Multimed Tools Appl
DOI 10.1007/s11042-014-1977-3

Memory efficient large-scale image-based localization

Guoyu Lu · Nicu Sebe · Congfu Xu · Chandra Kambhamettu

Abstract Local features have been widely used in the area of image-based localization. However, large-scale 2D-to-3D matching problems still involve massive memory consumption, which is mainly caused by the high dimensionality of the features (e.g. 128 dimensions of SIFT feature). This paper introduces a new method that decreases local features' high dimensionality for reducing memory capacity and accelerating the descriptor matching process. With this new method, all descriptors are projected into a lower dimensional space through the new learned matrices that are able to reduce the *curse of dimensionality* in the large scale image-based localization. The low dimensional descriptors are then mapped into a Hamming space for further reducing the memory requirement. This study also proposes an image-based localization pipeline based on the new learned Hamming descriptors. The new learned descriptor and the localization pipeline are applied to two challenging datasets. The experimental results show that the proposed method achieves extraordinary image registration performance compared with the published results from state-of-the-art methods.

Keywords Image-based localization · Large scale imagery · SIFT · Hamming descriptor · Dimensionality reduction

G. Lu (✉) · C. Kambhamettu
Video/Image Modeling and Synthesis Lab, University of Delaware, Newark, DE 19711, USA
e-mail: luguoyu@udel.edu

C. Kambhamettu
e-mail: xucongfu@cs.zju.edu.cn

N. Sebe
Department of Information Engineering and Computer Science, University of Trento,
38100 Trento, Italy
e-mail: sebe@disi.unitn.it

C. Xu
Institute of Artificial Intelligence, Zhejiang University, Hangzhou, 310027, People's Republic of China
e-mail: chandrak@udel.edu

Published online: 08 May 2014

1 Introduction

Image-based localization is one of the dominant localization methods used in mobile devices. Given an image captured by a mobile phone, an image-based localization system can provide the users their position information. Compared with GPS or cell tower triangulation, the image-based localization system could provide more precise information and work more reliably in the areas of large buildings [43, 49]. This method [37, 43, 57] originally achieves the localization by searching through an image database to find out the best matched image to the query image. Current image-based localization methods [19, 27, 38] match a 2D query image to a 3D model built by Structure-from-Motion reconstruction. By allowing further camera orientation estimation, this localization method achieves higher accuracy. In this localization process, local features such as SIFT feature [28] are now widely used for image-based localization. SIFT is invariant to scaling, rotation and partly illumination change, and it has been successfully applied to object recognition, 3D reconstruction, motion tracking and many other computer vision tasks. Due to these beneficial properties, SIFT feature is also widely used in the image-based localization problems. However, caused by the high dimensionality of SIFT features, a large memory is still required to store all the descriptors and points' information of a 3D Structure-from-Motion reconstruction model for the large-scale 2D-to-3D matching problems. This is especially critical for mobile devices whose available memory is limited. Due to the fact that SIFT contains 128 dimension real-value numbers and in addition a 3D model obtained by Structure-from-Motion reconstruction usually contains at least several million SIFT descriptors, the required memory for image-based localization is extremely large. Because the whole localization process consumes a large space to store all the descriptors, we want to find a way to save the space for storing the descriptors. Also, the lower dimensionality of SIFT feature will make the computation in localizing a query image faster. Because of these reasons, it would be ideal to learn a low dimensional binary code to represent the local feature.

In this paper, we present an image-based localization pipeline that uses LDAHash [44] and TRRF [54] to project the original SIFT descriptor into a low dimensional Hamming space and adopt the new learned descriptor into the adjusted localization pipeline. LDAHash and TRRF are methods for learning small descriptors that are applied to image retrieval problems. By learning new projection matrices, LDAHash and TRRF project the descriptors into a lower dimensional space to accelerate the matching process with the use of less memory. This method improves the matching performance through learning a new projection matrix that reduces the *curse of dimensionality* in fast matching application. In the process of learning projection matrices, we introduce the nearest neighbor negative descriptor pairs to make our projection matrix separate better positive and negative descriptor pairs. For reducing the memory requirement, we will further map the low dimensional descriptors into a Hamming space, which reduces the size for storage and simplifies the similarity computation. We employ LDAHash and TRRF in a more challenging localization task instead of image retrieval problems, for the descriptors belonging to the same 3D point are much denser than the descriptors belonging to the 2D points in common image retrieval problems. The tests on 2 challenging datasets show that our localization pipeline uses less than 10 % memory (for our 96 dimensional Hamming descriptor) to achieve similar accuracy as the state-of-the-art methods. For the Hamming descriptor, we quantize the descriptor to as low as 64 dimensions. For the real-value projected descriptor, we reduce the dimensionality to 32 dimensions (25 % memory compared to the original method), which still yields high accuracy.

We learn different dimensional projection functions and thresholds that can achieve lower dimensional Hamming descriptors for the 2 datasets. By assigning descriptors in the projected space to different visual words, we also learn the threshold for each visual word.

In the practical image-based localization task, we test the new learned descriptors from several different approaches. We test the descriptors in both Hamming space and the projected space using nearest neighbor search. For Hamming space, we also learn an individual threshold for each visual word and search the matching point through the visual word.

The paper is organized as follows. Section 2 briefly introduces the related work of image-based localization and the techniques for learning descriptors. Section 3 discusses the LDAHash and TRRF methods, including our adjustment of the two methods. Section 4 introduces the image-based localization method we use. Section 5 presents our localization results and the analysis corresponding to each experiment. Finally, Section 6 concludes the paper.

2 Related work

Image-based localization is widely used in localization problems. Different from GPS solutions, image-based localization solution can still be employed in weak GPS signal area. Image localization was introduced by Robertson et al. [37]. They utilize a database of building facades views to calculate the pose of the query image provided by the users. The images in the database are associated with a 3D coordinate system. Similar work is done by Shao et al. on urban scene image retrieval problems [40]. Zhang et al. [57] provide the localization information in the urban area by directly searching in an image database for the closest image in the descriptor space. Steinhoff et al. [43] build their localization model on a vocabulary tree [33] to achieve real-time pose estimation with a high accuracy. Similarly, Schindler et al. [39] select the vocabulary using the most informative features to improve the image retrieval performance on a large street side image database. Xiao et al. [51] improve the object localization accuracy by using a bag-of-words method together with the geometric verification. Beltran et al. [3] present a virtual globe tool for searching and visualizing geo-referenced media resources based on the combination of search technologies, description languages of geo-referenced media annotations and visualization techniques. By computing the solar zenith angle to the image intensity, Jacobs et al. [20] realize the coarse geo-localization based on the global network of outdoor webcams. [52] designs a weighted network Voronoi diagram and a multilevel range search query processing system that retrieves a set of objects located in some specified region within the searching range. With the development of Structure-from-Motion (SfM) techniques for reconstructing 3D scene cloud points [8, 11, 42], the 3D SfM model was used for image-based localization in improving localization accuracy. Irschara et al. [19] propose to retrieve images containing most descriptors matching the 3D points and Li et al. [27] realize the 3D-to-2D matching through the mutual visibility information. By generating virtual views in 3D space, Wendel et al. [49] introduce an algorithm of monocular visual localization for the micro aerial vehicles. Methods proposed in [9] and [7] estimate the camera pose based on the online-build 3D model based on the Simultaneous Localization and Mapping (SLAM) algorithm [26, 41], which is limited to small scenes. Sattler et al. [38] propose a framework to directly match the descriptors extracted from 2D images to the descriptors from the 3D model in order to improve the localization accuracy. The proposed localization framework accelerates the matching process and achieves a high image registration rate. Yu et al. [56]

propose an indoor localization method by detecting and tracking people that combines the information of color, person detection, face recognition and non-background information. As for the 3D object retrieval problem, [13] proposes an interactive 3D object retrieval scheme to fast and accurately retrieve the 2D views of 3D objects. They point out that, instead of using the whole set of 2D views, using query views that are judged to be the most discriminant ones based on the labeling information can give faster and more accurate result. Wang et al. [47] further improved the retrieval accuracy by building the probabilistic models for each object based on the distribution of its views. The distance between two objects is determined by the upper bound of the Kullback-Leibler divergence of the corresponding probabilistic models. Recent research [12] on image retrieval proposes that simultaneously utilizing both visual and textual information to estimate the relevance of user tagged images can retrieve relevant images more effectively. The visual and textual information are represented by bag-of-visual words and bag-of-texture words. The relevance estimation is determined by a hypergraph, where vertices represent images and hyperedges represent visual or textual terms. The weights of the hyperedges are updated by a learning process with the use of a set of pseudo-positive images. Different from the cross media retrieval (like using audio to query images) [55] where the different modalities are captured by different devices though various methods, the 3D model is reconstructed from the 2D images, either by Structure-from-Motion reconstruction or stereo reconstruction.

