# Similar Image Retrieval Using Color Histogram in HSV Space and SIFT Descriptor with FLANN

**Yaru Wang, HongXin Li and LongKui Wang**

**Abstract** This work proposes an efficient method for similar image retrieval based on color histogram and local feature descriptors. It has proven that the SIFT descriptor achieves the best performance among all the local descriptors. However, the high dimensionality of the scale-invariant feature transformation (SIFT) feature descriptor brought difficulties for image matching. We first extract color histogram in HSV and discard the image whose histogram is less than the threshold to the query image. Then, for the remaining images, we extract SIFT feature and use Fast Library for Approximate Nearest Neighbors (FLANN) matching algorithm to get a score for every candidate image. Finally, and most important conception proposed by us, we weight both color histogram and SIFT feature to get a score rank as our retrieval result. From the procedure, we have advantages in both time consumption and result accuracy. Experimental results demonstrate that the performance of this scheme is efficient.

**Keywords** SIFT · Color histogram in HSV space · FLANN · Similar image retrieval · Image matching

## 1 Introduction

With the rapid development of network technology and digital technology, there are many similar images photographed from different viewpoints but with the same scene or object. Content-based image retrieval extracts feature information in

Y. Wang (✉) · H. Li · L. Wang
School of Information Science and Engineering,
Lanzhou University, Lanzhou 730000, China
e-mail: yrwang2011@lzu.edu.cn

H. Li
e-mail: hongxinli@lzu.edu.cn

Z. Wen and T. Li (eds.), *Foundations of Intelligent Systems*,
Advances in Intelligent Systems and Computing 277,
DOI: 10.1007/978-3-642-54924-3_103, 

pixels of different images as a clue to calculate the similarity between images. CBIR can work well to find similar images in a dataset of a query image.

Traditional CBIR methods in feature detection and matching revolve around low-level features such as color, shape, and texture. Color histogram depicts the ratio of different colors occupied in the whole image. It takes no account of the spatial position of every color and cannot depict object in the image. However, it can be used in combination with local feature descriptors to get considerable result.

The scale-invariant feature transformation (SIFT) algorithm, proposed in [1] by Lowe in 2004, is found highly distinctive, and invariant to scale, rotation, and illumination changes. According to the evaluation [2], the SIFT descriptor [1] achieves the best performance among all the local descriptors. Because one of the drawback of SIFT is the computational complexity of the algorithm increases rapidly with the number of keypoints, especially at the matching step due to the high dimensionality of the SIFT feature descriptor, many approximate nearest neighbor search algorithm have been proposed, such as kd-tree [3] and vocabulary tree [4]. In [5], Muja and Lowe compared many different algorithms for approximate nearest neighbor search on datasets with a wide range of dimensionality and they found that two algorithms obtained the best performance, depending on the dataset and the desired precision. These algorithms use either the hierarchical *k*-means tree or randomized kd-trees, and we will call the algorithms Fast Library for Approximate Nearest Neighbors (FLANN).

The experimental results combining of color histogram and SIFT show the recall is better than the method with only color histogram or only SIFT with FLANN.

The rest of this paper is organized as follows. Section 2 presents a brief summary of SIFT method and color histogram. Section 3 describes our scheme of retrieval process. Section 4 gives our experiment setup and result. The last section provides conclusions and future work.

## 2 Image Feature

### *2.1 Scale-Invariant Feature Transformation*

SIFT was summarized by Lowe in 2004 [1] with the existing detection method [6]. There are four major steps to generate SIFT features for an image. A brief description of these four steps is provided as follow.

1. Scale-space extrema detection

In order to get the scale-invariant feature points, the algorithm used the difference-of-Gaussian function, and points (see Fig. 1) are defined as minimum or maximum of the result of the function.

