Evaluation of Global Descriptors for Large Scale Image Retrieval

Hai Wang and Shuwu Zhang

Institute of Automation Chinese Academy of Sciences
haiwang@hitic.ia.ac.cn, swzhang@hitic.ia.ac.cn

Abstract. In this paper, we evaluate the effectiveness and efficiency of the global image descriptors and their distance metric functions in the domain of object recognition and near duplicate detection. Recently, the global descriptor GIST has been compared with the bag-of-words local image representation, and has achieved satisfying results. We compare different global descriptors in two famous datasets against mean average precision (MAP) measure. The results show that Fuzzy Color and Texture Histogram (FCTH) is outperforming GIST and several MPEG-7 descriptors by a large margin. We apply different distance metrics to global features so as to see how the similarity measures can affect the retrieval performance. In order to achieve the goal of lower memory cost and shorter retrieval time, we use the Spectral Hashing algorithm to embed the FCTH in the hamming space. Querying an image, from 1.26 million images database, takes 0.16 second on a common notebook computer without losing much searching accuracy.

1 Introduction

There are more and more images in our daily life. It is of great significance to find the one needed among a large number of images. The content based image retrieval (CBIR) may be just a solution to this problem, which is a prosperous researching field. See a recent survey [3] for a deep understanding.

The CBIR retrieval process usually follows a similar pattern. Firstly, an image is represented by features, either a vector of global features like several MPEG-7 image descriptors or a set of local image features like SIFT [7]. After an image is represented by features, a similarity measure is proposed to calculate the similarity between images. Usually, the image representation and the distance measure should be considered simultaneously; Secondly, to tradeoff between effectiveness and efficiency, an indexing scheme has to be proposed to tackle the dilemma of the large scale image database and the requirement of a real-time response time.

Currently, in the field of the near duplicate detection and object recognition, the bag-of-words features based on local image descriptors have gained most of the attention, and have achieved some success, like [11,12,6]. However, the local image features take a long time to extract. When performing the visual key words generation process like the k-means clustering, it will consume a lot of time to deal with large database. At the same time, when the number of visual

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part I, LNCS 6978, pp. 626–635, 2011.

words is very large, for example, millions or even larger, the new comer image to be retrieved will take lots of time to compare with each visual word in order to get the bag-of-words representation. Although some ingenious methods like hierarchy quantization method Vocabulary Tree [9] have been proposed to reduce the bag-of-words quantization time, the quantization error has also increased. Besides, because each image has a set of local descriptors, ranging from hundreds to thousands of dimensions, the storage space for these features is very huge.

Considering the near duplicate images often share most of the same appearances, only some small parts change significantly. One vector of a global representation may suffice to depict the specific image, which indeed has the merit of easy computing and storage efficient. The global descriptors also have the merits of no need to take a long time and use a large dataset to train the bag-of-words model. In spite of these merits, the global features seem to be forgotten in the domain of object recognition and near duplicate detection.

Recently,the authors [5] evaluate the GIST descriptor [10] in the web-scale image search, which has achieved fairly exciting results. This encourages us to evaluate different global features against two famous datasets with ground truth. The results show that The GIST descriptor is indeed a better choice than several global MPEG-7 descriptors, see [1] for an overview, like Color Layout Descriptor, Edge Histogram Descriptor and Scalable Color Descriptor, but it seems that the FCTH [2] a fuzzy color and texture histogram outperforms GIST by a large margin with fewer dimensions of feature. FCTH feature only needs 72 bytes, while the GIST descriptor needs 960 floating numbers. The FCTH descriptor is also much efficient by using a simple similarity measure compared to the GIST descriptor, which using L_2 similarity measure. Considering this in a context of millions of images to be compared, this little promotion of performance will save a lot of computation resources as well as lots of time, which may make the retrieval to be processed in real time.

In this paper, we compare different global image features using the MAP protocol against two famous datasets with ground truth. We evaluate different similarity measures for two effective global features GIST and FCTH. The results show that the FCTH is outperforming GIST and several MPEG-7 descriptors. We propose to use the L_1 similarity measure for both the GIST and FCTH, considering the better performance and lower computational complexity. At the end, we use the state-of-art Spectral Hashing to represent the FCTH feature in the hamming space. We present the results of the scalability of using Spectral Hashing algorithm in large scale image retrieval context.

The rest of the paper is organized as follows. It starts with the image descriptors and similarity measures in Section 2, and then in Section 3 we give a short introduction to the Spectral Hashing algorithm and use it to derive the hamming features for retrieval. In Section 4 we show the datasets and measure to evaluate the performance of the retrieval results. In Section 5 we list the experiments we are performing and give the evaluation results. Conclusions are presented in Section 6.