Descriptors of local features usually have high dimensionality, such as 128 dimensions for SIFT feature [28] and 64 dimensions for SURF feature [2, 22]. When dealing with small number of images, the high feature dimensionality may not be an issue. However, for large-scale image retrieval problems, it would consume quite a lot of space to store all the descriptors. Large amount of research has been conducted to compact the high dimensional descriptors. Based on Principal Component Analysis [21], Yan et al. [23] reduce the descriptors' dimensionality and demonstrate that PCA-SIFT is more robust to image deformation. Hua et al. [18] adopt a discriminative approach from labeled matching and non-matching pairs to learn a lower dimensional embedding. Similar work is done by [50] to learn descriptors that can utilize the DAISY [45] configuration. Making use of Linear Discriminates Analysis [31] and Powell minimization [35], Brown et al. [6] reduce the descriptors' dimensionality by both linear and nonlinear transforms. Philbin et al. [34] distinguish the original descriptor pairs into three groups which are positive pairs, nearest neighbor negative pairs and random negative pairs. A margin-based cost function is built according to these three groups of pairs. The projection matrix is learned by minimizing the cost function. Yang et al. [54] formulate a trace ratio optimization problem and propose an efficient algorithm to solve the problem, by realizing a lower dimension data projection. Han et al. [14] propose a framework of sparse unsupervised dimensionality reduction for multiple view data to search a low-dimensional optimal consensus representation from multiple heterogeneous features by multiview learning. As in many cases, positive examples in supervised learning is limited, Ma et al. [29, 30] infer knowledge from other multimedia resources in the application of event detection using few positive examples. Han et al. [15] add a joint $l_{2,1}$-norm on multiple feature selection matrices to ensemble different classifiers loss function into a joint optimization framework to overcome the overfitting problem during feature selection process when only a few labeled data per class are available.

Through hashing method, descriptors could be projected from Euclidean space to Hamming space. In this way, the space for storing descriptors can be greatly reduced. Kulis & Grauman [25] generate locality-sensitive hashing for arbitrary kernel functions and show how one can use local patch descriptors such as SIFT in indexing. [24] introduces a scalable

coordinate-descent algorithm to learn functions based on minimizing the error caused by the binary embedding, which reduces the lost information between the original space and the Hamming space. Just by selecting a subset of eigenvectors of graph Laplacian, Weiss et al. [48] propose Spectral Hashing to compute a binary code for a new data point. Raginsky & Lazebnik [36] propose a distribution-free encoding scheme for random projections. Through this scheme, the Hamming distance between the vectors could be related to the vectors' shift invariant kernel value. Yagnik et al. [53] present an embedding method (WTAHash) based on the partial order statistics. Their embedding method can be extended to the case of polynomial kernels. They also show the simplicity in the implementation of their algorithm and the effectiveness in the distance computation. The well-known MinHash [4, 5] method is the special case of their Hashing method when applied to binary vectors. Strecha et al. [44] combine both projecting and hashing functions together to propose LDAHash method, with the purpose of learning lower dimensional binary descriptors.

Our method is employing a 2D-to-3D matching and image registration approach similar to [38]. But our method is more efficient in memory and storage consumption, as all the descriptors are projected into a lower dimensional Euclidean space and further mapped to the Hamming space. Based on our new distance calculation methods, the image registration performance can outperform most of the state-of-the-art methods with much smaller memory usage.

3 Descriptor learning

3.1 Feature dimensionality reduction

One of the main issues for the large memory requirement is the high dimensionality of the SIFT feature. To address this problem, we learn a new projection function to convert the SIFT descriptor's dimensionality into a lower dimensional Euclidean space, followed by mapping the low dimensional real-value descriptors into a binary space. In projecting SIFT features, we use two kinds of projection functions. One is derived from the original LDAHash method, namely NNLDAHash (nearest neighbor negative LDAHash) and another one is the Trace Ratio Relevance Feedback method [54]. Additionally, through the learned projection matrix, the distance of the descriptors belonging to the same point *(positive descriptor pair)* is minimized, and the distance of the descriptors belonging to different points *(negative descriptor pair)* is maximized.

3.1.1 NNLDAHash method

LDAHash introduces 2 projection functions, Differences of Covariance (DIF) and Linear Discriminant Analysis (LDA) to distinguish the positive and negative descriptors; here LDA is not the classical LDA method. Since the final projection function resembles the classical LDA method, the authors named their proposed approach as LDA. DIF is claimed to achieve higher image retrieval accuracy. However, we found that the performance obtained by DIF is heavily dependent on the appropriate choice of the relevant parameters used for the weights' assignment. Inappropriate weights' values may result in the failure of the eigen-decomposition for the real-value scope, which will make the projection matrix contain complex number entries. This would result in a significant issue for generating the projection matrix. Thus, the DIF method is under the risk of failing to learn new descriptors

through the projection matrix. Nevertheless, the LDA method can better separate positive and negative pairs without the dependence of parameters, where negative descriptors are randomly selected. However, the distance between the positive descriptors and the random selected descriptors is relatively large. In most cases, the mismatched descriptors are coming from the nearest neighbor negative descriptors to the query descriptors, which are defined as the retrieved descriptors not fitting the RANSAC verification in [34]. Meanwhile, the matching descriptors fitting RANSAC are considered as positive descriptors. So our goal is to distinguish the positive descriptors and the negative nearest neighbor descriptors. Nonetheless, we cannot throw away the random negative descriptors when learning the projection function as they may overlap with the nearest neighbor descriptors after projection. The learning process is illustrated in the following equations.

A loss function is used in the proposed projection function to reduce the mismatching between positive and negative descriptor pairs in (1).

$$\begin{aligned} L = \alpha \mathbb{E}\left\{||P * X_{Pos} - P * X'_{Pos}||^2 | Pos\right\} \\ - \beta \mathbb{E}\left\{||P * X_{NNeg} - P * X'_{NNeg}||^2 | NNeg\right\} \\ - \mathbb{E}\left\{||P * X_{RNeg} - P * X'_{RNeg}||^2 | RNeg\right\} \end{aligned} \tag{1}$$

where:

X_{Pos}, X'_{Pos} are two descriptors' vectors, which form a positive descriptor pair;
X_{NNeg}, X'_{NNeg} are two descriptors' vectors forming a nearest neighbor negative descriptor pair;
X_{RNeg}, X'_{RNeg} are two descriptors' vectors forming a random negative descriptor pair;
Pos and Neg represent the positive pairs and negative pairs respectively;
α and β are the weight parameters for positive pairs and nearest neighbor negative pairs;
$\mathbb{E}$ is the operation for calculating the expectation for the projected descriptor pairs' distance;
P is a projection matrix.

