Geometry and Topology Preserving Hashing for SIFT Feature

Chen Kang, Li Zhu*, Xueming Qian*, *Member IEEE* , Junwei Han, Meng Wang, Yuan Yan Tang, *Fellow IEEE*

***Abstract*—In recent years, content based image retrieval has been concerned because of practical needs on Internet services, especially methods that can improve retrieving speed and accuracy. SIFT feature is a well-designed local feature. It has mature applications in feature matching and retrieval, while the raw SIFT feature is high dimensional, with high storage cost as well as computational cost in feature similarity measurement. Thus, we propose a hashing scheme for fast SIFT feature based image matching and retrieval. First, a training process of hashing function involves geometric and topological information is introduced; second, a geometry enhanced similarity evaluation that considers both the global and details of images in evaluation is explained. Compared with state-of-the-art methods, our method achieves better performances.**

***Index Terms*—CBIR, geometric information, GTPH, hashing, SIFT**

I. Introduction

WITH the explosive increasing of multimedia data on the Internet, searching for the wanted information has become a problem for all users. An attractive domain in this issue is retrieving similar visual data. Efforts have been made to improve the accuracy and speed in image retrieval based on tags and surrounding texts significantly in industry, while Content Based Image Retrieval(CBIR) is limited used, because its accuracy with difficult and changeable background is not so satisfied. One solution to increase the searching speed in CBIR is by using hashing method to embed high-dimensional visual features of an image to lower dimension like Hamming space, as it is more efficient in similarity search [1].

The aim of hashing is to learn binary representations of an existing dataset to make sure that the neighbourhood structure in original space is the same with that in Hamming space. Similarity evaluation in hashing methods means to find the nearest points with the query in Hamming space. For a good hashing method, the nearer in original space two points are, the smaller Hamming distance between binary codes they should have. However, because of the embedding and the limit of binary code length, a loss of original data is usually caused and many discrete points may share the same code and let the distances to the query be the same, which will lead to a decrease of accuracy in image retrieval. Thus, making less loss and distinguish the ranking of results is very important.

In tour recommendation applications, users are usually asked to take a photo of a site and upload it from clients to the server to point out the place and get information [2, 3, 4, 5, 6], or share pictures on the Web [7]. Thus, there are a lot of duplicated images with complicated backgrounds. In this kind of pictures, objects usually have a common part with different view angles and scales, but they have common details. For example, when visitors taking pictures of an ancient gate while visiting, they usually stand at different distances and heights, but in all pictures there are the same gate. During retrieving this kind of images, details should be concerned, otherwise it will be very easy to mess around several similar objects, like tower tops in different ages with similar shapes. The global factor need to be concerned as well, because some architectures may share the same element. At the same time, noise pictures should be less appearing.

Wanting to keep the distance relationship of data, some methods use label information and colour information into feature and estimation like [8, 9, 10], some methods involve topology of features into hashing function like TPH method [11, 12]. When implementing CBIR methods in complex background pictures, we can find in practice that in many cases, a slightly changed architecture photo has a bigger Hamming distance with the original architecture picture than a patterned wrong picture has in our experiment process, since the latter contains a lot of easily-matched feature points.

The aim of this paper is to improve the precision of the hashing method in this situation. SIFT descriptors are widely used in image matching and retrieval [13-26], so in this paper we will focus on these local features. However, the proposed method could be applied to other local features with scale and orientation invariance. SIFT is a local descriptor invariant to image scale, rotation and changes of illumination, while it is

This work was supported in part by the NSFC under Grant 61732008, 61772407, 61332018 and u1531141), the National Key R&D Program of China under Grant 2017YFF0107700, the World-Class Universities (Disciplines) and the Characteristic Development Guidance Funds for the Central Universities, under No.PY3A022.

Chen Kang (kangchen@stu.xjtu.edu.cn) was with SMILES LAB and the School of Software Engineering, Xi’an Jiaotong University, 710049, Xi’an,China, and currently is with Laboratoire des Signaux et Systèmes (L2S), Université Paris-Sud-CNRS-CentraleSupélec, Université Paris-Saclay, 3, rue Joliot Curie, 91192, Gif-sur-Yvette, France.

L. Zhu is with the School of Software, Xi’an Jiaotong University, Xi’an, 710049, China (Correspond author, e-mail: zhuli@mail.xjtu.edu.cn).

X. Qian is with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, Xi’an Jiaotong University, Xi’an 710049, China, also with the SMILES Laboratory, Xi’an Jiaotong University, Xi’an 710049, China, and also with Zhibian Technology Co., Ltd., Taizhou 317000, China (Co-Correspond author, e-mail: qianxm@mail.xjtu.edu.cn)

Junwei Han(junweihan2010@gmail.com) is with the School of Automation and Information Engineering, Northwestern Polytechnical University, 710049, China.

Meng Wang (eric.mengwang@gmail.com) is with the Hefei University of Technology, China.

Yuan Yan Tang is with Macau university, China.

robust to affine distortion to some extent. However, SIFT descriptors are computed with Euclidean distances to measure the similarity, which leads to long computational time and needs big storage. Thus, converting SIFT descriptors into bit streams is introduced to use Hamming distance instead of L_2-distance. An efficient way is using the median number as the classifier, introduced in [20] by Zhou et al. Others also tried to improve the SIFT detector, like PCA-SIFT[27] and SURF[14]. The former one gets better speed and the invariance to changes of illumination, while the latter one improves more, but both of them are weaker than SIFT when images change in scale and orientation or with blur.

In the practical using of SIFT feature, we find out that adding geometric factor of SIFT descriptor can improve the performances of SIFT feature based image matching and retrieval, such as the spatial coding based approaches and their extensions presented in [2, 17, 21, 28, 29]. Thus, motivated by the successful of the geometric information in feature matching, we propose to add geometric and topological information into hashing embedding and similarity measurement to significantly improve the precision. We try to involve it into training process to make the hashed features not only preserve distance relation in Euclidean space, but also preserve the geometric relation to refine the embedded features. At the same time, similarity evaluation method is improved with the consideration of geometry from the aspects of changes of an object's scale and orientation in similar pictures.

The contributions of this paper are summarized as follows: 1) The proposed method has considered geometry and topology relation of SIFT features in hashing function training and feature matching, which keeps more information to improve the accuracy. 2) A geometry enhanced similarity evaluation method is proposed. It considers geometric information in a global aspect during the similarity evaluation to manage a better adaptation and precision, which can help selecting similar images from both view spot and content, as previous methods usually only count for correct matching pairs.

In the following sections, the related work is described in section II; the description of our method is in section III and IV; experiments and results are shown in section V; section VI is the conclusion.

II. Related Work

Many methods have been proposed to involve different factors into LSH (Locality Sensitive Hashing) based methods [30] in CBIR. In this section, we give related methods in part A, and similarity evaluation methods in part B. Part C is about a few normal features.

A. Related Hashing Methods

Content Based Image retrieval (CBIR) has attracted significant attention due to the explosion of the information. One of the most traditional problems is the nearest neighbour (NN) search. Approximate nearest neighbour (ANN) search has taken place the traditional linear NN search method in many situations because of the weakness with computational complexity when there are a lot of data. Later, many measuring methods are proposed to accelerate in finding neighbours, as summarized by Wang et al. in [31], like Euclidean distances, Manhattan distances, Jaccard Coefficient, $\chi 2$ Distance and Hamming distance. Among all of these, Hamming distance is the fastest as well as the easiest to implement since it only needs to learn the number of positions that the corresponding bits are different in two data. With this reason, hashing involved methods are very popular in efficient ANN search [1].

(a) Hashing involved methods

The hashing involved methods can be divided into two categories: the data-independent and the data-dependent.

Locality Sensitive Hashing (LSH) [30] method is one of the most well-known data-independent methods, as hashing functions use simple random projections [32, 33, 34]. In theory, the longer the code length is, the more of the similarity between data preserves. However, traditional LSH-related methods usually need multi-tables to keep a good distinction between data and deal with the collision. Moreover, data structure is ignored in finding hashing functions.

Data-dependent methods like K-means locality sensitive hashing (KLSH) [35] and PCA-hashing (PCAH) [27] are proposed to make hashing functions more specified. Yu et al. use principal component analysis (PCA) to reduce dimensions and preserve the principal component of a given dataset [27]. Later, Gong et al. improved PCAH by adding an orthogonal rotation matrix to refine the projection matrix to improve the accuracy of quantization and proposed it as iterative quantization (ITQ) [36]. This work gives an obvious improvement in large scale image retrieval. An advanced method called topology preserving hashing (TPH) [11, 12] is proposed after ITQ by Zhang et al. In this work, a neighbourhood distance difference matrix used for exploits the neighbourhood rankings and a topo-weighting matrix are presented to preserve the topology of given data in training process based on ITQ. Whereas, it is noticed that in cases of repetitive patterns, wrong matches occur, and in addition to descriptor similarity, geometric characteristics of the feature key points bring useful information in training and retrieval evaluation.

However, these have limited outcome, since the low dimensional descriptor lacks some original information already in quantising. This information is not so important in a few pictures, but it is valuable in large-scale image retrieval. Chu et al. proposed in [37] that many noise images appear in result, and many efforts have been made to distinguish the key features from a large amount of noisy features in a rotation invariant way. Therefore, adding more elements into hashing function training becomes feasible.

In the implementation of data-dependent methods, many efforts have been made to involve geometric information into different methods to work out different situations, as shape based coding is another simple way of carrying image information. For example, [38] proposed by Vyshali et al. shows a normalized scale contour coding to preserve shape description, but it is very limited in complex images. Zhang et al. proposed [39] to make a geometry preserving visual phrases (GVP) to combine bag-of-visual-words method and spatial information. Scalable graph hashing (SGH) [40] proposed by Jiang et al. can approximate the whole graph without explicitly computing the similarity graph matrix and preserve the entire similarity information in the dataset. Many supervised and semi-supervised methods such as [41-45] involve tagged data in training process to capture more data variance and sematic neighbourhood as well.

