [4]. 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 in patch level and image level, respectively.

3.1. Patch-based SIFT-LBP feature integration

We define $p_i\left(x,y,\sigma,\theta\right)$ as a keypoint detected by SIFT approach, where (x,y) is the location of pixel p_i in the original image, σ and θ is 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, then the SIFT-LBP descriptors are built as follows:

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

Step 2. Choose a 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 a 192-dimensional integrated vector: $SIFT\text{-}LBP_i = [SIFT_i\ LBP_i]$

3.2. Image-based SIFT-LBP feature integration

Because the patches around keypoints are sparse and then much texture information could be missing. Therefore, we propose another method of integrating by computing SIFT and LBP descriptors of an image independently and link them at an image level. These LBP-SIFT descriptors are built as follows:

Step 1. The same as *Step 1* in Section 3.1.

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

Step 3. For each image, we independently build codebooks for SIFT and LBP features by using the weighted K-means clustering algorithms introduced below.

3.3. Weighted K-means clustering

K-means clustering is one of the simplest unsupervised algorithm that has been widely used in image processing [14]. It is also used to cluster the SIFT descriptors to form a codebook in the bag-of-feature model [9]. K-means can be applied directly to the patch-based integration. In the image-based integration, the number of LBP keys is much smaller than that of SIFT keys (e.g., give a 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 a small percentage of the total cluster centers. In this case, the SIFT features may be dominated in the codebook whereas the LBP features are less effective. We then use a weight parameter w $(0 \leq w \leq 1)$ to balance the importance between these two sets of features.

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

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

where N is the predefined number of cluster centers (size of codebook). N_{SIFT} and N_{LBP} are the number of cluster centers selected from SIFT keys and LBP keys, respectively. We can adjust the weight of each feature in the codebook and construct a most effective integrated feature set.

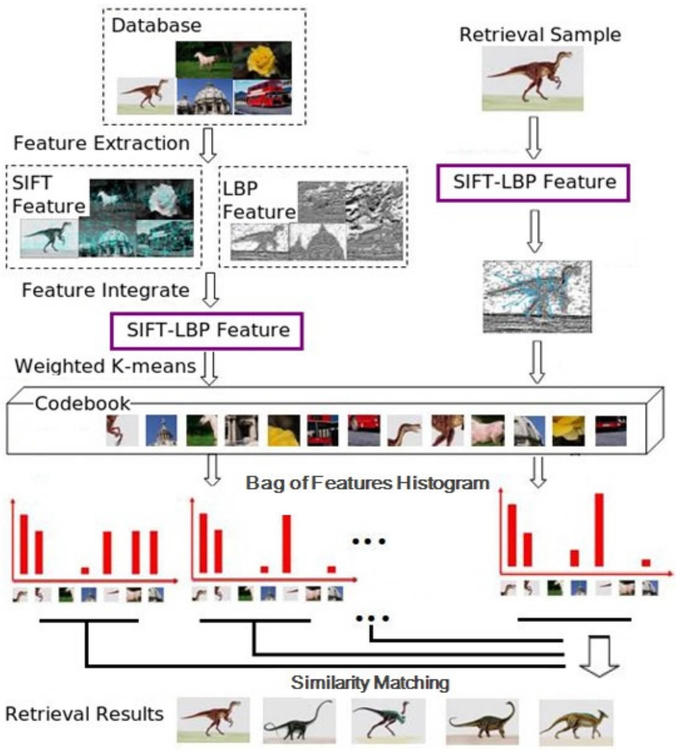

Fig. 1: The proposed framework of image retrieval based on the bag-of-features model using SIFT-LBP integrations. SIFT and LBP features are extracted independently and integrated by using proposed methods. A bag-of-features model is trained on these integrated features. Given a new query image, the similarity between the BoF histograms of the query image and the ones in database is calculated in order to find the most similar images from the database.