Take SIFT descriptor as an example. After 3D SfM reconstruction, all the points in the 3D space will have at least 2 associated SIFT descriptors. Here X_{Pos} and X'_{Pos} are two 128 dimensional vectors from the same point (positive). X_{NNeg} and X'_{NNeg} are two SIFT descriptor vectors from different points, which are the nearest neighbors in the Euclidean searching space (nearest neighbor negative). X_{RNeg} and X'_{RNeg} are two descriptors from two different points and not the nearest neighbor in the searching space (random negative). If our goal is to achieve 64 dimensional quantized descriptors, the projection matrix P will be a $64 * 128$ dimensional matrix. Here, we denote $\mathbb{E}\left\{||X - X'||^2|\cdot\right\}$ with a covariance matrix by $\sum$ and substitute (1) as:

$$\begin{aligned} L &= \alpha\, tr\left\{P \sum\nolimits_{Pos} P^T\right\} - \beta\, tr\left\{P \sum\nolimits_{NNeg} P^T\right\} - tr\left\{P \sum\nolimits_{RNeg} P^T\right\} \\ &= \alpha\, tr\left\{P \sum\nolimits_{Pos} P^T\right\} - tr\left\{P(\beta \sum\nolimits_{NNeg} + \sum\nolimits_{RNeg}) P^T\right\} \end{aligned} \tag{2}$$

where $\sum_{Pos}$, $\sum_{NNeg}$ and $\sum_{RNeg}$ are respectively the covariance matrix of positive descriptors, the covariance matrix of nearest neighbor negative descriptors and the covariance matrix of random negative descriptors. To utilize the conjunctive closure [17] formed by

the descriptors belonging to the same point, we use the points with more than 4 descriptors to generate the positive descriptor pairs. We use descriptor pairs that are the nearest neighbors in the searching space but from different points as the nearest neighbor negative descriptor pairs. The random negative descriptor pairs are selected randomly from different points' descriptors but not the nearest neighbors. After randomly taking enough negative samples (not less than the positive descriptor pairs), the negative descriptors' distance will be much larger than positive descriptors' distance (around 10 times larger). After the inverse operation on the negative covariance matrix (we will describe this later), the effect on the random selection of negative descriptor pairs will be pretty small, which will not affect the experiment result in localization.

To avoid the overfitting problem, here $\sum_{Pos}$, $\sum_{NNeg}$ and $\sum_{RNeg}$ are covariance matrices for all the positive descriptor pairs and the selected negative descriptor pairs instead of each descriptor pair's covariance matrix. In computing $\sum_{Pos}$, $\sum_{NNeg}$ and $\sum_{RNeg}$, we sum up each descriptor pair's covariance matrix and then divide the sum value by the descriptor pair number. Since this process is performed iteratively for each descriptor pair, it will not increase the memory cost. Correspondingly, P is also a global projection matrix rather than each descriptor pair's projection matrix. Then storing P's memory will also not cause the memory burden.

We denote $\left(\beta\sum_{NNeg} + \sum_{RNeg}\right)$ as $\sum_{SNeg}$ to represent the sum of the two kinds of negative descriptor pairs' covariance matrix. $\beta = 2$ in our experiments. And (2) will became:

$$L = \alpha tr\left\{P\sum\nolimits_{Pos}P^T\right\} - tr\left\{P\sum\nolimits_{SNeg}P^T\right\} \tag{3}$$

The coordinates are transformed by pre-multiplying $\sum_{SNeg}{}^{(-1/2)}$ and $\sum_{SNeg}{}^{(-T/2)}$, so that the second term of (2) turns into a constant.

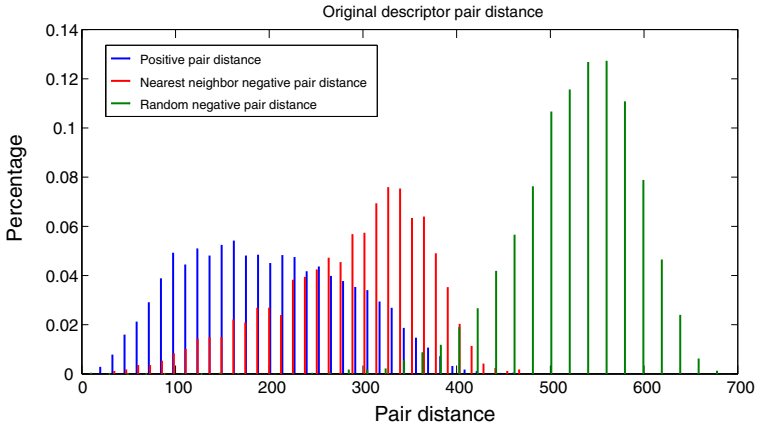
$$\begin{aligned}\widetilde{L} &\propto tr\left\{P\sum\nolimits_{SNeg}^{-1/2}\sum\nolimits_{Pos}\sum\nolimits_{SNeg}^{-T/2}P^T\right\}\\ &= tr\left\{P\sum\nolimits_{Pos}\sum\nolimits_{SNeg}^{-1}P^T\right\}\\ &= tr\left\{P\sum\nolimits_{R}P^T\right\}\end{aligned} \tag{4}$$

$$\sum\nolimits_{R} = \sum\nolimits_{Pos}\sum\nolimits_{SNeg}^{-1} \tag{5}$$

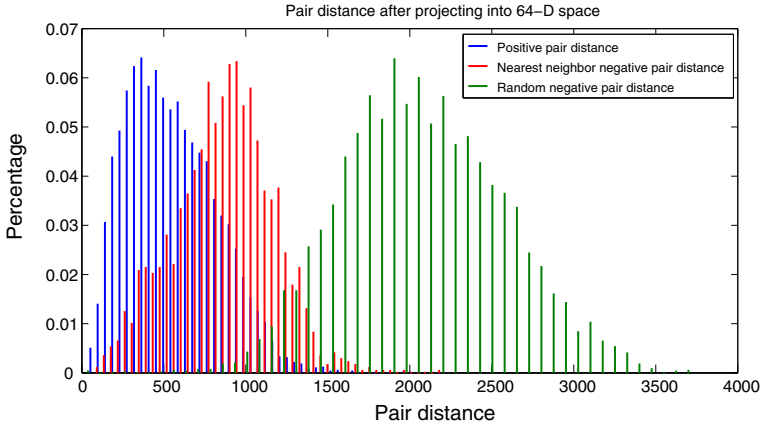
$\sum_R$ is the ratio between the positive descriptor pairs' covariance and the negative descriptor pairs' covariance. Since $\sum_{Pos}$ and $\sum_{SNeg}$ are both symmetric positive semi-definite matrices, $\sum_R$ is also a symmetric positive semi-definite matrix, which allows for eigendecomposition:

$$\sum\nolimits_{R} = USU^T \tag{6}$$

S is the non-negative diagonal matrix composed by the eigenvalues of the covariance matrix R and U is the corresponding eigenvectors for each eigenvalue. The projection matrix is composed by the m eigenvectors with the *smallest corresponding eigenvalues*, which minimizes the loss function. m is the dimensionality of the descriptors after projection. The results of the positive, nearest neighbor negative and random negative descriptor pairs' distance before and after the projection are plotted in Fig. 1. The descriptor pairs are projected into a 64-dimensional space.



(a)The original pair distance before the projection.



(b)The new pair distance after the projection.

Fig. 1 Descriptor pairs' distance histogram before and after they are projected into a new space, where X-axis is the distance of the descriptor pairs while Y-axis represents the percentage

As shown in Fig. 1, the distribution of positive descriptor pairs becomes much more compact after the projection, compared to the nearest neighbor negative pairs and the random negative pairs. Meanwhile, negative descriptor pairs are distributed more sparsely. More importantly, the distribution distance between positive and negative descriptors is enlarged. This result indicates that the projection function will certainly improve the matching performance in a low dimensional space. In the whole process of computing projection matrix P, the time complexity is $O(n)$, where n is the descriptor pair number and the space complexity is O(1), as we only keep one descriptor pair in the memory when computing the covariance matrix.