In addition, deep learning involved hashing function is proposed recently. Many methods are based on the characters of deep neural networks. [22] proposed by Lin et al. adds a latent-attribute layer in deep CNN to make a group of functions similar to hash, then uses traditional compare methods to retrieve in the smaller amount of results. Liong et al. propose deep hashing and supervised deep hashing based on back propagation in [24]. [23] by Xia et al. uses a supervised hashing method to learn a set of hashing functions with the back propagation in deep learning network. Li et al. proposed [25] to give a deep pairwise-supervised hashing (DPSH) for pairwise-labelled situations. Lin et al. gave an unsupervised approach called DeepBit with criterions on binary codes in the loss functions. Liu et al. proposed [47] called Deep Supervised Hashing(DSH) with a pair of input images as well . A supervised semantics-preserving deep hashing (SSDH) is proposed with defining a loss function considering an objective function with classification error, and it need not pairwise or triplet inputs[47]. [48] explored unsupervised deep learning a little while developing supervised learning models. Meanwhile, some deep hashing methods aim at different situations were proposed. For example, [49] used deep hashing neural networks (DHNNs) in large scale remote sensing image retrieval. Nonetheless, because of the characteristics of deep learning, it is necessary to use hand-crafted hashing in some cases. Deep learning methods need advanced hardware to support the computation, and might be too energy-expensive. Their performance might be affected significantly depending on the complexity of image background. Moreover, the transferring ability of a model between datasets is not good, and adjusting the excessive parameters leads to a hard effort.

(b) SIFT Feature Quantization Methods

Some extension methods of Bag-of-Words (BoW) involve with geometric information are worth learning. Yang et al. considered scale and orientation as geometry constraint and involved a geometry difference of two visual word paths into visual synonyms detection in [18]. Yang et al. introduced hierarchical sparse coding in [50]. In [28], Zhao et al. used spatial layers of visual word (SLW) for image location estimation by involving one visual word and its spatial relationships with its neighbour visual words. In [20], Binary SIFT is proposed, as using the median of each data as the threshold to quantise every bit. [21] improves this with Geometry Fan Coding (GFC). It uses BoW to pick out a group of top results first, and then uses a geometric coding algorithm to decide correct matches. At last, with the similarity evaluation, top answers are picked as result.

B. Similarity Evaluation Methods

Since many search results share the same similarity value with the query, how to evaluate each pair of data and pictures becomes important.

To distinguish features, most methods are developing the ways to separate dissimilar data that share the same Hamming distance with the query. For example, weighted hashing method [16] is proposed by Wang et al. to give every bit of a datum different weight in evaluation by a unified framework, so that similar coding data are distinguished.

However, for most situations, the way to measure the similarity of images is calculating the correct matches in each picture with the query. The more, the better. In [13] and [20], two methods about deciding correct matches are proposed, basically to give a threshold to limit acceptable matches. The former one uses a distance ratio as threshold, and the latter uses a distance threshold to decide if they are correct. [21] improves [20] with an involvement of BoW at first and makes a geometry coding in retrieval. In practice, other strategies are added to regular the number of compare features, like saliency detection.

C. Features in Image Matching

There are basically two kinds of visual features: global features and local features. Global features like GIST and colour histogram are usually suitable in small images, or not concerning on the details or easy background; local features like SIFT[13], PCA-SIFT[27] and SURF[14] are better in detail preserving or the situation that salient objects are under partial occlusion.

To get SIFT features, the first step is to use difference-of-Gaussian (DoG) function to identify possible interest points and get their geometry information <*x*, *y*, *scale*, *orientation*>. *x* and *y* represent the position of feature point; scale is the describing scope of DOG scale-space; orientation is the peak value in the point's histogram's direction. Then, use Hessian matrix and histograms to localise key points. Theses interest points are finally described by the 128-dimension SIFT descriptor. PCA-SIFT and SURF are based on SIFT feature. In PCA-SIFT feature extraction, PCA is used to normalise gradient patch instead of histograms in SIFT. SURF detector is based on Hessian matrix, and chooses to create a "stack" and filters the stack with a box filter instead of using pyramids in SIFT.

As performance, while SURF gets least time cost and best robust to changes of illumination in experiment on the same dataset, SIFT finds most matches and is invariant to scale, rotation and blur [15].

III. Geometry and Topology Preserving Hashing

In this paper, we propose a method called geometry and topology preserving hashing (GTPH) for SIFT feature matching. We first give a system overview of our approach. Then, we give the corresponding models in training the GTPH, which is done offline. Later in section IV, we will describe a detailed online similarity measurement with geometric reinforcement.

A. System Overview

The flowchart of the proposed GTPH is shown in Fig.1. It consists of the offline and online systems. First, a geometry and topology preserving hashing function is trained with the given dataset offline, and the dataset's SIFT descriptors are hashed by it. The training process is described in part B, including four relation-preserving matrices. For a query image, first we carry out SIFT feature extraction, and then hash it online by the same hashing function that we trained offline. Last, we use our geometry enhanced similarity evaluating method to measure the similarity of the input query image and a dataset image, as represented in section IV.

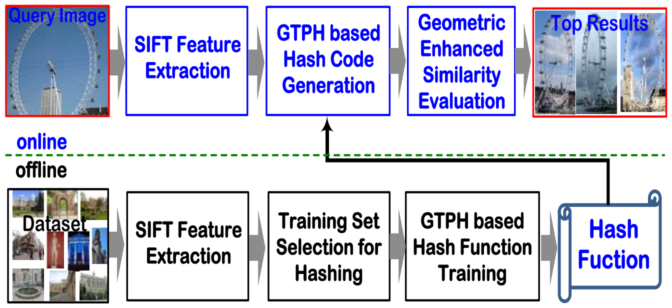

Query Image
SIFT Feature Extraction
GTPH based Hash Code Generation
Geometric Enhanced Similarity Evaluation
Top Results
online
offline
Dataset
SIFT Feature Extraction
Training Set Selection for Hashing
GTPH based Hash Function Training
Hash Fuction

Fig. 1. Flowchart of the proposed geometry and topology preserving hashing based image matching approach.

B. Training Dataset Generation

Hashing-based similarity search methods are popular in image retrieval with big data. In order to acquire effective hashing function, topology has been taken into consideration [11, 12]. Adding more elements into hashing function training can cover the shortage that the 128-dimension SIFT descriptor has lacks of information in quantising. Our GTPH approach for SIFT matching is described as following.

To make the trained GTPH robust to the different applications, we need to generate a large scale SIFT point dataset. For example, a dataset of SIFT feature points selected from 5K images. The total number of selected SIFT feature points is n.

The feature points in a dataset are defined as a dataset D'. We create a zero-centred matrix $\mathbf{X}_{128\times n}$ consisting of all of the n SIFT descriptors. Each column of $\mathbf{X}$ is a 128d SIFT feature descriptor. Then, we randomly choose num samples (i.e. the SIFT points) from D'. For each sample x_i ($x_i \in \mathrm{D}'$), we get its nearest k neighbour points which are measured by the Euclidean distance of 128 dimensional SIFT feature descriptors, and l points that are selected randomly from the points that are far from its nearest neighbours as well. These nearest points and non-neighbour points build up the neighbour set N_i of x_i.

The selected training samples and all their neighbours build up our training set which is defined as $\mathrm{T}' = \{x_i\}_{i=1}^{num} \cup \{N_i\}_{i=1}^{num}, x_i \in \mathrm{D}', \mathrm{N}_i \subset \mathrm{D}'$. The training set T' makes up a training matrix $\mathbf{T}_{128\times t}$. It consists of t SIFT feature points, $t = num \times (1 + k + l)$. The rows and columns of $\mathbf{T}$ are ranked in the same order, which will be emphasised in our later description.

In this paper the l points are used for getting more robustness to train hashing function, and usually l is less than k [11, 12]. In this paper, we set $k = 10$ and $l = 1$.

C. Geometry and Topology Extraction and Regulation

To involve geometry and topology information into hash training, we generate a matrix $\mathbf{M}$ which takes the geometry and topology relationships among training samples at first. Then we build up four matrices: $\mathbf{\Gamma}_r$, $\mathbf{\Gamma}_s$, $\mathbf{G}_r$, and $\mathbf{G}_s$. More detailed explanations are given below. In the following, we will expound these four matrices, and build up the matrix $\mathbf{M}$ in the end. When computing them, the order of columns and rows should be the same with T's correspondingly.

(a) Topology Preserving Matrices

We define a similarity rank weighting matrix $\mathbf{\Gamma}_r$ and Simi-weighting matrix $\mathbf{\Gamma}_s$, as that utilized in [11].

First, we build $t \times t$ matrix $\mathbf{\Gamma}_r$. x_i is a training sample from the selected training sample set, $x_i \in \mathrm{D}'$, and $i \le num$. We define $d(x_i, x_j)$ as the Euclidean distance between two SIFT point x_i and x_j. For every sample x_i expressed as a 128×1 SIFT descriptor matrix in the training sample set, the average Euclidean distance to x_i is defined as $\overline{d_i}$, calculated as in (1).

$$\overline{d_i} = \textstyle\sum_{x_j \in \mathrm{N}_i} d(x_i, x_j)/|N_i|, 1 \le j \le |N_i| \tag{1}$$

where $|N_i|$ is the member number of N_i.