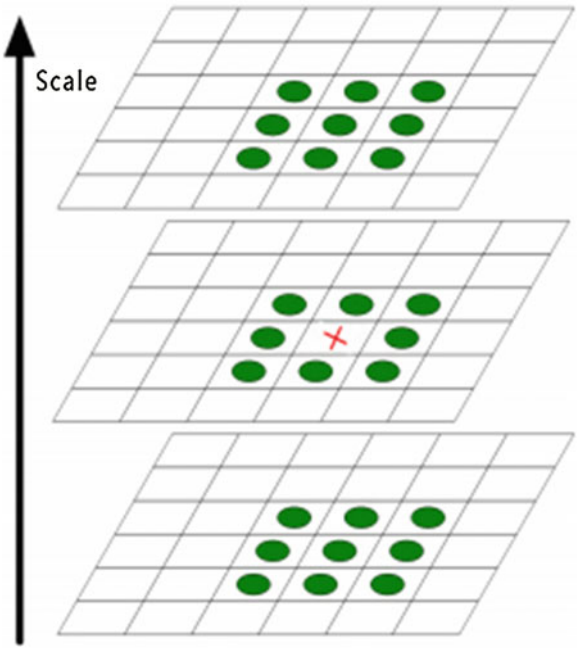


**Fig. 1** Extrema of the difference-of-Gaussian images are detected by comparing a point (marked with *red X*) to its 26 neighbors (marked with *green circles*)

2. Accurate Keypoint Localization

The location and scale of the keypoints can be accurately localized by fitting a 3D quadratic function to the local sample points. At the same time, low-contrast candidate points and edge response points along an edge are discarded in order to improve the stability of the matching and capability of noise immunity.

3. Orientation Assignment of the Keypoints

Each keypoint is assigned a specified orientation by using the gradient distribution of its adjacent pixels. Dominant orientation is selected as the main orientation (see Fig. 2) of the keypoint to ensure the descriptors of image are invariant to rotation.

4. Keypoint Descriptor

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian function. These data are then used to create a set of histograms over a window center on the keypoint. Keypoint descriptor typically uses a set of 16 histograms, aligned in a $4 \times 4$ grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. The results in a feature vector containing 128 elements (see Fig. 3).

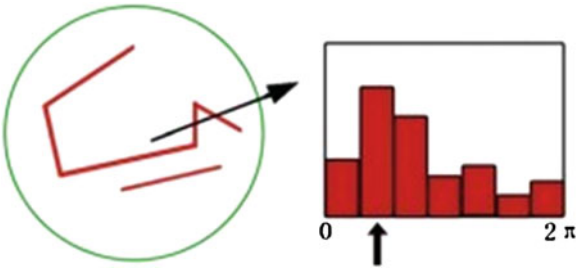


**Fig. 2** The peak in a histogram of local image gradient orientations is selected as the main gradient orientation

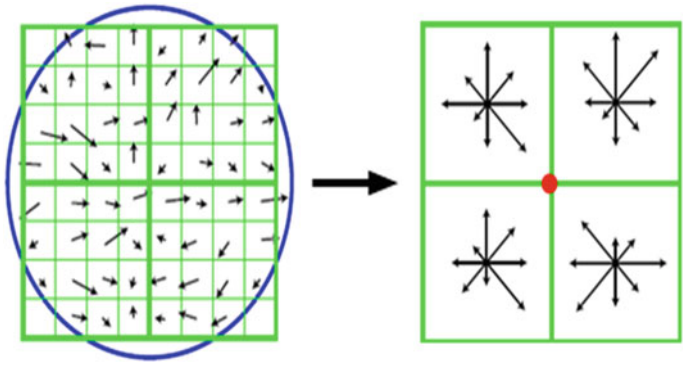

**Fig. 3** The *left* image is the gradient information around the keypoint, and the *right* image is the 128 dimensions of the descriptor

### *2.2 Color Histogram in HSV Space*

Color feature is one of the most widely used traditional low-level features in CBIR systems. Color histogram can be based on different color space and coordinate system. However, RGB color space is not close to human perception. And we use HSV color space.

## 3 Matching Algorithm

### *3.1 FLANN*

FLANN is a library for performing fast approximate nearest neighbor searches in high-dimensional spaces (in large datasets and for high-dimensional features). It contains a collection of algorithms that found to work best for nearest neighbor

**Fig. 4** Dataset image a

**Fig. 5** Dataset image b

search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset.