3.1.2 Trace ratio relevance feedback method

Wang et al. [46] showed that trace ratio outperforms ratio trace in dimensionality reduction. For this reason, we want to use trace ratio to further improve the image retrieval accuracy

using the projected descriptors. Trace Ratio Relevance Feedback method (TRRF) [54] is making use of the relevance feedback, positive and negative examples marked by the user, to refine the multimedia representation. The basic idea is to minimize the positive examples' distance and maximize the negative examples' distance. The positive examples' distance is defined by (7).

$$\begin{aligned}&\sum_{(x,x')\in Pos}\left(W^Tx-W^Tx'\right)^T\left(W^Tx-W^Tx'\right)\\&=\sum_{(x,x')\in Pos}Tr\left[W^T(x-x')(x-x')^TW\right]\\&=Tr(W^TD_{Pos}W)\end{aligned}\tag{7}$$

where D_{pos} equals to $\sum_{(x,x')\in Pos}(x-x')(x-x')^T$ and Tr is the trace operator. (x, x') is a positive descriptor pair and W is the projection matrix. In our method, we separate negative examples into two parts, nearest neighbor negative descriptor pairs and random negative descriptor pairs. The negative examples' distance is defined as (8).

$$\begin{aligned}&\beta\sum_{(x,x')\in NNeg}\left(W^Tx-W^Tx'\right)^T\left(W^Tx-W^Tx'\right)\\&+\sum_{(x,x')\in RNeg}\left(W^Tx-W^Tx'\right)^T\left(W^Tx-W^Tx'\right)\\&=\beta\sum_{(x,x')\in NNeg}Tr\left[W^T(x-x')(x-x')^TW\right]\\&\quad+\sum_{(x,x')\in RNeg}Tr[W^T(x-x')(x-x')^TW]\\&=Tr\left[W^T\beta D_{NNeg}W\right]+Tr\left[W^TD_{RNeg}W\right]\\&=Tr\left[W^T(\beta D_{NNeg}+D_{RNeg})W\right]\\&=Tr\left(W^TD_{SNeg}W\right)\end{aligned}\tag{8}$$

$\sum_{(x,x')\in NNeg}$ and $\sum_{(x,x')\in RNeg}$ are separately representing the nearest neighbor negative examples' distance and random negative examples' distance. β is the weight between the nearest neighbor negative examples and the random negative examples. We set $\beta = 2$ in our experiments. $D_{SNeg} = \beta D_{NNeg} + D_{RNeg}$ is the sum of the nearest neighbor negative examples' distance and the random negative examples' distance. The trace ratio optimization problem is proposed by (9)

$$\max_{W^TW=I}\frac{Tr(W^TD_{SNeg}W)}{Tr(W^TD_{Pos}W)}\tag{9}$$

Wang et al. [46] proposes to perform eigen-decomposition of $(D_{SNeg}-\lambda D_{Pos})$ and select d eigenvectors as the projection matrix W to achieve the maximization effect proposed by (9), where d is the dimensionality we want to achieve and W is initialized as an arbitrary orthogonal matrix. λ is defined in (10).

$$\lambda=\frac{Tr(W^TD_{SNeg}W)}{Tr(W^TD_{Pos}W)}\tag{10}$$

The eigen-decomposition operation is interactively repeated until W converges. As the eigen-decomposition is relatively time-consuming for repeating many times, Yang et al. [54]

propose to use the following two steps to substitute the iterative eigen-decomposition operation.

$$\eta = \frac{Tr(W^T D_{SNeg} W)}{Tr(W^T D_{Pos} W)} \tag{11}$$

$$W = e^T (D_{SNeg} - \eta D_{Pos}) e \tag{12}$$

e is the eigen-vector by the first time eigen-decomposition of $(D_{Neg} - \lambda D_{Pos})$. This process stops until W converges. As this method avoids repeating the eigen-decomposition operation, computing W is much faster than the method proposed by [46]. Trace ratio relevance feedback is used in parallel with NNLDAHash. After our derivation, the final projection matrix is presented in (12). D_{SNeg} and D_P are two matrices of descriptors' distance. We sum up all the square distance of positive descriptor pairs and then divide the positive descriptor pair number to get the covariance matrix of positive descriptor pairs. In this way, we generate D_P. D_{SNeg} is composed by another two Distance matrices, D_{NNeg} and D_{RNeg}. D_{NNeg} is given the weight of β while summing up the matrices of D_{NNeg} and D_{RNeg}. D_{NNeg} and D_{RNeg} are calculated in the same way of D_P. Our NNLDAHash is a ratio trace method, which is learned by the idea of decreasing the distance of positive descriptor pairs and increasing the distance of negative descriptor pairs. Compared with NNLDAHash, TRRF method's image registration number can increase dramatically with the increase of descriptor's dimensionality, as shown in Table 5.

Different from our learning methods, principal component analysis (PCA) is an unsupervised learning approach, which cannot take advantages of our learned positive and negative descriptor pairs. Meanwhile, PCA takes a lot of memory during the learning process. The classic Linear Discriminant Analysis (LDA) assumes that the variance in each group is approximately equal. However, the difference of covariance matrices for our positive and negative descriptor pairs is large (negative pairs covariance matrix is one order of magnitude larger). That makes our learning method better than the classic LDA. Meanwhile, both PCA and LDA are formulated in the form of ratio trace $max_W(Tr(W^T BW)^{-1}(W^T AW))$, whose refined multimedia vector representations are worse than TRRF method due to the the deviation from the original objectives [46]. We compare the experiment results of trace ratio and ratio trace in Table 7 of Section 5.

3.2 Hamming descriptor projection

To reduce the computational cost, we further map the low dimensional descriptors into the Hamming space. The computation in Hamming space is much easier and faster than in the Euclidean space. Most importantly, the memory required for storing descriptors can be significantly reduced if real-value descriptors are transformed into a Hamming space. Hereby, the projection from a real-value space to a Hamming space is formulated as:

$$y = \begin{cases} 0, & x <= T \\ 1, & x > T \end{cases} \tag{13}$$

where x is the projected real number descriptor, y is the descriptor mapped into a Hamming space, and T is the threshold to be learned to ensure that the projected Hamming descriptors in the greatest extent keep the original real-value descriptors' property.

The idea used here for optimizing T is to either minimize the false matching rate or maximize the true matching rate. In this study, we choose to optimize T by minimizing the

false matching rate. The false negative (FN) rate after mapping descriptors to the Hamming space can be calculated by the following equation:

$$\begin{aligned} FN(T) &= Pr\left\{\min\left(x_{Pos}, x'_{Pos}\right) < T \leq \max\left(x_{Pos}, x'_{Pos}\right)|Pos\right\} \\ &= Pr\left\{(\min(x_{Pos}, x'_{Pos} < T)|Pos\right\} + 1 \\ &\quad - Pr\left\{(\max(x_{Pos}, x'_{Pos}) < T)|Pos\right\} \\ &= cdf\left\{\min(x_{Pos}, x'_{Pos})|Pos\right\} \\ &\quad - cdf\left\{\max(x_{Pos}, x'_{Pos})|Pos\right\} \end{aligned} \tag{14}$$

Here, x_{Pos} and x'_{Pos} are descriptors after projecting in a lower dimensional Euclidean space. min and max correspond to selecting the smaller and larger value between x_{Pos} and x'_{Pos}. cdf is a cumulative distribution function with regard to each T value, representing the probability density accumulation. As x_{Pos} and x'_{Pos} are from the positive descriptor pairs, the value in the same dimension should be mapped to the same Hamming value after comparing with the threshold. However, since the smaller value between x_{Pos} and x'_{Pos} (obtained by the min operation) is smaller than the threshold T and the larger value (obtained by max operation) is larger the T, the corresponding value in Hamming space is different, which forms false negative pairs in Hamming space. Our goal is to reduce the number of false negative pairs.

False positive pairs are separated into the nearest neighbor false positive descriptor and the random false positive descriptor as well. The nearest neighbor false positive rate is computed by (15):

$$\begin{aligned} NFP(T) = Pr\Big\{&\min(x_{NNeg}, x'_{NNeg}) \geq T \\ &\cup \max\left(x_{NNeg}, x'_{NNeg}\right) < T\Big)|NNeg\Big\} \\ = 1 &- cdf\left(\min\left(x_{NNeg}, x'_{NNeg}\right)|NNeg\right) \\ &+ cdf\left(\max\left(x_{NNeg}, x'_{NNeg}\right)|NNeg\right) \end{aligned} \tag{15}$$

Similarly, the random false positive rate is computed in (16):

$$\begin{aligned} RFP(T) = Pr\Big\{&\min\left(x_{RNeg}, x'_{RNeg}\right) \geq T \\ &\cup \max\left(x_{RNeg}, x'_{RNeg}\right) < T\Big)|RNeg\Big\} \\ = 1 &- cdf\left(\min\left(x_{RNeg}, x'_{RNeg}\right)|RNeg\right) \\ &+ cdf\left(\max\left(x_{RNeg}, x'_{RNeg}\right)|RNeg\right) \end{aligned} \tag{16}$$

The overall false matching rate is given by:

$$F(T) = FN + NFP + RFP \tag{17}$$

We want to find the T that minimizes the value of F. The distance between three classes of descriptor pairs after mapping to the Hamming space is shown in Fig. 2.