For each corresponding neighbour point x_j, Euclidean distance deviation $\mathbf{\Gamma}_r(i,j)$ can be calculated as follows:

$$\mathbf{\Gamma}_r(i,j) = \begin{cases} \overline{d_i} - d(x_i, x_j) & x_j \in N_i \\ 0 & x_j \notin N_i \end{cases} \tag{2}$$

The nearer of x_j to x_i, the larger value $\mathbf{\Gamma}_r(i,j)$ will get. Notice that in the i_{th} row, only positions of x_j (that belong to neighbour set N_i) have non-zero values, which makes the matrix very sparse.

Setting

$$\mathrm{Norm1}(\mathbf{X}(i,j)) = 2 \times \frac{\mathbf{X}(i,j) - min_i}{max_i - min_i} - 1$$

$$\mathrm{Norm2}(\mathbf{\Gamma}_s(i,j)) = 2 \times \mathrm{e}^{\{-d^2(x_i,x_j)/\sigma^2\}} - 1$$

where min_i and max_i are the minimum and maximum values of Euclidean distance deviation $\mathbf{\Gamma}_r(i,j)$ in row *i*.

Then, $\mathbf{\Gamma}_r(i,j)$ can be normalized as follows:

$$\mathbf{\Gamma}_r(i,j) = \mathrm{Norm1}(\mathbf{\Gamma}_r(i,j)) \tag{3}$$

where min_i and max_i are the minimum and maximum values of Euclidean distance deviation $\mathbf{\Gamma}_r(i,j)$ in row *i*.

$$\begin{cases} min_i = \min\limits_j \mathbf{\Gamma}_\mathbf{r}(i,j) \\ max_i = \max\limits_j \mathbf{\Gamma}_\mathbf{r}(i,j) \end{cases} \tag{4}$$

Thus, $\mathbf{\Gamma}_r(i,j)$ can represent the relative ranking of each sample point's neighbours' distances.

Second, we build a $t \times t$ matrix $\mathbf{\Gamma}_s$. It is the Euclidean neighbourhood relationship preserving matrix as that utilized in [12] as follows:

$$\mathbf{\Gamma}_s(i,j) = \mathrm{Norm2}(\mathbf{\Gamma}_s(i,j)) \tag{5}$$

where σ is the average Euclidean distances of the 128-dimension SIFT descriptors of the dataset. Mind that $\overline{d_i}$ is different with σ. σ is the average distance with each other; d_i is the average distance of the other SIFT points in N_i to x_i. A bigger absolute value of $\mathbf{\Gamma}_s\,(i,j)$ means a bigger similarity between x_i and x_j, as mentioned in [11, 12].

(b) Geometry Preserving Matrices

Inspired by these two matrices $\mathbf{\Gamma}_r$ and $\mathbf{\Gamma}_s$ in topology preserved hashing, to involve geometric information in our hashing function training, we address two $t \times t$ matrices $\mathbf{G}_r$ and $\mathbf{G}_s$ that can preserve the geometry ranking and similarity into hash training.

First, we shall have $\mathbf{G}_r$ to keep the geometric ranking. The normalised distance between these two scales is $ds(s_i, s_j)$ and that between two orientations is $do(o_i, o_j)$. That is,

$$\begin{cases} ds(s_i, s_j) = |s_i - s_j|/\max(s_i, s_j) \\ do(o_i, o_j) = |o_i - o_j|/\max(o_i, o_j) \end{cases} \tag{6}$$

We compute the average scale difference $\overline{s_i}$ for point *i* in the training set as:

$$\overline{s_i} = \textstyle\sum_{x_j \in \mathrm{N}_i} ds(s_i, s_j)/|\mathrm{N}_i| \tag{7}$$

Next, scale deviation $\mathbf{S}_r(i,j)$ is defined as:

$$\mathbf{S}_r(i,j) = \begin{cases} \overline{s_i} - ds(s_i, s_j) & x_j \in \mathrm{N}_i \\ 0 & x_j \notin \mathrm{N}_i \end{cases} \tag{8}$$

The smaller $ds(s_i, s_j)$, the more similar s_j is to s_i, which implies a smaller absolute value of $\mathbf{S}_r(i,j)$. As a result, the contribution of the corresponding x_j is going to be weighted more during training. Similarly to [11, 12], we normalized this score to the [0,1] interval as

$$\mathbf{S}_r(i,j) = 2 \times \frac{\mathbf{S}_r(i,j) - smin_i}{smax_i - smin_i} - 1,$$

where $smin_i = \min_j \mathbf{S}_r(i,j)$ and $smax_i = \max_j \mathbf{S}_r(i,j)$.

Similarly, we repeat the same procedure for orientation deviation by function (9) and (10):

$$\overline{o_i} = \textstyle\sum_{x_j \in \mathrm{N}_i} do(o_i, o_j)/|\mathrm{N}_i| \tag{9}$$

$$\mathbf{O}_r(i,j) = \begin{cases} \overline{o_i} - do(o_i, o_j) & x_j \in \mathrm{N}_i \\ 0 & x_j \notin \mathrm{N}_i \end{cases} \tag{10}$$

and we normalize $\mathbf{O}_r(i,j)$ to take values on [0,1] as follows

$$\mathbf{O}_r(i,j) = \mathrm{Norm1}(\mathbf{O}_r(i,j)) \tag{11}$$

where $omin_i = \min_j \mathbf{O}_r(i,j)$ and $omax_i = \max_j \mathbf{O}_r(i,j)$.

Then we merge the two matrices into the geometry rank weighting matrix:

$$\mathbf{G}_{r=}\frac{1}{2}(\mathbf{S}_r + \mathbf{O}_r) \tag{12}$$

Furthermore, we use the deviations to get similarity relation function $\boldsymbol{G}_s$ as follows:

$$\mathbf{G}_s\,(i,j) = \exp\left\{-\frac{ds^2(x_i,x_j)}{\sigma_s^2}\right\} + \exp\left\{-\frac{do^2(x_i,x_j)}{\sigma_o^2}\right\} - 1 \tag{13}$$

where σ_s is the average Euclidean distance of the scales of SIFT points in D'. σ_o is the average Euclidean distance of points' orientations. A larger absolute value of $\mathbf{G}_s\,(i,j)$ means x_j is more different with x_i.

(c) Geometry and Topology Preserving Matrix

The above four matrices can enhance both the geometry and the topology relationship between training points. The last process in this part is to integrate the four matrices, as follows:

$$\mathbf{M} = \beta[(1-\gamma)\mathbf{\Gamma}_r + \gamma\mathbf{\Gamma}_s] + (1-\beta)[(1-\gamma)\mathbf{G}_r + \gamma\mathbf{G}_s] \tag{14}$$

where γ is the weight to balance the matrix $\mathbf{\Gamma}_r$ and $\mathbf{G}_r$ with neighbourhood relationship preserving matrix $\mathbf{\Gamma}_s$ and $\mathbf{G}_s$, while β weights the importance of the 128-dimension feature's topology information to the geometry information.$\beta, \gamma \in (0,1)$.

To make it clearer, an overflow of the Geometry and Topology Extraction and Regulation is given as Algorithm.1.

ALGORITHM 1. OVERFLOW OF THE GEOMETRY AND TOPOLOGY EXTRACTION AND REGULATION

```
Input: matrix $\mathbf{X}_{128\times n}$, training matrix $\mathbf{T}_{128\times t}$
Output: matrix $\mathbf{M}$
-Begin
  //Calculate Topology Preserving Matrices
  For every sample $x_i$ do
    For each corresponding neighbour point $x_j$ do
      calculate Euclidean distance deviation $d(x_i,x_j)$, $ds(s_i,s_j)$, $do(o_i,o_j)$
    calculate $\overline{d_i}$, $\overline{o_i}$, $\overline{s_i}$
  calculate $\mathbf{\Gamma}_r(i,j)$, $\mathbf{S}_r(i,j)$ , $\mathbf{O}_r(i,j)$
  normalize $\mathbf{\Gamma}_r(i,j)$, $\mathbf{S}_r(i,j)$ , $\mathbf{O}_r(i,j)$
  calculate $\mathbf{G}_r$
  calculate Euclidean neighbourhood relationship preserving matrix $\mathbf{\Gamma}_s\,(i,j)$, $\mathbf{G}_s$
  calculate $\mathbf{M}$
-End
```

D. Hashing function Training

A hashing function is to map descriptors into a lower dimension while making the most similar ones in original space still has a large possibility being similar in the lower dimension. Thus, the training consists of two steps: First, we use PCA to get the principal component of geometry and topology emphasised dataset. Second, we use singular value decomposition (SVD) to minimise the loss of quantisation. We describe them as follow.

(a) Principal Component Extraction

Follow PCA [36] method to solve an Eigen-decomposition of matrix **A** given by (15) (16).

$$\mathbf{G} = \mathbf{T} \times (\mathbf{M} + \mathbf{M}^{\mathrm{T}}) \times \mathbf{T}^{\mathrm{T}} + \mathbf{X} \times \mathbf{X}^{\mathrm{T}} \tag{15}$$

$$\mathbf{A} = \mathbf{G} + \mathbf{G}^{\mathrm{T}} \tag{16}$$

where $\times$ represents matrix multiplication. $\mathbf{X}$ is the matrix that is built up with all 128-dimensional descriptors of D' in the dataset.

$\mathbf{G}$ is a 128×128 matrix that can carry both of neighbourhood ranking and relationship information. From the first term $\mathbf{T} \times (\mathbf{M} + \mathbf{M}^{\mathrm{T}}) \times \mathbf{T}^{\mathrm{T}}$ we can extract the principal components of descriptors that preserve geometry and topology relationship of the training set. The second term $\mathbf{X} \times \mathbf{X}^{\mathrm{T}}$ is for preventing overfitting. $\mathbf{G}$ is not a symmetric matrix, thus we should process it as function (16).

The m largest eigenvalues of the decomposition construct a projection matrix $\mathbf{W}_{128\times m}$, where m decides the bit number of hashing result.