Suppose m1 is a feature of image I1, through FLANN algorithm, we get the feature m2 in image I2, who has the minimum distance from feature m1, and we take (m1, m2) as a matching point pair. According to the distances of all the matching points, we calculate the minimum distance, and then, we get the threshold $T = \mu \times \min$. If the distance of the matching point pair (m1, m2) is less than $T$, we will take m2 as m1's matching point.

## *3.2 The Definition of Score*

Comparing the color similarity of query image with the image from the dataset, if it less than the preset threshold, the image will be discarded. If the image in the dataset is remaining, we will extract its SIFT local descriptors and use FLANN

algorithm to achieve the matching score which we define it as the number of matching point with the query image for the dataset image. The final score of each image in the dataset is the sum of the matching score which indicates the matched points of SIFT descriptors and 100 multiples of the similarity of color histogram. Then, according to the final score, we will get the rank of images in the dataset for a query image. The definition can be described as an equation.

$$S = \text{score1} + \text{score2} \times 100 \tag{1}$$

The score1 is the matching points of two images, score2 $\subseteq [-1, 1]$ is the color histogram similarity of two images, and $S$ is the final scores.

## 4 Experiment and Results

### *4.1 Experiments Setup*

Dataset: The images we used contain 144 images partitioned into 36 groups, each of which represents a distinct scene, location, or object. They are randomly selected from two datasets, one is the INRIA Holidays dataset [7] and the other is the ZuBuD dataset [8]. We randomly choose one image of each group as the query image, and the correct retrieval results are the other images of the group.

Experiments system: Our experimental program is developed based on Ubuntu 10.04.

Evaluation measure: recall@R. Measuring the recall at a particular rank R, the ratio of relevant images ranked in top R positions, is a very good measure of the filtering capability of an image search system.

### *4.2 Experiments Results*

By comparing the similarity of color histogram, we discard more than half of the images whose similarity is less than the threshold 0.24 in the dataset from each query image. Figures 6 and 7, respectively, show the color histogram of Figs. 4 and 5, which have been converted into RGB space from HSV space for depiction. The threshold also makes sure that all of the relevant images of each query image are remaining. The average number of remaining images for each query image is 40 compared to 108 (the sum of the dataset).

For the remaining images, we extract the SIFT features. The average number of SIFT feature in the dataset is about 1,358. The average time for detecting and extracting SIFT feature is about 0.565 s.