Figure 2 clearly shows that the Hamming descriptors maintain the property of the lower dimensional projected descriptors whose positive descriptors' distance is reduced and negative descriptors' distance is enlarged. Previously in Fig. 1, the histograms display the original SIFT descriptor pairs distance distribution and the projected real-value descriptor

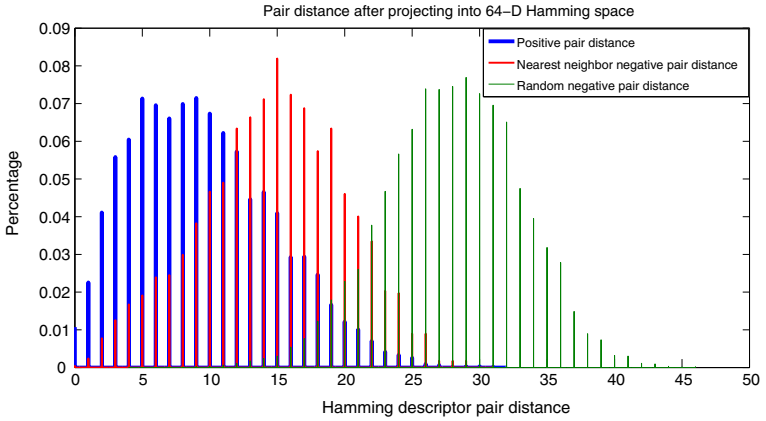


Fig. 2 The descriptor pair distance after projecting into a Hamming space

pairs distance distribution. Compared with original SIFT descriptor pairs distance distribution in Fig. 1, the positive Hamming descriptor pairs distance distribution gets denser and the nearest neighbor negative descriptor pairs distance and the random negative descriptor pairs distance are getting sparser in Fig. 2. This effect is the same as lower dimensional projected descriptors.

4 Image-based localization

Image-based localization is used to provide users the navigation information according to the query image provided by them. Just using a mobile phone, the users can take a photo of their surroundings and transmit it to the image-based localization system. The user receives the feedback with the navigation information from the image-based localization system, as shown in Fig. 3.

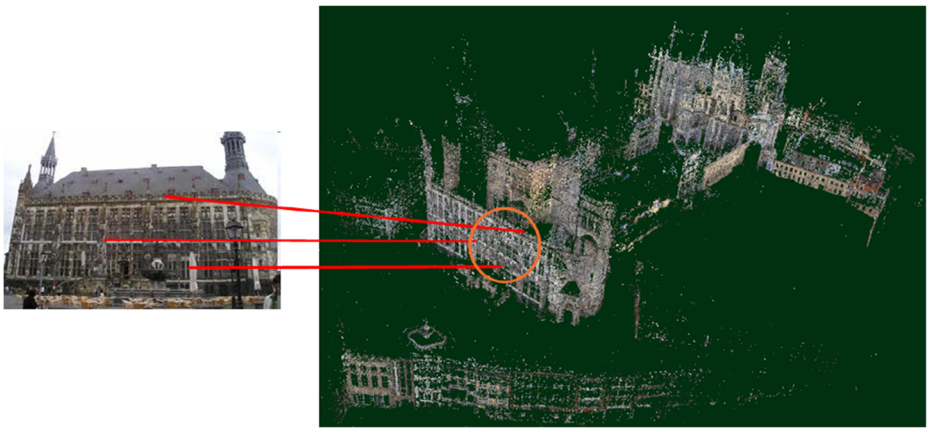


Fig. 3 2D-to-3D image matching

Originally, the image-based localization method is based on searching for the best candidate within an image database. The response image contains the most correspondences to the query image. To achieve higher accuracy, a 3D model obtained by reconstruction is utilized, which allows for orientation estimation. Compared with descriptors extracted from image database, descriptors from a 3D reconstruction model are much denser.

Our image-based localization method uses the direct 2D-to-3D matching [38]. The basic idea is to find the correspondences between 2D features and 3D points. These correspondences are built by searching for the 2D descriptor's nearest neighbors from all the descriptors in a 3D space.

We use a kd-tree based approach to find the approximate nearest neighbor descriptors, which is supported by FLANN library [32]. Each 3D point is represented by the mean value of all descriptors belonging to this point. The 2D-to-3D correspondence is accepted if it passes the SIFT ratio test. When more than one 2D feature matches the same 3D point, the 2D feature with the smallest distance in Euclidean space will be accepted as the matching feature. Finally, only the images consisting of at least 12 inliers will be registered. A registered image means that the image is correctly matched to the 3D model and the camera pose estimation based on the matching points is successful. The inliers are found by the Random Sample Consensus (RANSAC) algorithm [10]. A 6-point-direct-linear-transformation (6-point DLT) [16] is used in the RANSAC loop to estimate the camera 3D pose.

As suggested in [1], using the Hellinger distance instead of Euclidean distance in histogram local features (like SIFT) can largely increase the image retrieval performance. For this reason, we use (18) to compute the descriptor distance.

$$D = \sum_{i=1}^{N} \sqrt{x_i x_i'} \tag{18}$$

where D is the final distance between two descriptors. x_i and x_i' are the corresponding entries of two N dimensional descriptors.

This is the localization framework for our learned low dimensional projected descriptor in L1 normalized space. For the new learned Hamming descriptors, we also need to adjust the localization pipeline according to the Hamming descriptor properties. Different from SIFT descriptors, for Hamming descriptors, if the value of the corresponding dimension in two Hamming descriptors coincides, we consider the distance for this dimension of the two descriptors as 0. Otherwise, the distance of the two descriptors' corresponding dimension is 1. The final distance of the two descriptors is the sum of each dimension's distance. The distance computation in Hamming space is shown in (19).

$$D_H = \sum XOR(desc1, desc2) \tag{19}$$

where $desc1$ and $desc2$ are the two descriptors whose distance needs to be computed. XOR is the exclusive disjunction operation.

In the Hellinger space, a correspondence between a 2D feature and a 3D point (2df, 3dp) is accepted if the distance of the 2d feature's two nearest neighbor descriptors meets the following equation:

$$H(2df, 3dp_1)/H(2df, 3dp_2) < 0.8 \tag{20}$$

In (20), $2df$ and $3dp$ denote the descriptors belonging to the 2d feature and the 3d point. $H(2df, 3dp)$ is the Hellinger distance between the two descriptors. Equation 20 represents the ratio threshold between the first nearest neighbor 3D point descriptor to the 2D feature query descriptor and the second nearest neighbor 3D point descriptor to the 2D feature

query descriptor. This threshold is set in the way similar to the SIFT Euclidean distance threshold. The threshold is generated from the experiments result. From our experiments, the thresholds between 0.64 (0.8*0.8) to 0.81 (0.9*0.9) provide the highest image registration number. Out of this scope, the image registration number is decreased. If we set the threshold too low, there will be many correct feature correspondences rejected. If we set the threshold too high, the number of false positive correspondences will be increased. Both of these two kinds of settings result in lower image registration rate. Section 5 contains the experiment results of the image registration number based on different thresholds.

In Hamming space, the two nearest neighbor descriptors are quite likely to have the same distance to the query descriptor. Thus, we cannot set a ratio to reject the 2D-to-3D correspondences. Instead, in order to reject the descriptors that are less likely to be the true positive descriptor pairs, we set a maximum distance to accept the possible correspondences. If two Hamming descriptors' distance is smaller than the maximum distance, the descriptor will be accepted as a matching candidate. Otherwise, the descriptor will be rejected directly. We will evaluate the maximum distance threshold's effect on the number of images registered in the testing part.