2 Descriptors and Similarity Measure

2.1 Image Descriptors

In this section, we give a brief description on the image features and distance functions we are going to evaluate.

FCTH feature, which includes color and texture information in one histogram, is very compact and only needs 72 bytes to characterize it. This feature is derived from the combination of 3 fuzzy systems. To compute this feature, the image is initially segmented into blocks. For each block, a 10-bin histogram is generated from the first fuzzy system. The 10-bin histogram is derived from 10 preselected colors in the HSV color space. This histogram is then expanded to 24-bins using the second fuzzy system by including hue-related information for each color. For each image block, a Haar Wavelet transform is applied to the Y component. After a one-level wavelet transform, each block is decomposed into four frequency bands, and the coefficients of the three high frequency bands HL, LH, and HH are used to compute the texture features. The intuition for using these three high frequency bands is that each of them reflects the texture changing directions. After using the third fuzzy system, the histogram is expanded to 192-bins by integrating the extracted texture information and the 24-bins color information. A quantization is applied to limit the final length of the feature descriptor to 72 bytes per image.

GIST feature is based on a low dimensional representation of the scene, bypassing the segmentation and the processing of individual objects or regions. The authors propose a set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) that represent the dominant spatial structure of a scene. The descriptor is gained as follows: the image is segmented by a 4×4 grids, and the orientation histograms are extracted.

MPEG-7 Color Layout Descriptor(CLD) is designed to represent the spatial color distribution of an image in YCbCr color space. This feature is obtained by applying the discrete cosine transform (DCT) in a 2-D image space. It includes five steps to compute this descriptor: (1) partition image into 8×8 blocks; (2) calculate the dominant color for each of the partitioned blocks; (3) compute the DCT transform; (4) nonlinear quantizate the DCT coefficients; (5) zigzag scan of the DCT coefficients.

MPEG-7 Edge Histogram Descriptor (EHD) is describing spatial distribution of four directional edges and one non-directional edge in the image. An image is divided into non-overlapping 4×4 sub-images. Then, from each sub-image an edge histogram is extracted, each sub-image histogram consists of 5 bins with vertical, horizontal, 45-degree diagonal, 135-degree diagonal, and non-directional edge types. Each image is represented by an edge histogram with a total of 80 (4×4×5) bins.

MPEG-7 Scalable Color Descriptor(SCD) is a color histogram in HSV color space encoded by Haar Transform. SCD aims at improving storage efficiency and computation complexity. Usually the number of bins can span from 16 to 256.

2.2 Similarity Measure

In terms of the CLD, EHD, SCD and FCTH, we use the excellent image retrieval LIRe [8] framework to extract these features, and for CLD, EHD and SCD, we use the default similarity measure to measure the similarity between images. For the FCTH and GIST features, from the later experiment results, we can clearly see their better performance, so we compare different similarity function including L_1, L_2, Histogram Intersection(HI), Tanimoto (T) [2] and evaluate the retrieval results.

$$L_1(x,y) = \sum_{i=1}^{d} \|x_i - y_i\| \tag{1}$$

$$L_2(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{2}$$

$$HI(x,y) = 1 - \frac{\sum_{i=1}^{d} \min(x,y)}{\min(\sum_{i=1}^{d} x_i, \sum_{i=1}^{d} y_i)} \tag{3}$$

$$T(x,y) = \frac{x^T y}{x^T x + y^T y - x^T y} \tag{4}$$

3 Image Indexing Scheme

In this section, we present the image indexing scheme used in this paper to solve the problem of retrieval from a large scale image dataset. We use the state-of-art technique Spectral Hashing [13] to map features into hamming space, and apply the hamming distance to compare image similarities. The computing of hamming distance runs fairly fast in that it only needs bits processing. Furthermore features embedded into the hamming space are very distance preserving, which means that the similar data points in the original feature space will also be mapped nearly in the hamming space. The result will be shown later. Next we give a brief introduction to Spectral Hashing.