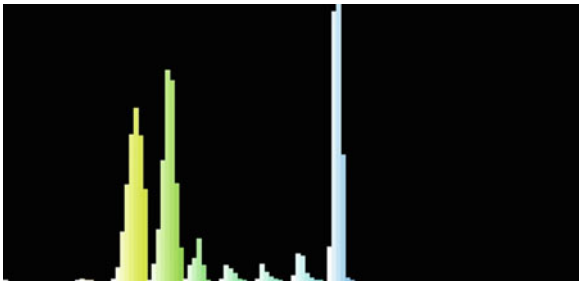

**Fig. 6** The color histogram of Fig. 4

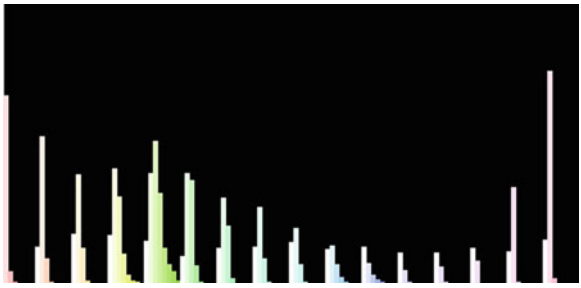

**Fig. 7** The color histogram of Fig. 5

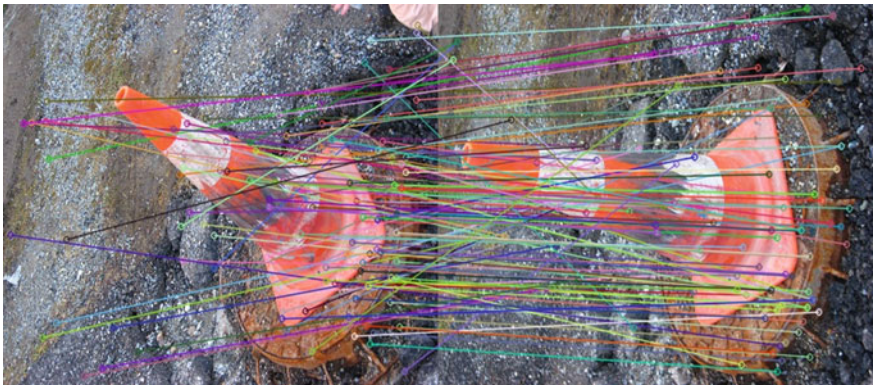

**Fig. 8** One example of matching image

The average time of FLANN algorithm for matching two images for the dataset is about 0.217 s. In the experiment, we set the threshold $T$ ($T = \mu \times \min$) to 300 (for every query image) that decide two features whether matching point pairs or not. Figures 8 and 9 show the example for matching between two different images from the same group. We can see there are some mismatching points, but most are correct matching points.

**Fig. 9** The other example of matching image

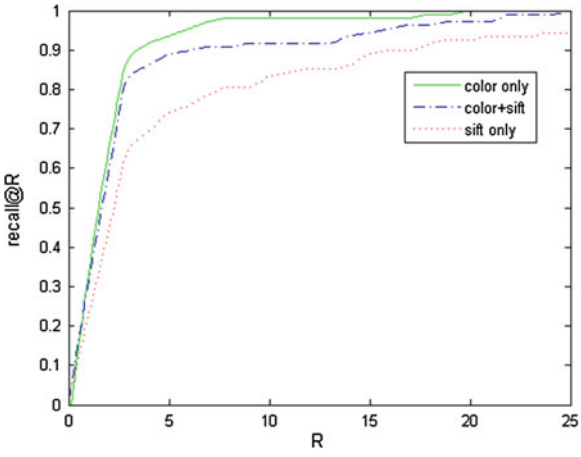


**Fig. 10** Rate of relevant images found in the top R images

According to Eq. 1, we calculate the final score for the dataset images and use rapid sorting algorithm to get the rank of them.

In our experiments, we use recall@R to evaluate the performance of our image retrieval system. Experimental results on recall@R for our method that combine color histogram in HSV space and SIFT descriptor with FLANN and only color histogram in HSV and only SIFT with FLANN are shown in Fig. 10.

From Fig. 10, we can observe that our method is better than the others, and we almost get all the relevant images at the top 7 recall.

## 5 Conclusions and Future Work

This paper proposes a combination of keypoint-based descriptor (SIFT) with FLANN and color-based descriptor. The sole purpose of this work is to provide an efficient similar image retrieval system. The results show that the recall@R has been improved by about 7 and 18 % compared to only color histogram and only SIFT on top 7. For future work, we will concentrate on integrating other global feature into the presented scheme to improve the retrieval efficient.

## References

1. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
2. Mikolajczyk K, Schmid C (2005) A performance evaluation of local descriptors. IEEE Trans Pattern Anal Mach Intell 27:1615–1630
3. Firedman JH, Bentley JL, Finkel RA (1977) An algorithm for finding best matches in logarithmic expected time. ACM Trans Math Softw 3:209–226
4. Nistér D, Stewénius H (2006) Scalable recognition with a vocabulary tree. CVPR
5. Muja M, Lowe DG (2009): Fast approximate nearest neighbors with automatic algorithm configuration. In: International conference on computer vision theory and applications
6. Lowe D (1999) Object recognition from local scale-invariant features. In: Proceedings of seventh international conference on computer vision, pp 1150–1157
7. The dataset is available at http://lear.inrialpes.fr/people/jegou/data.php
8. Shao H, Svoboda T, Gool LV (2003) ZuBuD–Zurich buildings database for image based recognition, technical report 260 computer vision laboratory. Swiss Federal Institute of Technology, Zurich