4. EXPERIMENTAL STUDIES

A schematic illustration of the experiment is shown in Fig. 1. In the training process (the left-hand side of Fig. 1), SIFT and LBP features are extracted from each image in the database, and then be integrated for constructing the SIFT-LBP descriptors (in two possible ways). After the weighted K-means clustering, the codebook is generated by the integrated SIFT-LBP descriptors. Each image is mapped to the codebook in order to obtain its BoF histogram. In the retrieval process (the right-hand side of Fig. 1), we input a query image, by comparing its BoF histogram and other BoF histograms in the database, we can obtain a ranked set of most similar images based on Eq. (1).

In our experiments, we test a benchmark image dataset Corel [12] that comprises 1000 images from 10 categories. The images are with the size of either 256×384 or 384×256. The quantitative measure we use is average precision [6] which is defined by:

Table 1: Comparisons of average retrieval precision (ARP) obtained by two proposed methods to other existing image retrieval systems.

Class	Color, texture, shape based [6]	Block based LBP [5]	BoF based SIFT	Patch-based SIFT-LBP	Image-based SIFT-LBP
Africa	.48	.23	.55	.54	.57
Beaches	.34	.23	.47	.39	.58
Building	.36	.23	.44	.45	.43
Bus	.61	.23	.93	.80	.93
Dinosaur	.95	.23	.98	.93	.98
Elephant	.48	.23	.52	.30	.58
Flower	.61	.23	.77	.79	.83
Horses	.74	.23	.65	.54	.68
Mountain	.42	.23	.34	.35	.46
Food	.50	.23	.52	.52	.53
Total ARP	.549	.230	.617	.561	.657

$$P(i) = \frac{1}{M}\sum_{j=1}^{M}\gamma(i,j) \tag{4}$$

where

$$\gamma(i,j) = \begin{cases} 1 & id(i) = id(j) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $P(i)$ is precision of query image i, $id(i)$ and $id(j)$ are the category ID of image i and j, respectively, which are in the range of 1 to 10. M is the original size of the category that image i is from. This value is the percentage of images belonging to the category of image i in the first M retrieved images. For example, if the query image is a dinosaur (Fig. 1), if 70 of the first 100 (there are 100 dinosaur images in the training set) retrieved images are belonging to the category of dinosaurs, then the retrieval precision is 0.7.

First, we study the influence of the codebook size on retrieval performance of the system. We choose the size of codebooks from $\{50, 100, 150, 200, 250\}$. The performance is shown in Fig. 2. As we can see from the results, the best size is 200 for this data set. In image-based integration experiments, we tested different value of weight by allowing w equals to the following numbers: $\{0.4, 0.5, 0.6, 0.7\}$ given codebook size equals 200. The results are shown in Fig. 3 from which we can see that $w = 0.6$ outperforms all the other weights. The detailed results of using two integration methods for each of the 10 categories are shown in Table 1. For the patch-based SIFT-LBP integration, we tested patch region of 16×16 and 8×8 with different size of codebook, the best results are obtained with patch size of 8×8 and $N = 200$ are listed in Table 1.

The new proposed patch-based SIFT-LBP scheme generally performs better than the previous two methods [5, 6].

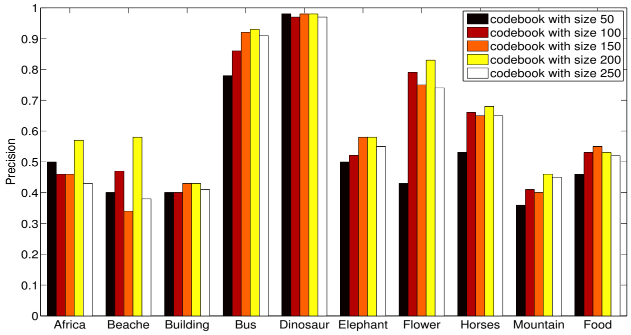

Fig. 2: Comparison of retrieval results with different number of codebook size. Average retrieval precision (ARP) for 10 classes with different codebook size are presented using histograms. Codebook size of 200 gives the best performances in most of the 10 categories except for the category of *Food*.