(b) Minimise the Loss of Quantisation

For the hashing process, we want to find an optimal mapping $\mathbf{R}_{m\times m}$ that can transfer the original SIFT feature matrix $\mathbf{A}_{128\times n}$ to a short binary matrix $\mathbf{Y}_{m\times n}$. We have

$$\mathbf{Y}_{m\times n} = \mathrm{sgn}(\mathbf{R} \times \mathbf{Z}) \quad (17)$$

where $\mathrm{sgn}(\cdot)$ is the symbolic function and $\mathbf{Z} = \mathbf{W}^{\mathbf{T}} \times \mathbf{A}$. $\mathbf{Y}_{m\times n}$ represents the hashed $\mathbf{A}$. Our target is to find an orthogonal matrix $\mathbf{R}_{m\times m}$ that can minimise the quantisation loss $\mathbf{Q}(\mathbf{Y},\mathbf{R})$ as follows:

$$\mathbf{Q}(\mathbf{Y},\mathbf{R}) = \|\mathbf{Y} - \mathbf{R} \times \mathbf{Z}\|_{\mathbf{F}}^{2} \quad (18)$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, $\mathbf{R}$ is initialized as a random square matrix with an $m \times m$ size.

Then, we utilize an iterating procedure to get the optimized $\mathbf{R}_{m\times m}$ as follows: 1) use the $\mathbf{R}$ to update $\mathbf{Y} = \mathrm{sgn}(\mathbf{R} \times \mathbf{Z})$. 2) use a SVD decomposition to solve $\mathbf{Z} \times \mathbf{Y}^{\mathrm{T}}$, lead to matrices $\mathbf{U}, \mathbf{\Sigma}$ and $\mathbf{V}$ as [12]. 3) update $\mathbf{R} = \mathbf{V} \times \mathbf{U}^{\mathrm{T}}$. 4) if $\mathbf{Q}$ not convergent then continue the steps 1)~3). Otherwise, when $\mathbf{Q}$ satisfies the convergence condition, then stop the iteration and we denote $\mathbf{P} = \mathbf{W} \times \mathbf{R}^{\mathrm{T}}$.

(c) Hash Feature Generation

Assume that the SIFT feature matrix of an image is denoted as **Input,** then its hashing representation is as

$$f(\mathbf{Input}) = \mathrm{sgn}(\mathbf{P}^{\mathrm{T}} \times \mathbf{Input}) \quad (19)$$

where $\mathrm{sgn}(\cdot)$ is the symbolic function and $\mathbf{P} = \mathbf{W} \times \mathbf{R}^{\mathbf{T}}$. $f(\mathbf{Input})$ means the corresponding output binary descriptors in columns as a matrix.

IV. Geometry Enhanced Similarity Evaluation

In this section, the steps of our similarity evaluation is given. For the SIFT feature matching based on our hashing method, we involve geometry information into the similarity evaluation step. We shall have Hamming Distance Score, Scale Score and Orientation Score to get a final score for each image. We are going to describe the online process step by step in the following.

A. SIFT Feature Hashing

SIFT features of the dataset images should be extracted offline. For a query image, first, we shall extract its SIFT features and get geometry information. Second, with the same hashing function, we input the query's features as matrix **Input**, and then get binary descriptors of query's feature points.

B. SIFT Feature Matching

Let us define the SIFT feature points in the query image as a set $Q = \{q_1, q_2, \ldots, q_n\}$, and those of the i_{th} image in the retrieving dataset as $Pi = \{p_1, p_2, \ldots, p_z\}$, i={1,2, ..., I}, i is the total number of dataset images and z is the number of feature points in image Pi. $\mathrm{H}(q,p)$ denotes the hamming distance between two feature points q and p.

A feature point q_k in query Q is deemed to be a reliable match to a point p_j in the retrieving dataset according to the following ration criterion. In image Pi, p_j and p_l are the closest and the second-closest feature points to q_k in Hamming space, which leads to $\mathrm{H}(q_k,p_j) < \mathrm{H}(q_k,p_l)$. The ratio $\frac{\mathbf{H}(q_k,p_l)}{\mathbf{H}(q_k,p_j)}$ decides whether the match is reliable. The ratio criterion was originally proposed in [13], where also the impact of r on matching has been studied. When $\frac{\mathbf{H}(q_k,p_l)}{\mathbf{H}(q_k,p_j)} \geq r$, q_k and p_j are defined as a reliable match. We chose $r = 1.05$ in our experiments.

C. Geometric Enhanced Similarity Measurement

Here we calculate the matching score of an input query image with dataset images by a geometric enhanced similarity measurement approach. In our similarity measurement approach, the average hamming distance and geometric score are fused in the similarity measurement.

(a) Hamming Distance Score

Hamming distance score (HDS) of Pi is defined as

$$HDS_i = \frac{\sum_{k=1}^{\omega} \min \mathbf{H}(q_k,p_j)}{\omega}, k \in [1,\omega], j \in [1,z] \quad (20)$$

where z is the number of feature points in query, and ω is the number of reliable-matched feature points number. Obviously, $\omega \leq n$. Every p_j and q_k are reliable matches.

(b) Geometric Score

According to the description of SIFT feature, similar feature points should have similar 128×1 descriptors regardless of position, orientation, scale and luminance. Whether being hashed or not, considering about feature points of similar objects or salient spots, they should have steady scale ratios and rotation degrees with reliable matched pairs. A matched pair of feature points' scale ratio should be 1, as well as orientation variance should be 0, if they are exactly the same in two pictures. If one point's surrounding area is enlarged equably, its scale and its neighbour points' scales should get larger with a same ratio according to [13]. If the point and its surrounding area are rotated with a same degree, their orientations should have similar difference values comparing to the original image.

If the ratios or difference values are not the same, there can be two possible situations. One is that the view changes with the same spot, such as the front view and an oblique view of a building-they have same architectural elements and similar details with shearing, but are not geometrically the same. Another possible situation is that the 'reliable matched pairs' are still not the correct ones, even though they are more reliable to be correct in possibility. The latter situation will absolutely decrease the accuracy of retrieval and matching.

Therefore, if an image's feature points have more steady geometric information with matched points in query, we can safely assume they are more similar to the query from both content and view spot than images with wavy ones.

Thus, we propose Geometric Score to help improving similarity evaluation. It consists of Scale Score and Orientation Score, representing the geometric difference in general between pictures and the query.

A scale score (SS) between image P_i and query Q is defined based on the reliable matches as function (21).

$$\mathrm{SS}_i = \sum_{k=1}^{\omega} v\left(\left|\frac{s(q_k)}{s(p_j)} - 1\right|\right), k \in [1,\omega], j \in [1,z] \quad (21)$$

where $s(q_k)$ and $s(p_j)$ are the scales of p_j and q_k. $v(x)$ is the variance of x, which is determined as follows:

$$\mathrm{v}(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2. \quad (22)$$

where x_i is the samples in and $\bar{x}$ is the average value of x. The variance of the whole picture is calculated to evaluate the scale ratios' fluctuation.

Orientation score (OS) between image P_i and query Q is defined as function (23).

$$\mathrm{OS}_i = \sum_{k=1}^{\omega} v\left(\left|o(q_k) - o(p_j)\right|\right), k \in [1,\omega], j \in [1,z] \quad (23)$$

where $o(q_k)$ and $o(p_j)$ are the orientations of p_j and q_k.

(c) Geometry Enhanced Similarity Evaluation

After matching with images, because of the various score scales resulted by different queries, all Hamming Distance Scores, Scale Scores and Orientation Scores need to be resized as function (24) (25).

$$\mathrm{SS}_i' = 2 \times \frac{\mathrm{SS}_i - \min(\mathrm{SS})}{\max(\mathrm{SS}) - \min(\mathrm{SS})} \quad (24)$$

$$\mathrm{OS}_i' = 2 \times \frac{\mathrm{OS}_i - \min(\mathrm{OS})}{\max(\mathrm{OS}) - \min(\mathrm{OS})} \quad (25)$$

An integrated score presented as SCORE_i is given to image P_i in the dataset as function (26).

$$\mathrm{SCORE}_i = \mathrm{HDS}_i + b \times \mathrm{SS}_i' + c \times \mathrm{OS}_i', b, c \geq 0 \quad (26)$$

where parameters b and c are utilized to balance the weights of HDS, SS and OS. In the case that b=0 and c=0, it is the commonly utilized hamming distance based similarity measurement, while $b \neq 0$ or $c \neq 0$ corresponds to our proposed geometry enhanced hamming distance based similarity evaluation approach. Their impacts to the final matching based image retrieval are discussed in section V.

Finally, in our matching based image retrieval approach, we rank the dataset images according to their SCORE_i in the dataset. The smaller value SCORE_i is, the image is more similar to the query. Adding scale score and orientation score can balance the general performance of a query, because it gets off some noise images as their geometric information is different. The experiment results are presented in section VII.

V. Experimentation and Results

To express the accuracy of our method, we implement experiments on Oxford Building benchmark dataset [17] and Geo-Tagged Large Scale Dataset (GOLD) mentioned in [18].

Precision comparison are made among SIFT matching [13], Binary SIFT [20] (in short bSIFT), GFC[21], SGH[40], ITQ[36], TPH[11, 12] and our method. We use the mean precision as the criteria to compare method [26]. The ground truth is labelled manually.

A. Dataset Processing

Oxford Building benchmark dataset consists of 5062 photos of Oxford landmarks images provided by "Flickr", grouped by 11 landmarks. GOLD dataset is more complex and noisy than Oxford Building dataset. We use it to show the performance and discuss a few parameters in hash training.

(a) Oxford Building benchmark dataset

In our experiments, we randomly selected 29 queries in 3 manually selected groups based on the 11 landmarks in Oxford Building benchmark dataset and manually choose 11 ground truth sets.