3.1 Spectral Hashing

In [13] the authors aim at designing a code which has three properties: (1) is to compute easily for a novel input; (2) is that the code should be compact which only take a small number of bits to represent the feature; (3) maps similar items to similar binary code-words. Considering these properties the authors seek to minimize the average Hamming distance between similar points as follows:

$$\text{Minimize}: \qquad \sum_{ij} W_{ij} \|y_i - y_j\|^2 \tag{5}$$

$$\begin{aligned} & & y_i &\in \{-1,1\}^k \\ \text{Subject to}: & & \textstyle\sum_i y_i &= 0 \\ & & \tfrac{1}{n}\textstyle\sum_i y_i y_i^T &= 1 \end{aligned}$$

Where $\{y_i\}_{i=1}^n$ is the n data-points embedded into hamming space with the length of k, and $W_{n\times n}$ is the distance matrix from the original space. There are three constraints, each of which requires the code should be binary. Every bit has probability 0.5 to equal 1, and the bits should be uncorrelated.

The direct solution to the above optimization is non-trivial since even a single bit binary code is a balanced graph partition problem, which is NP hard. The authors relax the constraints, and the relaxed problem can be efficiently solved by using spectral graph analysis. Further, the authors assume that the data-points are sampled from a multidimensional uniform distribution, which means that the probability distribution p(x) is a separable distribution. After this assumption the out of samples problem can be efficiently solved by a closed form solution not using the Nystrom method which computes linearly by the size of the database for a new point.

The final Spectral Hashing algorithm has two input parameters. One is a list containing n data points,and each one is represented by a d-dimensional vector; the other is the number k, using k binary bits to represent the final embedded hamming feature. The algorithm has three main steps: (1) finding the principal components of the data using PCA; (2) for each coordinate of the final k bits, assume the data distribution are uniform and learn analytical eigenfunction by a sinusoidal function; (3) threshold the analytical eigenfunction to obtain binary codes.

4 Datasets and Evaluation Protocol

4.1 Datasets

We have used two famous evaluation datasets with ground truth, the University of Kentucky dataset and the INRIA Holidays dataset. Apart from the two datasets with ground-truth manual annotations, we also use the large scale IMAGENET dataset as distracting images to evaluate the performance of different image descriptors and the indexing scheme in a large scale dataset.

The University of Kentucky Recognition Benchmark Images [11]. This dataset contains 10200 images altogether, with 4 images in a group to depict either the same object or the same scene from different viewpoints. When searching an image, the first four images should be the images in that group.

INRIA Holidays dataset [6], this dataset mainly contains personal holiday photos. The remaining ones are taken on purpose to test the robustness against various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset includes a very large variety of scene types (natural, manmade, water and fire effects, etc.) and images are of high resolution. The dataset contains 500 image groups, each of which represents a distinct scene. The first

image of each group is the query image and the correct retrieval results are the other images in the group.

IMAGENET Large Scale Visual Recognition Challenge 2010 [4]. We use a subset of 1256612 images from the datasets training set of the JPEG format. The number of images for each category ranges from 668 to 3047.

4.2 Evaluation Protocol

To evaluate performance we use Average Precision, computed as the area under the precision-recall curve. Precision is the number of retrieved positive images relative to the total number of images retrieved. Recall is the number of retrieved positive images relative to the total number of positives in the database. We compute an Average Precision score for each of the query image, and then average these scores to obtain a Mean Average Precision (MAP) as a single value to evaluate the results. The bigger the number is, the better the performance is.

5 Experiments

5.1 Evaluate Global Features

At first, we evaluate the different global features listed in the Section 2. For the GIST we scale the image to 128×128 pixels, then use the implementation in [10] to extract the 960 dimensions feature vector. For other features, like FCTH, SCD, EHD, and CLD, we use the wonderful package LIRe [8] to extract these features and use the default similarity measures to calculate the similarity between images. The result is shown in the Figure 1. From the figure we can see that in both the Kentucky and the Holidays datasets, the FCTH is much better than the GIST descriptor by a large margin, which is much surprising since the FCTH only use 72 bytes while GIST has to use 960 floating numbers. The GIST descriptor performs almost the same as the Color Layout Descriptor, while both the Edge Histogram Descriptor and Scalable Color Descriptor show unsatisfactory results.

From this graph we can see that all the features from the Kentucky dataset perform better than the Holidays dataset. This is because the images in the former group share most of their appearances, while the latter change a lot in the same group. From the results of Holidays dataset, the best performance is still lower than 0.5, we admit that this is an intrinsic defect of the global features compared to local descriptors. In the Kentucky dataset the result is much encouraging, with the FCTH feature has achieved a MAP score almost close to 0.7. We attribute this to the merit of that FCTH consider both the color and texture feature simultaneously.