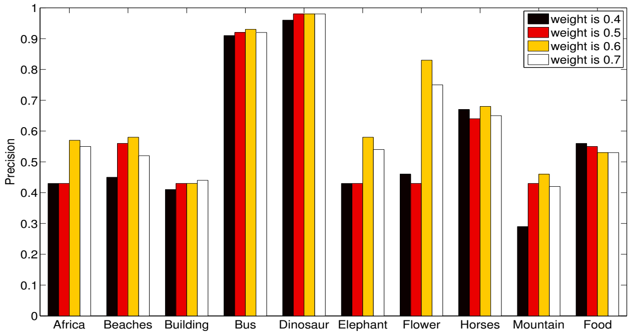

Fig. 3: By applying the weighted K-means clustering, the average retrieval precision with different weights are presented as histograms. The difference of performance is category dependent: in class *Flower*, the weight plays an important role where the ARP could vary as big as 30% with different weight; however, in class like *Bus* and *Dinosaur*, the difference is really small, it is within 3% by choosing different weights.

However, its average performance is lower than BoF model using SIFT only. That gives us a clue that adding LBP features in patch-based scenario may not provide more discrimination than using the SIFT features only. It is also very obvious to see that if we integrate these two sets of features at image level, we can achieve the best performance.

5. CONCLUSIONS

In this paper, we have proposed a novel method for image retrieval based on the bag-of-features model. The SIFT and LBP features are integrated in two levels: patch-level and image-level. Based on the experimental studies on a benchmark image retrieval problem, the image-based integration gives the best performance comparing to other existing models when adopting codebooks size $N = 200$ and K-means weight w=0.6, which means that the SIFT features and LBP features taking the relative importance of 0.6 and 0.4, respectively. The optimal parameters setting for this system will be the future work.

6. REFERENCES

[1] M. Heikkila, M. Pietikainena and C. Schmid, “Description of interest regions with local binary patterns”, *Pattern Recognition*, Vol. 42:3, pp. 425-436, 2009.

[2] D. Joshi, J. Li, J. Wang, and R. Datta,“Image retrieval: ideas, influences, and trends of the new age,” *Proceedings of the 7th ACM SIGMM on Multimedia Information Retrieval*, 2005.

[3] D.G. Lowe,“Object recognition from local scale-invariant features,” *ICCV*, pp. 1150-1157, 1999.

[4] D.T. Ojala, M. Pietikinen, and T. Maenpaa, “Multiresolution gray scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans on PAMI*, vol. 24, pp. 971-987, 2002.

[5] M. Pietikainen, T. Ahonen, and V. Takala, “ Block-based methods for image retrieval using local binary patterns,” *Proceesings of the 14th Scandinavian Conference on Image Analysis*, pp. 882–891, 2005.

[6] J. Pujari and P.S. Hiremath, “Content based image retrieval using color,texture and shape features,” *Proceedings of the International Conference on Advanced Computing and Communications*, pp. 780-784, 2007.

[7] Z. Qin, M. Thint and Z. Huang, “Ranking answers by hierarchical topic models,” *Proceedings of IEA/AIE*, LNCS 5579, pp. 103-112, 2009.

[8] C. Schmid and K. Mikolajczyk, “A performance evaluation of local descriptors,” *ICPR*, vol. 2, pp. 257-263, 2003.

[9] A. Streicher, H. Burkhardt, and J. Fehr, “A bag of features approach for 3D shape retrieval,” *International Symposium on Visual Computing*, 2009.

[10] Q. Tian, S. Zhang, “Descriptive visual words and visual phrases for image applications,” *ACM Multimedia*, pp. 19-24, 2009.

[11] B. Triggs and F. Jurie, “Creating efficient codebooks for visual recognition,” *ICCV*, vol. 1, pp. 604-610, 2005.

[12] http://wang.ist.psu.edu/ jwang/test1.tar

[13] T. Wan and Z. Qin, “An application of compressive sensing for image fusion,” *CIVR*, pp. 3-9, 2010.

[14] T. Wan and Z. Qin, “A new technique for summarizing video sequences through histogram evolution,” *SPCOM*, pp. 1-5, 2010.