Usually, for a query descriptor in the Hamming space, there will be several points' descriptors having the distance smaller than the maximum distance threshold. Thus, the first descriptor does not guarantee that the corresponding point is most likely to be the correct point. In dealing with this problem, we take a majority vote in this process. We set the nearest neighbor number as n. For these n nearest neighbors, we check if the descriptors exceed a maximum distance threshold. For all the nearest neighbor descriptors with smaller distance than the maximum distance threshold, we count each 3D point's descriptor number. The point with the most descriptors passing the maximum distance test will be selected as the matching point. Usually, assigning n the value of 10 can achieve the best result in our test. If there are two points having the same number of descriptors, we will reject these two points, since it does not provide a trusted choice. Meanwhile, this query will be ignored. We refer the number of descriptors belonging to one point as Nbp. The majority vote is further improved by checking the ratio of the second largest Nbp divided by the largest Nbp as in (21):

$$Nbp\_second/Nbp\_most < 0.8 \tag{21}$$

where Nbp_second is the second largest number that the descriptors associated to the same 3D point, Nbp_most is the largest number of the descriptors from the same 3D point. We set the ratio threshold as 0.8. If this equation holds, we accept the point with the most descriptors. Otherwise, we will reject this point. This threshold is to restrict the ratio between the second largest descriptor number of a 3D point and the largest number of the descriptors from the same 3D point. This threshold is set up from the experiments' results. Generally, the ratio from 0.6 to 0.9 can provide good image registration number result. If the threshold is set too low, the 3D point certainly has high confidence to be the correct corresponding 3D point of the 2D query descriptor. However, this would potentiality reject many other correct corresponding 3D points, as the largest descriptor number of the correct 3D point may not be so much higher than the second largest descriptor number of another 3D point. If we set the threshold too high, like more than 0.9, the largest descriptor number of the 3D point is not distinct enough to separate from the rest of the 3D points, resulting many false positive correspondences. In the extreme case, when the largest and the second largest descriptor number associated to two 3D points are the same, we disregard the 3D points instead of randomly selecting one of the 3D points with largest descriptor number. We further give the exepriment results for different thresholds in Section 5.

5 Experimental results

To evaluate the performance of our new proposed method for image-based localization, we present experiments using the new learned projected Hamming descriptors and the projected real-value descriptors on two challenging datasets, Dubrovnik [27] and Vienna [19]. Dubrovnik is a large dataset, reconstructed by using the photos from Flickr. Some images are removed from the reconstruction together with their descriptors and 3D points that can be seen in only one camera. The removed images are used for query images. Vienna dataset is reconstructed by the landmark images taken by a single calibrated camera. The query images of Vienna have a maximum dimension of 1,600 pixels in both width and height. The 266 query images for Vienna dataset are selected from the Panoramio website. The datasets are representatives of different scenarios. The Vienna dataset images are from uniform intervals of urban scenes. Dubrovnik depicts large clustered sets of views usually found on Internet photo collection websites. Detailed information can be found in Table 1.

5.1 Experiments on Hamming descriptor

We first test our new learned Hamming descriptors on the localization pipeline. Dimensionality plays a critical role in the descriptor's size. Since our purpose is to largely reduce the memory consumption for storing all the descriptors, the performance of different dimensional descriptors is essential to be tested. We give the number of images registered on the use of Hamming descriptors from 64 dimensions to 128 dimensions in Table 2. The registered image number shown in Table 2 tells us the number of 2D images that correctly matched to the 3D reconstruction model.

Note that there is a big boost for the number of the registered images from 64 dimensions to 96 dimensions. However, from 96 dimensions to 128 dimensions, the improvement is little. For the trade-off between memory consumption and accuracy, the 96 dimensional Hamming descriptor has the best performance. For 96 dimension Hamming descriptors, the memory is just 9.3 $\%(1/8 * 96/128)$ of original SIFT descriptor's memory consumption. From the image registration number perspective, the 128 dimensional Hamming descriptor has the best performance. However, the improvement of image registration number from 96 dimensional Hamming descriptor to 128 dimensional Hamming descriptor is not significant, while it consumes another 32 descriptor dimensions memory cost. From 64 to 96 dimensions, with the same memory consumption increase from 96 to 128 dimensions, the image registration of Hamming descriptors has a considerable improvement. To make a trade-off between the image registration number and the memory cost, we select 96 dimensional descriptors.

We conduct a majority vote during the selection of the 3D points based on Hamming descriptors. We set a threshold to select the 3D point containing the most nearest neighbor descriptors to 2D query descriptor. This ratio threshold is to restrict the ratio between the

Table 1 The datasets used for evaluation

Dataset	Number of 3D points	Number of descriptors	Size(MB)	Number of query images
Dubrovnik	1,886,884	9,606,317	1419	800
Vienna	1,123,028	4,854,056	702	266

Size describes the binary .info file size with all descriptors and 3D points information

Table 2 Performance of different dimensional Hamming descriptors for Vienna and Dubrovnik datasets

Dimensionality of Hamming descriptor	Number of images registered (Vienna Dataset)	Number of images registered (Dubrovnik Dataset)
64	139	573
96	161	663
128	166	672

second largest descriptor number of a 3D point and the largest number of the descriptors from another 3D point. We provide the experiment results of the different thresholds in Table 3.

From the experiments, the thresholds of 0.7 and 0.8 provide the highest image registration number. When the threshold is too low, many correct descriptor correspondences are rejected. On the contrary, many false positive descrioptors are accepted as correspondences when the threshold is set too high.

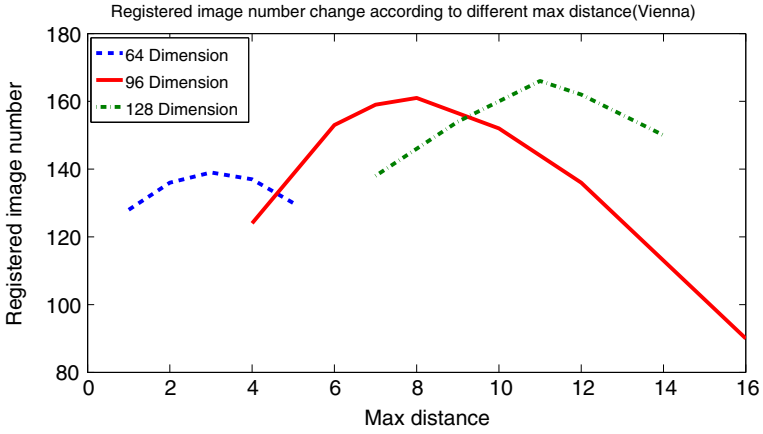
In the last section, we talked about setting a maximum distance threshold to select the matching point candidate. For different dimensional Hamming descriptors, the best maximum distance thresholds also differ. Figure 4 illustrates the trend for the number of registered image for each maximum distance threshold. Usually, the maximum distance 2 or 3 for 64 dimensions, 7 or 8 for 96 dimensions and 10 or 11 for 128 dimensions give the best result.

To achieve Hamming descriptors, we need to use a threshold to map the real-value descriptors into Hamming descriptors. As the same point's descriptors are quite likely to be clustered within each visual word, we conduct two experiments for searching within the visual words. The first is using the global threshold on learning Hamming descriptors and searching within each visual word. Another experiment is to learn the individual threshold for each visual word and quantize the descriptors into the Hamming space with the use of local thresholds.

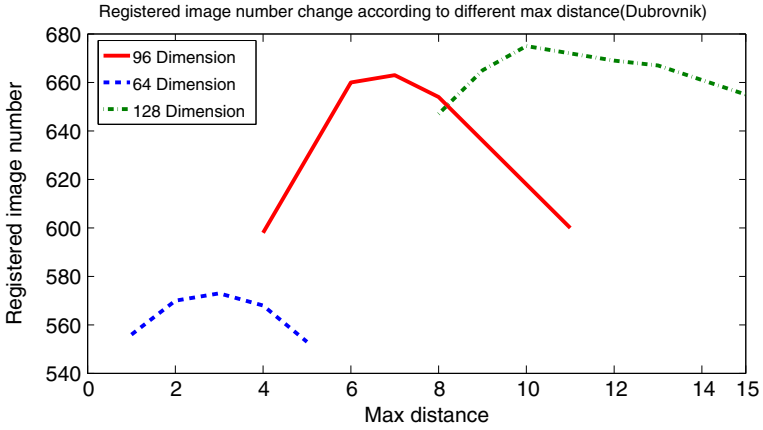
By searching through the visual words, all the 3D descriptors are first clustered into 1000k visual words. Each visual word is represented by the mean value of all the 3D descriptors within the visual word. Each 2D descriptor in an image is assigned to a visual word

Table 3 Performance of different thresholds between the second largest descriptor number of a 3D point and the largest number of the descriptors from another 3D point based on 96 dimensional Hamming descriptors for Vienna and Dubrovnik datasets

Threshold value	Number of images registered using 96 dimensional Hamming descriptor (Vienna Dataset)	Number of images registered using 96 dimensional Hamming descriptor (Dubrovnik Dataset)
0.4	138	582
0.5	151	633
0.6	159	657
0.7	161	662
0.8	161	663
0.9	157	650
1.0	151	628



(a) The new pair distance after projection in Hamming space.