In training process, 1000 feature points are randomly selected from all feature points in Oxford building benchmark dataset as the training set. γ is 0.2 and α is 1. Each training point's 10 nearest neighbour feature points are chosen in Euclidean space, and 1 non-neighbour random feature point is added to increase adaptability, leading to k=10 and l=1.

For each landmark, we try to separate them into 3 image sets- "detail", "normal", and "general". "Normal" set contains images with an obvious spot and an obvious background, like what normal camera man get when we take photo of a cathedral or a gate; "detail" images have a part of the spot, like a window or half of a gate; "general" is the pictures far from the spot like bird's-eye view, or with irrelevant background like a lot of trees and extra buildings. According to practical experience, "detail" and "normal" pictures should be better as queries, but it is possible that users want to search for "general" pictures. We also use this grouping way to make sure the queries are fair to some extent.

(b) GOLD Dataset

GOLD contains more than 227 thousand images covers 80 different places from Flickr, Baidu and Google. It is designed as a test set for GPS location estimation and place of interest recognition, and images are further processed to 200×200 pixel size in [19]. We use the processed latter one as the dataset. There are not only images about the architectures themselves, but also a lot of travelling photos, including those with tourists, selfies and scenes. It is much more complex than the Oxford Building benchmark dataset.

In this experiment, we use the trained hashing function in Oxford Building dataset. 20 queries are randomly chosen from a few sites picked by hand. We mainly choose images contain obvious sites without more than 1 person, including Angkor Wat, Great Hall of the People, Colosseum, Easter Island statue, Arc de Triomphe, etc. Due to the complexity of GOLD, we choose correct images by hand.

Parameter b and c are set as 0, which means that the geometry enhanced similarity evaluation part is not added, to prove the role of geometric factor in training process.

B. Evaluation Criterion

We use the mean precision [26] at top K of queries as the criteria to compare performance which is defined as:

$$P@\mathrm{K} = \frac{1}{T} \times \sum_{i=1}^{T} \left(R_i / K \right) \quad (27)$$

where R is the number of correct images in top K images of the evaluation result. In our experiment, we use $K = 2, 5, 10, 15, 20, 25, 30$.

C. Compared Methods

Original SIFT matching, GFC, SGH, ITQ and TPH methods are used as comparisons. We have parameter discussion as well in section F.

Original SIFT descriptors are used to see the loss of hashing. SIFT uses the average Euclidean distance of matched pairs instead of Hamming distance to measure the similarity.

Binary SIFT uses the median as the hashing rule, and the matches' Hamming distances under a threshold are considered as correct matches. GFC adds a coding involves with position, scale and orientation of SIFT features. First, we need to use BoW to get top 500 pictures. Then we use GFC on these images. With spatial verification, acceptable matches can be picked out. α is 5 and τ is 2, as [11] recommends.

SGH involves a whole graph matrix; ITQ improves the way to get hashing function with PCA methods; TPH involves topology consideration. For SGH and ITQ, we uses the recommended parameters in [40] and [36]. TPH is in the same parameter values with our methods. All training feature points are chosen in the same way with ours if necessary.

D. Objective Performance Comparison

In order to show the effectiveness of different hashing approach, we systematically evaluate the SIFT, binary SIFT (bSIFT), GFC, SGH, ITQ, TPH and GTPH on Oxford Building benchmark dataset. TABLE 1 is the comparison of methods with normal similarity evaluation in 128 bit, i.e. we set *m*=128, which means we want the embedded descriptors to be 128 bits long. It can be seen from TABLE 1 that our trained hashing function can make an obvious promotion in top 30 precision, comparing to the state-of-art methods. It can be seen in TABLE 1 that our method goes to 0.845 for precision at top 2, but other methods reach 0.689 at most. GTPH method keeps both geometry and topology information in hashing function training. From TABLE 1, we find that the top 2 performance of our approach is very close to the raw SIFT matching.

TABLE 1. OXFORD BUILDING BENCHMARK DATASET, 128 BITS, TOP 30'S MEAN PRECISION COMPARISON BETWEEN SIFT, BINARY SIFT, SGH, ITQ, TPH AND GTPH.

Top	SIFT	bSIFT	GFC	SGH	ITQ	TPH	GTPH
2	0.862	0.340	0.623	0.672	0.689	0.672	**0.845**
5	0.724	0.140	0.435	0.421	0.441	0.386	**0.635**
10	0.593	0.100	0.291	0.303	0.352	0.272	**0.500**
15	0.538	0.140	0.232	0.232	0.290	0.230	**0.432**
20	0.488	0.120	0.175	0.195	0.241	0.186	**0.369**
25	0.432	0.110	0.159	0.23	0.200	0.23	**0.315**
30	0.383	0.130	0.150	0.147	0.183	0.145	**0.290**

TABLE 2. GOLD DATASET, 128 BITS, TOP 30'S MEAN PRECISION COMPARISON BETWEEN SGH, ITQ, TPH AND GTPH.

Top	SGH	ITQ	TPH	GTPH
2	0.900	0.900	0.867	**0.933**
5	0.733	0.760	0.707	**0.787**
10	0.627	0.673	0.600	**0.667**
15	0.547	0.587	0.529	**0.591**
20	0.493	0.533	0.477	**0.540**
25	0.451	0.491	0.429	**0.499**
30	0.420	0.458	0.407	**0.458**

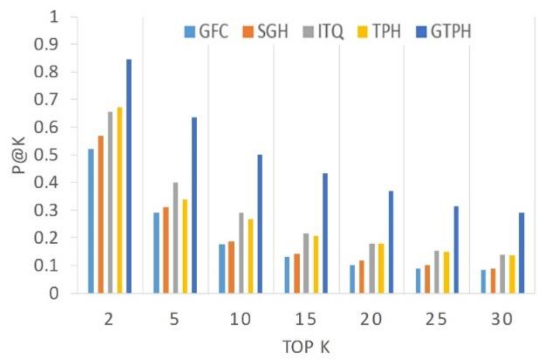


Fig.2. Top 30's mean precision comparison between SGH, ITQ, TPH and our GTPH method in Oxford Building dataset. The approaches are all with 32 bits and in our GTPH we set *b*=0.3 *c*=0.5

Due to GOLD dataset is far larger than Oxford Building dataset, it is computational intensive for SIFT feature matching. We provide the comparison for SGH, ITQ, TPH, and GTPH in TABLE 2, since they are all data-dependent hash training methods. From it, we can see that GTPH performs better than SGH, ITQ and TPH. It reaches 0.933 at Top 2, even the dataset is more complicated than Oxford Building benchmark dataset, while ITQ is 0.900, TPH is 0.867 and SGH is 0.900.

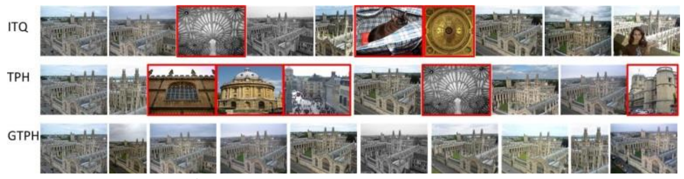
Fig. 2 shows the mean precision comparison in 32 bits (i.e. m=32) in Oxford Building dataset for the GFC, SGH, ITQ, TPH and GTPH based hashing approaches. From the comparisons, we find that GTPH outperforms the other approaches. When looking at top 30 results in pictures, we can see that some noise pictures are wiped out or pushed backward by our GTPH, and correct results in top 10 rank better than ITQ and TPH.

E. Subjective Performance Comparison

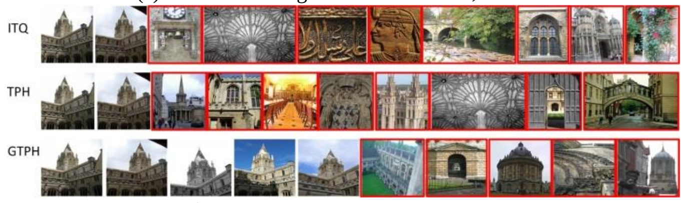
To show the hash feature matching based performances of ITQ, TPH and GTPH under different bits, we give three examples and provide their corresponding retrieval results. Two typical queries with four groups of results in Oxford Building benchmark dataset are shown in Fig.3, as well as a query in GOLD dataset. The first image of each line is both the query and the first result, showing the correctness of each experiment implementation. The top line of each group is the ITQ's result, the middle is TPH's, and the lowest one is our method's. Fig.3(a) and (b) are the results using 128-bit hash representation; Fig. 3(c) and (d) are the results of 32-bit. Fig.3 (e) is an image of Angkor Wat, with 32-bit, and Fig.3(f) is with 128-bit. In Fig.3, the incorrect results are marked in red.

By comparing Fig.3(a) and (c), Fig.3(b) and (d), and Fig.3(e) and (f), we find that by assigning the hash code more bits, better matching performances are achieved. For the same input image, our GTPH based image matching gets fundamental improvements over ITP and TPH.