5.2 Evaluate Similarity Measures

In this subsection we will evaluate how the different similarity measures can affect the retrieval performance. We choose the FCTH and GIST descriptors to

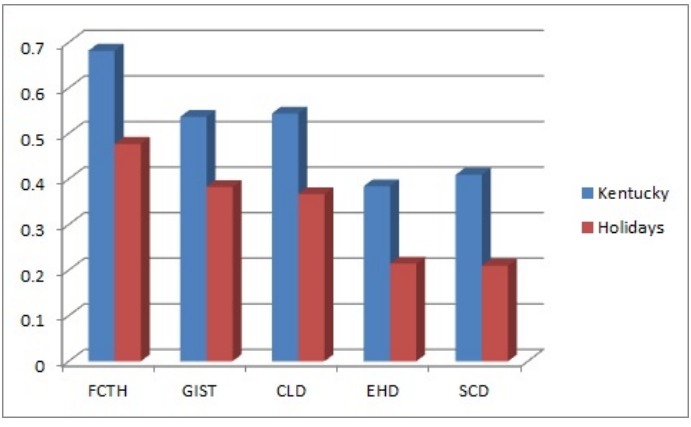

Fig. 1. Evaluate different global descriptors

compare in this round for their better performance in the above experiments. Firstly, we evaluate how the different similarity measures affect the performance of the FCTH feature, and the result is shown in Figure 2, which shows that the Tanimoto, L_1 and L_2 all perform well, achieving almost the same result. In the Kentucky dataset the Tanimoto measure is the best and in the Holidays the L_1 is the best, while in both datasets the Histogram Intersection gives the most unsatisfying results. Then we evaluate how the different similarity measures influence the GIST descriptor, the result of which is also shown in Figure 2. Clearly, the L_1 is the best performer in both datasets, and the L_2 and Tanimoto almost achieve the same score. The Histogram Intersection again performs worst.

The authors in [2] use Tanimoto measure as the similarity measure. Judging from the results it performs well, but it seems that the L_1 measure is much better, not only that they make a draw from the evaluation with the Tanimoto measure, but also it is much computational efficient in the large scale retrieval context, where it requires to compare millions or billions of image features, so a lower complexity will indeed decrease the retrieval time, and promote the user experience. For the GIST feature, no doubt, the L_1 is the best choice, which also contradicts with [10]. The authors use L_2 as similarity measure. From this evaluation the L_1 is indeed better than the L_2 measure because of its performance and its lower complexity.

Now let's compare the best result from the FCTH and the GIST feature similarity measure, the FCTH outperforms GIST in both datasets. So despite of the success of the GIST descriptor in the domain of object recognition and near duplicate detection, it seems much wiser if we can try the FCTH feature to test if they can achieve a better result. Judged from the two famous datasets the FCTH indeed gives a better result.

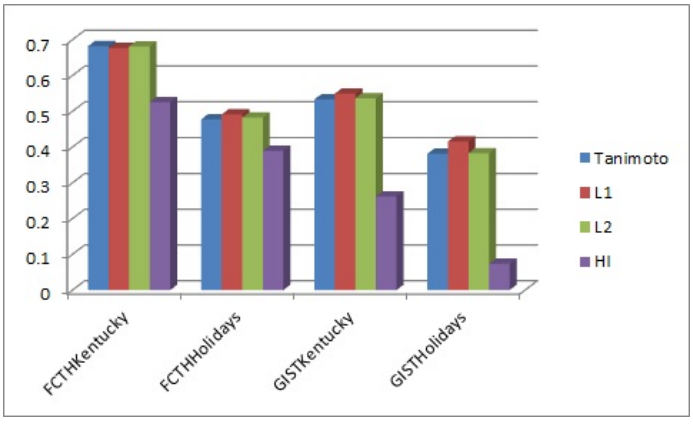

Fig. 2. Evaluate different similarity measures for FCTH and GIST descriptors

5.3 Evaluate Spectral Hashing

From the above experiments we can clearly see that the FCTH descriptor performs better than several MPEG-7 descriptors and even better than the GIST feature. So in this subsection we select the FCTH feature as our final descriptor to evaluate the Spectral Hashing [13] algorithm, to see how the length of embedding bits can affect the results. We use our own implementation of the Spectral Hashing algorithm. The result is shown in Figure 3. Clearly, we can see the trend that the longer hamming bits used as feature, the better performance will achieve in both datasets, which is conform to our intuition. For the different length of the hamming feature from 32 bits to 192 bits, the searching time difference is almost negligible, because this only takes a XOR bits processing, so we use the 192 hamming bits to compare with best performance in the above experiments. First we check the effect on the Kentucky dataset and use the best performance similarity measure. The best MAP score from the four different similarity measures is 0.68, while the 192 bits hamming MAP score is 0.59, decreased by 0.09, but we should also note that the feature is reduced to one third, from the 72 bytes to 24 bytes. In the Holidays dataset the best performance of the L_1 similarity measure is 0.49, decreased to 0.39 with a 192 bits hamming representation. From the above experiments we can conclude that when using the hamming feature derived from the Spectral Hashing, the feature size and retrieval time are reduced significantly, and also can preserve the most of correct results. Later we will mix a large scale of distracting images with each of the two datasets to see how the performance will be.