(b) The new pair distance after projection in Hamming space.

Fig. 4 The number of registered images according to each max distance threshold for Vienna(*above*) and Dubrovnik(*below*) datasets, where X-axis is the max distance while Y-axis represents the registered image number

decided by searching the nearest distance to the visual words. Furthermore, we project both of the 2D and 3D descriptors into a lower dimensional Euclidean space. We further quantize the lower dimensional descriptors into the Hamming space by both the global threshold and the local threshold. The global threshold is learned from the whole 3D descriptor set, and the local threshold is learned from each visual word. We present the image registration performance using the global threshold and the local threshold in Table 4.

From Table 4, we can see that using the local threshold can improve the image registration rate. Also based on the different maximum distance, the result is also different. Based on the maximum distance of 8, the image registration performance is also largely improved compared with the maximum distance of 5. As we use the same kind of hashing method, the image registration rate using the Hamming descriptors learned from both lower dimensional descriptors by NNLDAHash and TRRF are basically the same.

Table 4 Registered image number using the global and local thresholds for Vienna dataset's 96 dimension Hamming descriptor with different maximum distance

Dataset	Number of registered images (Global threshold)	Number of registered images (Local threshold)
Vienna 96D with max distance 5	134	153
Vienna 96D with max distance 6	140	157
Vienna 96D with max distance 7	143	160
Vienna 96D with max distance 8	144	161
Vienna 96D with max distance 9	142	159
Vienna 96D with max distance 10	140	156
Vienna 96D with max distance 11	136	153

5.2 Experiments on low dimensional projected descriptor

Before mapping descriptors into the Hamming space, we also explore the low dimensional projected descriptors in Euclidean space. This projected descriptor contains low dimensionality, which could be used for localization tasks. For using the low dimensional projected descriptor, we need to pre-process the value after projection. In original SIFT descriptors, the value for each dimension is limited to $[0, 255]$. However, after projection, this range will become several times larger and the value for each dimension will contain decimal component. As a result, the projected value is not an integer anymore. For storing this projected value, we have to use the float or double data format instead of integer or character data. This potentially increases the memory requirement for the system. In dealing with this problem, we scale the projected value into the range of $[-128, 127]$ followed by rounding the decimal component into an integer. In this way, each dimension's value can be stored using the character data type. The number of registered images using the low dimensional projected descriptor is given in Table 5.

From Table 5, we see that the TRRF 64 dimensional descriptor achieves a higher image registration number compared with the other 64 dimensional descriptors. However, for NNLDAHash, even in projecting 128 dimensions to 32 dimensions, the descriptor still maintains a relatively high registered image number. The 32 dimensional projected descriptor consumes only 25 % memory compared with the original SIFT descriptor. Comparing TRRF with NNLDAHsh method, TRRF achieves lower registration performance using a low dimensional descriptor, but the performance will increase dramatically when the dimensionality is increased. NNLDAHash keeps the relatively high registration performance in a certain range.

Table 5 Performance of different dimensional projected descriptors for Vienna and Dubrovnik datasets

Dimensionality of projected descriptor	Number of images registered (Vienna Dataset)	Number of images registered (Dubrovnik Dataset)
32 (NNLDAHash)	185	737
64 (NNLDAHash)	195	771
32 (TRRF)	166	710
64 (TRRF)	204	780
SIFT (our pipeline)	214	785

We also set a ratio threshold between the first nearest neighbor 3D point descriptor to the 2D feature query descriptor and the second nearest neighbor 3D point descriptor to the 2D feature query descriptor when using Hellinger distance, which select the corresponding 3D point when the descriptor distance meeting the threshold constraint. The experiment results of different thresholds based on Hellinger distance are shown in Table 6.

From the experiments, thresholds from 0.6 to 0.9 can generally provide good results. The threshold too low or too high adds false negative or false positive correspondences in registering an image.

As the most time consuming parts in image registration are searching correspondences and RANSAC, the time performance using the new pipeline and the projected descriptors do not differ a lot (the descriptor projection time is made up by the speed-up part of the reduced dimensionality). Basically, registering an image costs about 0.3 seconds and rejecting an image costs around 0.9 to 1.2 seconds. We also compare our new learned descriptors and the adjusted localization pipeline with the state-of-the-art methods, shown in Table 7.

From Table 7, our 96 dimensional Hamming descriptor using 9.3 % memory can achieve similar result to the vocabulary tree based method [27]. The 128 dimensional Hamming descriptor can achieve higher image registration number than both methods in [27] and [19] with 12.5 % of the original memory consumption. The 32 dimensional projected descriptors use 25 % memory to obtain 90 % of the image registration number of the best performance method; our 64 dimensional projected descriptors outperform most of the state-of-the-art methods. In our new pipeline, using the original SIFT feature can achieve the highest image registration number.

5.3 Discussion

Our experiment results clearly show that the proposed method using the new learned low-dimensional features (either real-value or binary) have consistently produced high qualified matching accuracy on both Vienna and Dubrovnik data. Both new learned descriptors on the adjusted localization framework can achieve similar accuracy as the state-of-the-art methods. In this study, we have also investigated the dependency of image-based localization accuracy on the different number of Hamming descriptor dimensions and the corresponding maximum distance values. In general, the higher the Hamming descriptor dimension used, the better the matching accuracy. Although higher dimensionality results in higher

Table 6 Performance of different thresholds between the first nearest neighbor 3D point descriptor to the 2D feature query descriptor and the second nearest neighbor 3D point descriptor to the 2D feature query descriptor for Vienna and Dubrovnik datasets based on Hellinger distance using original SIFT feature

Threshold value	Number of images registered using 96 dimensional Hamming descriptor (Vienna Dataset)	Number of images registered using 96 dimensional Hamming descriptor (Dubrovnik Dataset)
0.4	189	712
0.5	207	756
0.6	212	778
0.7	214	785
0.8	214	785
0.9	210	770
1.0	203	737

Table 7 Comparison between our method and the different state-of-the-art methods

Method	Number of images registered (Dubrovnik)	Number of images registered (Vienna)
P2F [27] (128D)	753	204
Voc.tree(all) [27] (128D)	668	–
Fast Direct 2D-to-3D [38] (128D)	781	205
Voc. tree GPU [19] (128D)	–	165
Hamming descriptor (96D)	663	161
Hamming descriptor (128D)	672	166
Projected descriptor (LDAHash 64D)	771	195
Projected descriptor (TRRF 64D)	780	202
Projected descriptor (LDAHash 32D)	737	185
Projected descriptor (Classic LDA 32D)	704	160
Projected descriptor (Classic PCA 32D)	710	156
SIFT (our pipeline)(128D)	785	214

The red color is for comparing Hamming descriptor and the blue color is for comparing the projected real-value descriptor

accuracy, the improvement in accuracy also slows with the higher dimensionality. After a certain point, the gain in accuracy is not worth the drawbacks of the higher dimensionality, such as higher memory usage and computation time. For instance, as shown in Table 2, the obtained registered image number is 161 and 663 on Vienna and Dubrovnik data using 96-D Hamming descriptors respectively, which is very similar to the results obtained by using 128-D Hamming descriptors. So depending on the task and the memory condition, choosing a best trade-off of the dimensionality is critical. The performance of Fast Direct 2D-to-3D is very similar to the method proposed in this paper. However, the memory cost of our method is much smaller than the Fast Direct 2D-to-3D method. This is a substantial improvement. Meanwhile, the learning procedure of the projection matrix and the threshold can be conducted in off line time, which does not add the time cost during the localization process. The time for projecting the features can be compensated by the reduced time of descriptor distance calculation.