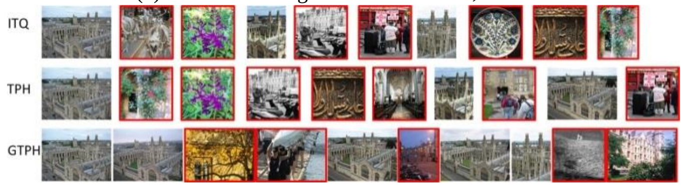
In Fig.3(a), the false matching image numbers on the top 10 of ITQ, TPH and GTPH are 3, 4 and 0, when the hash codes are with 128 bits. While we only assign 32 bits for them, the corresponding correct matching numbers in top 10 are 3, 3, and



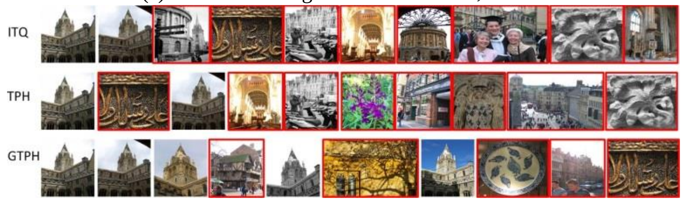
(a) Oxford Building benchmark dataset, 128 bits



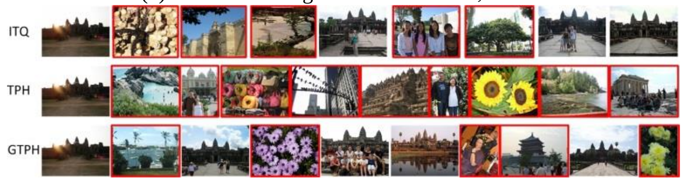
(b) Oxford Building benchmark dataset, 128 bits



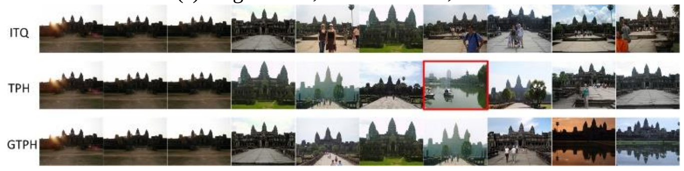
(c) Oxford Building benchmark dataset, 32 bits



(d) Oxford Building benchmark dataset, 32 bits



(e) Angkor Wat, GOLD dataset, 32bits



(f) Angkor Wat, GOLD dataset, 128bits

Fig. 3. Top 10 retrieval results of ITQ, TPH and our GTPH method. The first is the query and the most similar result. Incorrect images are marked in red.

5. In Fig.3(b), the correct answers in top 10 are 2, 2 and 5 in ITQ, TPH and GTPH. Similarly for the case in Fig.3(b) and (d), and in Fig.3(e) and (f), our GTPH based approach can find more correct matchings even with short hash codes.

Some noise pictures can be easily recognized in these result, like the 3rd of ITQ result in Fig.3(a) also appears in TPH in Fig.3(a) and (b). They are likely to be considered as similar images since they have too many repeated local feature points with a pattern, so that after hashing, in similarity calculating, there is more possibility for a query feature point to find a wrong one with a small Hamming distance. Adding geometry factor in hashing function training can separate principal features, and adding it in similarity evaluation adds global geometry information, which has made the retrieving focus on both global and local. From a local view, the feature points of the top of Angkor Wat may be similar to the heads of three people, but from a global aspect, features in these situations have different scale and orientation. Geometry score can eliminate the impact of noise matchings which makes the Hamming distance score higher than correct answers. This means, firstly noise pictures are tried to be pushed back by average Hamming distance; secondly they get smaller geometric score. Thus, results get much better.

F. Discussion

In this section, parameters β, b, c, relation between n and t are discussed. The parameter β denotes whether to utilize the geometric information in hashing function training. b and c are the corresponding factors of scale and orientation of geometric information in geometry enhanced similarity measurement. The results are shown in Fig.4, Fig.5, and Fig. 6. For GTPH, the algorithm complexity is the same with TPH as O(N), and the computing speed is slower as it involves more information, the precision improvement makes it worth.

(a) Geometry Parameters

We choose β=0.2, β=0.4, β=0.8, and a few b and c values to show the performance in Fig.4 and Fig.5.

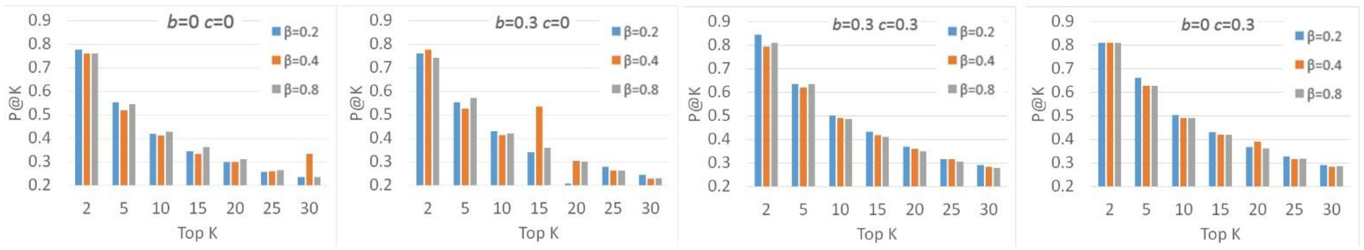
In Fig.4, GTPH gives a steady good performance, though fluctuating with parameters. In Fig.4(a), when b=0 and c=0, it is equal to not adding geometry information. It can be seen that when change b to 0.3 and keep c=0, the performance rises a little bit, while change c to 0.3 and keep b=0, it rises more. It means that either part of our geometry information is effective. Comparing the first and the fourth histogram of Fig.4(a), where the latter one is with b=0.3 and c=0.3, an obvious raise can be seen. It can also be found in Fig.4(b) that even the hashing representation uses shorter bit length, the geometry information works, as the fourth histogram in b=0.3 and c=0.5 performs best. The orientation of a SIFT feature is more sensitive than the scale, as the histogram peak of a point is easier to change than the scope, thus c is more sensitive than b. The values of these parameters are a little bit controversial and need to be adjusted by an amount of queries.

Different β's performance is shown in Fig.5, b=0, c=0. Overall, the best among three is $\beta = 0.4$, which goes for 0.800 at top 2. The result shows neither too big nor too small of β is recommended, since β represents the geometry information's impact in hashing function training. When it is too small, it will be the same with TPH, and when it is too big, it will not preserve enough data information.

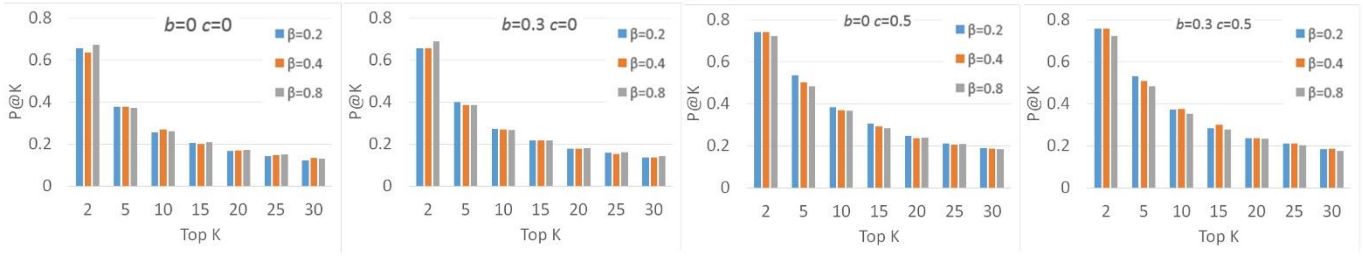
(b) Dataset Parameters

We have also discussed the parameter n and t's relation, and have experiment in GOLD dataset as in Fig. 6, with t=10%, t=2%, t=1%, and t=0.8% of n in β=0.8.

From Fig. 6, we show the impact of the amount of the training dataset $\mathbf{X}_{128\times n}$ in part B, section III. We cut off a certain percentage of each picture's feature points randomly and trained the hashing function with the same $\mathbf{T}_{128\times t}$. In part A of section V, $\mathbf{X}_{128\times n}$ is 118 times as large as $\mathbf{T}_{128\times t}$, which means $t = 0.8\%$ of n. The other values are designed to 10%, 2% and 1%.



(a) m = 128 Top 30’s mean precision comparison in different parameters



(b) m = 32 Top 30’s mean precision comparison in different parameters
Fig.4. mean precision comparison in different parameters

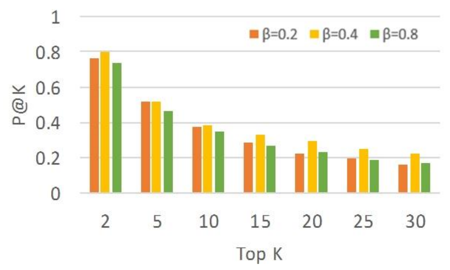


Fig.5. GOLD dataset, 32 bits, top 30’s mean precision comparison in different parameters

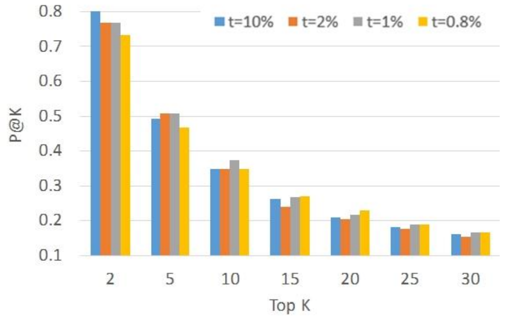


Fig.6. GOLD dataset, 32 bits, top 30’s mean precision comparison in different parameters

It can be seen that the size of training dataset does impact the precision in GOLD dataset. Basically, the more features involved in a dataset, the more robustness it will get, so it can be seen that $t = 0.8\%$ works, where $\mathbf{X}_{128\times n}$ consists of all feature points in Oxford Building benchmark dataset. It may be difficult to pick out enough principle useful feature points in training. Thus, making $\mathbf{X}_{128\times n}$ a little smaller to $t = 1\%$ performs better than $t = 0.8\%$, which means we need not to use all of the features in dataset D'. When $t = 2\%$ and $t = 10\%$, the precisions get down. It is possible that a smaller n makes the robustness down, and the closer t and n are, the easier to get features share the same neighbour set and keeps the topology and geometry relationship in training, leading to an unsteady raise in precision. As for time costs, obviously a smaller n saves time, but in the same programming condition, $t = 2\%$ costs 6703s, three times of time of $t = 10\%$, and $t = 2\%$ costs about 13914s, because of the computing of $\mathbf{M}$ need to traverse every feature in D' a few times. The convergence in III.D.(b) is nearly the same when m=32, for 194.82s in $t = 10\%$, 113.03s in $t = 2\%$, and 373.77s in $t = 1\%$. Optimising programme can make it more efficient, but the choosing of n and t need to be considered in practice.