In this round we will evaluate how a mixture of distracting images will affect the final retrieval performance. We evaluate both datasets. When each of the two datasets is chosen, the IMAGENET dataset with a size of 1256612 images is mixed with the benchmark dataset. We use the 192 bits hamming feature as descriptor. In the Holidays dataset, when mixing with the IMAGENET 1256612

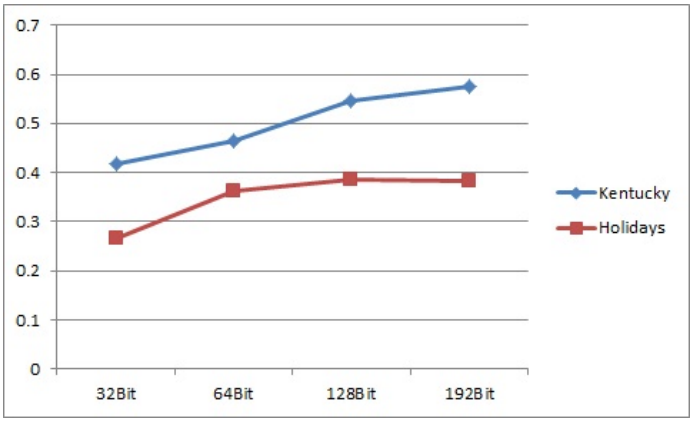

Fig. 3. Evaluate how the length of hamming bits affect results

images, the MAP score is dropping to 0.14 compared to 0.39 without distracting images and also uses the 192 hamming bits as descriptor. In the Kentucky dataset when mixing with the IMAGENET dataset set, the MAP score is from the 0.59 to the 0.39. Although there are some performance dropping, we should also note that we only use the 72 bytes global descriptor to derive the hamming bits features. When using more complicated features, the performance will indeed boost a lot. Also when using these bits features derived from the Spectral Hashing algorithm, the retrieval process can be very efficient, we just exhaustively compare the query image to all the images in the database, and sort the results, without using other indexing methods, the average query response time is 0.16 second from a database of more than 1.26 million images.

6 Conclusions

In this paper, we evaluate the different global features in the domain of object recognition and near duplicate detection against two famous datasets with ground truth. We show the result that FCTH global feature outperforms the state-of-art GIST global feature and several other MPEG-7 global features. This may give the resurgence of the global features when performing some specific image understanding tasks, and may be a complement to the local features to achieve a better result. We also evaluate the different similarity measures to compute the similarity between images, the result of which shows that the L_1 is a better choice for its performance and its low computation complexity for GIST descriptor. To tackle the dilemma of the large scale of image database and the requirement of a real-time response time, we use the Spectral Hashing to embed the feature points to the hamming space, and simply use the hamming distance to efficiently compute similarities between images, which is very efficient because the computation is only the bits processing. This technique is not only efficient

but effective with an average query response time of 0.16 second from a database of more than 1.26 million images with a little performance degradation.

References

1. Chang, S.F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE Transactions on Circuits and Systems for Video Technology 11(6), 688–695 (2001)
2. Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: Fuzzy color and texture histogram a low level feature for accurate image retrieval. In: 9th International Workshop on Image Analysis for Multimedia Interactive Services, pp. 191–196 (2008)
3. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2) (2008)
4. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
5. Douze, M., Jegou, H., Sandhawalia, H., Amsaleg, L., Schmid, C.: Evaluation of gist descriptors for web-scale image search. In: ACM International Conference on Image and Video Retrieval, pp. 140–147 (2009)
6. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
8. Lux, M., Chatzichristofis, S.A.: Lire: Lucene image retrieval - an extensible java cbir library. In: 16th ACM International Conference on Multimedia, pp. 1085–1087 (2008)
9. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2161–2168 (2006)
10. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42(3), 145–175 (2001)
11. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Ninth IEEE International Conference On Computer Vision, vol. 2, pp. 1470–1477 (2003)
12. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: 26th IEEE Conference on Computer Vision and Pattern Recognition (2008)
13. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems (2009)