(c) Effect of Geometry Enhancement

We have done the experiment in 128 bits in Oxford Building benchmark dataset, and the mean precision results are shown in TABLE 3 and Fig.7.

TABLE 3 is given to show the effect of each part of GTPH. TPH method preserves the topology information. GTPH without geometry enhanced similarity evaluation, leading to b=0 and c=0, performs better than TPH shows that adding geometry information in hashing function training works. The third column is TPH with geometry enhanced similarity evaluation. As it performs better than TPH, the precision of top 2 can reach 0.707. When geometry enhanced similarity is also added, b=0.3 and c=0.3, the mean precision can reach 0.845 at top 2.

To shows the geometry enhancement similarity evaluation is not only effective in GTPH, but also useful for other methods, we give out Fig.7. “With” means the data with geometry enhanced similarity evaluation, with b=0.3 and c=0.3; “without” means without that, leading to b=0, c=0. Obviously, every method has a great improvement. GTPH is better than TPH, and the performance of front result, like top 2, is even close to the SIFT matching precision. Even though ITQ seems good, it is while keeping the content information need parameter adjusting. When we adjust the parameters in TPH and GTPH, it is quite possible that the precision will be better than ITQ for our information preserving, according to the result in [11].

Enhancing geometry information makes the hashed features get better classification. By geometry enhanced similarity, this classification is adjusted due to the global character of images. The unity of these two steps makes GTPH stands out.

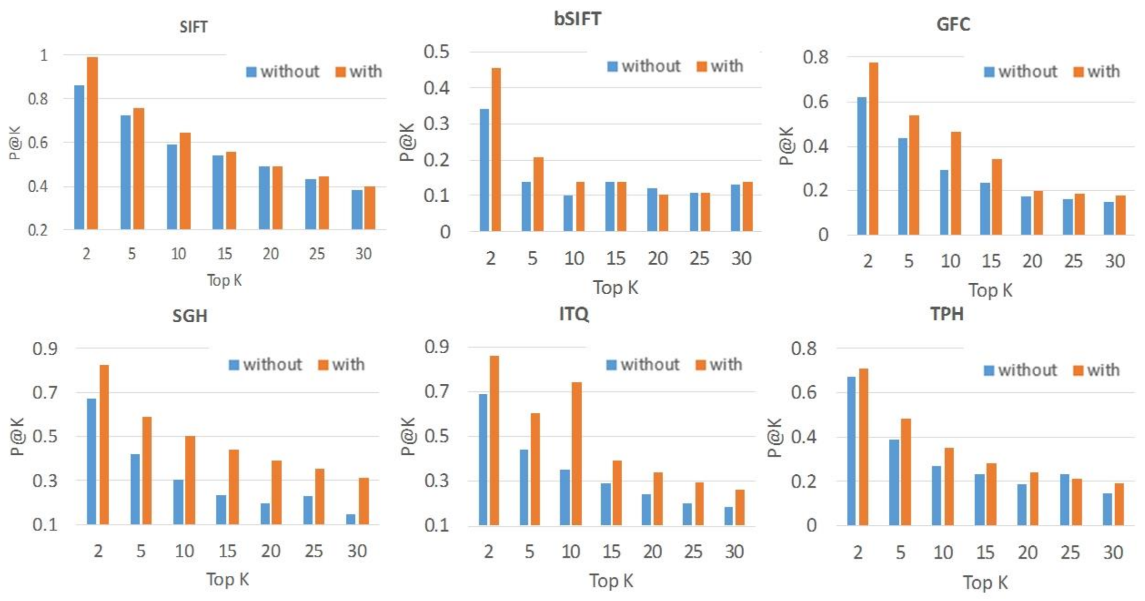


Fig.7. Oxford Building benchmark dataset, mean precision comparison of with and without geometry enhanced similarity evaluation

TABLE 3. 128 BITS, TOP 30'S MEAN PRECISION COMPARISON BETWEEN TPH AND GTPH

Top K	TPH (without)	TPH (with)	GTPH (without)	GTPH (with)
2	0.672	0.707	0.776	0.845
5	0.386	0.483	0.552	0.635
10	0.272	0.352	0.417	0.500
15	0.230	0.278	0.345	0.432
20	0.186	0.240	0.298	0.369
25	0.163	0.208	0.258	0.315
30	0.145	0.190	0.234	0.290

VI. CONCLUSION

This paper proposes Geometry and Topology Preserving Hashing method to acquire a hashing function considers both of geometry and topology of feature points, and then ameliorate the similarity evaluation to deal with the problem of inaccuracy in image retrieval. This method improves the shortcoming of only considering distance relationship of feature points' topology in training and the limitation of implementing in complex background dataset by measuring from a global aspect using local features to protect both of local and global features of an image, especially in tour recommendation situation. A significant improvement in precision with a comparison between SIFT, Binary SIFT, GFC, SGH, ITQ, TPH and our GTPH, along with the comparison of different factor values in Oxford Building benchmark dataset and GOLD are shown in experiment part.

In future work, we will further explore salient detection to improve retrieval performance.

REFERENCES

[1] Zhang, Y., Zhang, L., & Tian, Q. (2014). A prior-free weighting scheme for binary code ranking. *IEEE Transactions on Multimedia*, *16*(4), 1127-1139.

[2] Yang, X., Qian, X., & Mei, T. (2015). Learning salient visual word for scalable mobile image retrieval. *Pattern Recognition, 48*(10), 3093-3101.

[3] Wang, Y., Zhu, L., Qian, & X., Han, J. (2018). Joint Hypergraph Learning for Tag-Based Image Retrieval. *IEEE Transactions on Image Processing* (pp.4437-4451)

[4] Qian, X., Li, C., Lan, K., Hou, X., Li, Z., & Han, J. (2018). POI Summarization by Aesthetics Evaluation From Crowd Source Social Media. *IEEE Transactions on Image Processing* (pp.1178-1189)

[5] Qian, X., Wang, H., Zhao, Y., Hou, X., Hong, R., Wang, M., & Tang, Y. (2017). Image Location Inference by Multisaliency Enhancement. *IEEE Transactions on Multimedia* (pp.813-821)

[6] Qian, X., Lu, X., Han, J., Du, B., & Li, X. (2017). On Combining Social Media and Spatial Technology for POI Cognition and Image Localization. *Proceedings of the IEEE* (pp. 1937-1952)

[7] Tan, W., Yan, B., Li, K., & Tian, Q. (2015). Image retargeting for preserving robust local feature: application to mobile visual search. *IEEE Transactions on Multimedia, 18*(1), 1-1.

[8] Kordelas, G. A., Alexiadis, D. S., Daras, P., & Izquierdo, E. (2016). Content-based guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation. *IEEE Transactions on Multimedia, 18*(2), 155-170.

[9] Jian, M., & Jung, C. (2016). Semi-supervised bi-dictionary learning for image classification with smooth representation-based label propagation.*IEEE Transactions on Multimedia*, 1-1.

[10] Lu, D., Liu, X., & Qian, X. (2016). Tag-based image search by social re-ranking. IEEE Transactions on Multimedia, 18(8), 1628-1639.

[11] Zhang, L., Zhang, Y., Gu, X., Tang, J., & Tian, Q. (2014). Scalable similarity search with topology preserving hashing. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society,23*(7), 3025-3039.

[12] Zhang, L., Zhang, Y., Tang, J., Gu, X., Li, J., & Tian, Q. (2013). Topology preserving hashing for similarity search. *ACM International Conference on Multimedia* (pp.123-132).

[13] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(60), 91-110.

[14] Bay, H., Tuytelaars, T., & Gool, L. V. (2006). Surf: speeded up robust features. *Computer Vision & Image Understanding, 110*(3), 404-417.

[15] Luo, J., & Gwun, O. (2009). A comparison of sift, pca-sift and surf.*International Journal of Image Processing, 3*(4), 143-152.

[16] Wang, Q., Zhang, D., & Si, L. (2013). Weighted hashing for fast large scale similarity search. *ACM International Conference on Conference on Information & Knowledge Management* (pp.1185-1188).

[17] Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. 1-8.

[18] Yang, X., Qian, X., & Xue, Y. (2015). Scalable Mobile Image Retrieval by Exploring Contextual Saliency. *IEEE Transactions on Image Processing* (pp.1709-1721)

[19] Qian, X., Tan, X., Zhang, Y., Hong, R., & Wang, M. (2016). Enhancing Sketch-Based Image Retrieval by Re-Ranking and Relevance Feedback. *IEEE Transactions on Image Processing* (pp.195-208)

[20] Zhou, W., Li, H., Wang, M., Lu, Y., & Tian, Q. (2012). Binary SIFT: towards efficient feature matching verification for image search.*International Conference on Internet Multimedia Computing and Service*(pp.1-6).

[21] Zhou, W., Li, H., Lu, Y., & Tian, Q. (2013). Sift match verification by geometric coding for large-scale partial-duplicate web image search. *Acm Transactions on Multimedia Computing Communications & Applications,9*(1), 319-339.

[22] Lin, K., Yang, H. F., Hsiao, J. H., & Chen, C. S. (2015). Deep learning of binary hash codes for fast image retrieval. *Computer Vision and Pattern Recognition Workshops* (pp.27-35). IEEE.

[23] Xia, R., Pan, Y35Supervised hashing for image retrieval via image representation learning.

[24] Liong, V. E., Lu, J., Wang, G., & Moulin, P. (2015). Deep hashing for compact binary codes learning. *Computer Vision and Pattern Recognition*. IEEE.`

[25] Li, W. J., Wang, S., & Kang, W. C. (2015). Feature learning based deep supervised hashing with pairwise labels. *Computer Science*.

[26] Yang, X., & Qian, X. (2014). Spatial Verification for Scalable Mobile Image Retrieval. *ACM International Conference on Conference on Information and Knowledge Management* (pp.1903-1906). ACM.

[27] Yu, X., Zhang, S., Liu, B., & Zhong, L. (2013). Large Scale Medical Image Search via Unsupervised PCA Hashing. *Computer Vision and Pattern Recognition Workshops* (Vol.13, pp.393-398). IEEE.

[28] Zhao, Y., Qian, X., & Mu, T. (2015). Image Taken Place Estimation via Geometric Constrained Spatial Layer Matching. *MultiMedia Modeling*.

[29] Zhang, S., Tian, Q., Huang, Q., & Rui, Y. (2014). Embedding multi-order spatial clues for scalable visual matching and retrieval. *IEEE Journal on Emerging & Selected Topics in Circuits & Systems, 4*(1), 130-141.

[30] Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. *Twentieth Symposium on Computational Geometry* (Vol.34, pp.253-262).

[31] Wang, J., Shen, H. T., Song, J., & Ji, J. (2014). Hashing for similarity search: a survey. *Computer Science*.

[32] Andoni, A., & Indyk, P. (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Foundations of Computer Science Annual Symposium on, 51*(1), 117-122.

[33] Lv, Q., Josephson, W., Wang, Z., Charikar, M., & Li, K. (2007). Multi-probe LSH: efficient indexing for high-dimensional similarity search.*International Conference on Very Large Data Bases, University of Vienna, Austria, September* (pp.950-961).

[34] Ji, J., Li, J., Yan, S., Zhang, B., & Tian, Q. (2012). Super-bit locality-sensitive hashing. *Advances in Neural Information Processing Systems,1*, 108-116.

[35] Kulis, B., & Grauman, K. (2012). Kernelized locality-sensitive hashing.*Pattern Analysis & Machine Intelligence IEEE Transactions on, 34*(6), 1092-1104.

[36] Gong, Y., Lazebnik, S., Gordo, A., & Perronnin, F. (2011). Iterative quantization: a Procrustean approach to learning binary codes for large-scale image retrieval. *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (Vol.35, pp.2916-2929). IEEE Computer Society.

[37] Chu, L., Jiang, S., Wang, S., Zhang, Y., & Huang, Q. (2013). Robust spatial consistency graph model for partial duplicate image retrieval.*IEEE Transactions on Multimedia, 15*(8), 1982-1996.

[38] Vyshali, S., Subramanyam, M. V., & Raajan, K. S. (2015). Geometry preserving image retrieval using normalized scale coding. *International Conference on Electrical, Electronics, Signals, Communication and Optimization*. IEEE.

[39] Zhang, Y., Jia, Z., & Chen, T. (2011). Image retrieval with geometry-preserving visual phrases. *IEEE Conference on Computer Vision & Pattern Recognition* (Vol.42, pp.809-816).

[40] Jiang, Q. Y., & Li, W. J. (2015). Scalable graph hashing with feature transformation. *International Conference on Artificial Intelligence* (Vol.9, pp.331-337). AAAI Press.

[41] Shen, F., Shen, C., Liu, W., & Shen, H. T. (2015). Supervised discrete hashing. *Computer Science*, 37-45.

[42] Salakhutdinov, R., & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning, 50*(7), 969-978.

[43] Kulis, B., & Darrell, T. (2009). Learning to Hash with Binary Reconstructive Embeddings. *Advances in Neural Information Processing Systems 22:, Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada* (pp.1042--1050).

[44] Wang, J., Kumar, S., & Chang, S. F. (2012). Semi-supervised hashing for large-scale search. *Pattern Analysis & Machine Intelligence IEEE Transactions on, 34*(12), 2393-2406.

[45] Wang, J., Kumar, S., & Chang, S. F. (2010). Semi-supervised hashing for scalable image retrieval. *IEEE Conference on Computer Vision & Pattern Recognition* (Vol.23, pp.3424-3431).

[46] H. Liu, R. Wang, S. Shan, and X. Chen, “Deep supervised hashing for fast image retrieval,” in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 2064–2072.

[47] H.-F. Yang, K. Lin, and C.-S. Chen, “Supervised learning of semantics-preserving hash via deep convolutional neural networks,” in *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 40, no. 2, pp. 437–451, Feb. 2018

[48] M. Wang, W. Zhou, Q. Tian and H. Li, "A General Framework for Linear Distance Preserving Hashing," in *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 907-922, Feb. 2018.

[49] Y. Li, Y. Zhang, X. Huang, H. Zhu and J. Ma, "Large-Scale Remote Sensing Image Retrieval by Deep Hashing Neural Networks," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 950-965, Feb. 2018.

[50] Yang, X., Liu, L., Qian, X., Mei, T., Shen, J., & Tian, Q. (2014). Mobile visual search via hierarchical sparse coding. *IEEE International Conference on Multimedia and Expo* (pp.1-6). IEEE.

Chen Kang, She is a graduate student at SMILES LAB and School of Software Engineering, Xi’an Jiaotong University. She received the B.S. and M.E. degrees from Xi'an Jiaotong University, Xi'an, China, in 2014 and 2017 respectively. She is currently working towards the Ph.D. at Laboratoire des Signaux et Systèmes (L2S), Université Paris-Sud-CNRS-CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France. She is mainly engaged in the research of content based image retrieval, recommendation and quality assessment.

Li Zhu, He is an associate professor in School of Software, Xi’an Jiao-tong University. He received his M.S. and Ph.D. degrees from Xi’an Jiaotong University in 1995 and 2000, respectively. He received his B.S. degree from Northwestern Polytechnical University in 1989. His main research interests include multimedia processing & communication, parallel computing and networking.

Xueming Qian, He is with SMILES LAB, Xi’an Jiaotong University. (M’10) received the B.S. and M.S.degrees from Xi’an University of Technology, Xi’an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi’an Jiaotong University, Xi’an, China, in 2008. He was awarded the Microsoft Fellowship in 2006. From 1999 to 2001, he was an Assistant Engineer at Shannxi Daily. Since 2008, he has been an Associate Professor in the School of Electronics and Information Engineering, Xi’an Jiaotong University. Now, he is an Associate Professor in the School of Electronics and Information Engineering, Xi’an Jiaotong University. He is the Director of SMILES LAB. He was a Visiting Scholar at Microsoft Research Asia from August 2010 to March 2011. His research interests include social media big data mining and search. His research is supported by NSFC, Microsoft Research, and MOST. He is a member of the IEEE, ACM, and Senior Member of CCF.

Appendix A: Notations and Definitions

The Symbol	Meaning
n	The total number of selected feature points from image dataset.
D'	The feature points set of the image dataset.
$\mathbf{X}_{128\times n}$	Matrix consisting of all of the n SIFT descriptors.
num	The randomly chosen sample number.
x_i	A chosen sample.
k	The number of nearest neighbor points of a sample.
l	The number of non-neighbor points of a sample.
N_i	The nearest points and non-neighbour points.
T'	The training set.
t	A training sample.
$\mathbf{T}$	The training matrix.
$\mathbf{M}$	A matrix that takes the geometry and topology relationships among training samples.
$\mathbf{\Gamma}_r$	Similarity rank weighting matrix.
$\mathbf{\Gamma}_s$	Simi-weighting matrix.
$\mathbf{G}_r$	Geometry rank weighting matrix.
$\mathbf{G}_s$	Geometric similarity relation matrix.
$\bar{d_\iota}$	The average Euclidean distance to x_i.
σ	The average Euclidean distances of the descriptors of the dataset.
$ds(s_i, s_j)$	The normalised distance between the two scales of x_i and x_j.
$do(o_i, o_j)$	The normalised distance between the two orientations of x_i and x_j.
$\bar{s_\iota}$	Average scale difference.
$\bar{o_\iota}$	Average orientation difference.
$\mathbf{S}_r$	Scale deviation.
$\mathbf{O}_r$	Orientation deviation.
σ_s	Average Euclidean distance of the scales of SIFT points in D'
σ_o	Average Euclidean distance of the orientations of SIFT points in D'
γ	The weight to balance the matrix $\mathbf{\Gamma}_r$ and $\mathbf{G}_r$ with neighbourhood relationship preserving matrix $\mathbf{\Gamma}_s$ and $\mathbf{G}_s$,
β	The weight for the importance of the 128-dimension feature's topology information to the geometry information.
$\mathbf{A}$	An eigen-decomposition of matrix.
$\mathbf{W}$	Project matrix.
$\mathbf{R}$	An optimal mapping that can transfer the features.
$\mathbf{Y}_{m\times n}$	The short binary matrix, the hashed **X.**
$\mathbf{Q}(\mathbf{Y}, \mathbf{R})$	Tthe quantisation loss.
$\mathbf{U}, \mathbf{S}$ and $\mathbf{V}$	Result matrices of SVD decomposition.
Input	The input feature matrix.
Q	The feature point set in the query image.
Pi	The i_{th} retrieval image's feature set.
$\mathrm{H}(q, p)$	The hamming distance between two feature points q and p.
r	A threshold.
HDS_i	Hamming distance score (HDS) of Pi.
I	Number of retrieving images.
z	The number of feature points in image Pi.
ω	The number of reliable-matched feature points number.
SS_i	A scale score between image P_i and query Q.
OS_i	A orientation score between image P_i and query Q.
b	Weight balance parameter.
c	Weight balance parameter.
SCORE_i	Evaluation score of image P_i.