6 Conclusion

In this paper, we learn the Hamming descriptor and the low dimensional projected real-value descriptor for the use of image-based localization. The current state-of-the-art image-based localization frameworks use high dimensional descriptors in Euclidean space to provide accurate localization result. The use of high dimensional descriptor in Euclidean space causes high memory usage for the image-based localization task.

Based on this limitation, we learn a lower dimensional Hamming descriptor for the image-based localization task. We tested different methods for improving the image registration performance in the localization pipeline. Besides the Hamming descriptor, we also learned the lower dimensional projected descriptors for matching in Euclidean space. Using each localization method, we did extensive tests on two large datasets.

We adjust the projection matrix learning process, making the learned projection matrix better separate the positive and negative descriptor pairs. Meanwhile, the localization

pipeline is also adjusted to achieve higher image registration number. Using the original SIFT descriptor in the new pipeline could achieve higher accuracy than the original pipeline. Our learned Hamming descriptor performs slightly worse than the state-of-the-art methods. However, the memory we use is less than 10 percentage of their methods (considering 96 dimensional Hamming descriptor). Our method shows considerable potential for the utilization in real world applications. The use of the lower dimensional projected descriptor can outperform most state-of-the-art localization methods, with only one quarter dimensions of their local descriptors.

Future work includes learning better binary descriptors for preserving the original descriptors' properties, which could further improve our descriptors in the use of image-based localization.

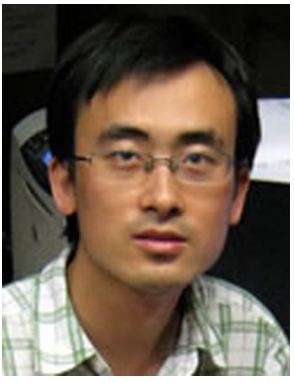
Acknowledgements This work has been financially supported by European Master in Informatics program, RWTH Aachen University, University of Trento and the PhD program of University of Delaware. The authors are grateful to Torsten Sattler and Leif Kobbelt from RWTH Aachen University for their great help to make this work accomplished.

References

1. Arandjelovic R, Zisserman A (2012) Three things everyone should know to improve object retrieval. In: Proceedings of 2012 IEEE conference on computer vision and pattern recognition (CVPR). pp 2911–2918
2. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). J Comput Vis Image Underst (CVIU) 110(3):346–359
3. Beltran A, Abargues C, Granell C, Núñez M, Díaz L, Huerta J (2013) A virtual globe tool for searching and visualizing geo-referenced media resources in social networks. Multimed Tools Appl (JMTA):1–25
4. Broder A (1997) On the resemblance and containment of documents. In: Proceedings of compression and complexity of sequences. pp 21–29
5. Broder A, Charikar M, Frieze A, Mitzenmacher M (1998) Min-wise independent permutations. J Comput Syst Sci 60:327–336
6. Brown M, Hua G, Winder S (2011) Discriminative learning of local image descriptors. IEEE Trans Patt Anal Mach Intell (TPAMI) 33(1):43–57
7. Castle R, Klein G, Murray D (2008) Video-rate localization in multiple maps for wearable augmented reality. In: Proceedings of the 2008 12th IEEE international symposium on wearable computers (ISWC). pp 15–22
8. Crandall D, Owens A, Snavely N, Huttenlocher D (2011) Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of the 2011 IEEE conference on computer vision and pattern recognition (CVPR). pp. 3001–3008
9. Cummins M, Newman P (2008) Fab-map: probabilistic localization and mapping in the space of appearance. Int J Robot Res(IJRR) 27(6):647–665
10. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
11. Frahm J, Georgel P, Gallup D, Johnson T, Raguram R, Wu C, Jen Y, Dunn E, Clipp B, Lazebnik S, Pollefeys M (2010) Building Rome on a cloudless day. In: Proceedings of the 11th European conference on computer vision (ECCV). pp 368–381
12. Gao Y, Wang M, Zha Z, Shen J, Li X, Wu X (2013) Visual-textual joint relevance learning for tag-based social image search. IEEE Trans Image Process (TIP) 22(1):363–376
13. Gao Y, Wang M, Zha Z, Tian Q, Dai Q, Zhang N (2011) Less is more: efficient 3-d object retrieval with query view selection. IEEE Trans Multimed (TMM) 13(5):1007–1018
14. Han Y, Wu F, Tao D, Shao J, Zhuang Y, Jiang J (2012) Sparse unsupervised dimensionality reduction for multiple view data. IEEE Trans Circ Syst Video Tech 22(10):1485–1496
15. Han Y, Yang Y, Zhou X (2013) Co-regularized ensemble for feature selection. In: Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI)
16. Hartley R, Zisserman A (2004) Multiple view geometry in computer vision. Cambridge University Press. ISBN: 0521540518

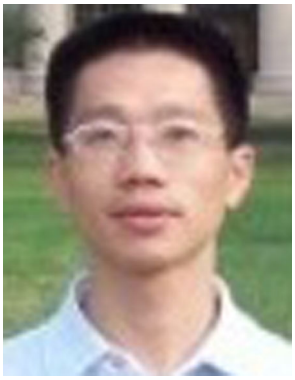
17. Heath K, Gelfand N, Ovsjanikov M, Aanjaneya M, Guibas L (2010) Image webs: computing and exploiting connectivity in image collections. In: Proceedings of the 2010 IEEE conference on computer vision and pattern recognition (CVPR). pp 3432–3439
18. Hua G, Brown M, Winder S (2007) Discriminant embedding for local image descriptors. In: Proceedings of the 2007 IEEE 11th international conference on computer vision (ICCV). pp 1–8
19. Irschara A, Zach C, Frahm J, Bischof H (2009) From structure-from-motion point clouds to fast location recognition. In: Proceedings of the 2009 IEEE computer society conference on computer vision and pattern recognition (CVPR). pp 2599–2606
20. Jacobs N, Miskell K, Pless R (2011) Webcam geo-localization using aggregate light levels. In: Proceedings of 2011 IEEE workshops on applications of computer vision (WACV). pp 132–138
21. Jolliffe I (1986) Principal component analysis. Springer Verlag
22. Kalia R, Lee KD, Samir B, Je SK, Oh WG (2011) An analysis of the effect of different image preprocessing techniques on the performance of surf: speeded up robust features. In: Proceedings of the 2011 17th Korea-Japan joint workshop on frontiers of computer vision. pp 1–6
23. Ke Y, Sukthankar R (2004) Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 2. pp 506–513
24. Kulis B, Darrell T (2009) Learning to hash with binary reconstructive embeddings. In: Proceedings of the 23nd annual conference on neural information processing systems (NIPS). pp 1042–1050
25. Kulis B, Grauman K (2009) Kernelized locality-sensitive hashing for scalable image search. In: Proceedings of the 2009 IEEE 12th international conference on computer vision (ICCV). pp 2130–2137
26. Leonard J, Durrant-Whyte H (1991) Simultaneous map building and localization for an autonomous mobile robot. In: Proceedings of the 1991 IEEE/RSJ international workshop on intelligent robots and systems '91. 'Intelligence for mechanical systems, vol 3. pp 1442–1447
27. Li Y, Snavely N, Huttenlocher DP (2010) Location recognition using prioritized feature matching. In: Proceedings of the 11th European conference on computer vision (ECCV). pp 791–804
28. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis (IJCV) 60(2):91–110
29. Ma Z, Yang Y, Cai Y, Sebe N, Hauptmann A (2012) Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In: Proceedings of the 20th ACM international conference on multimedia (MM). pp 469–478
30. Ma Z, Yang Y, Sebe N, Hauptmann A (2014) Knowledge adaptation with partially shared features for event detection using few exemplars. In: IEEE transactions on pattern analysis and machine intelligence. doi:10.1109/TPAMI.2014.2306419
31. Mika S, Ratsch G, Weston J, Scholkopf B, Mullers K (1999) Fisher discriminant analysis with kernels. In: Proceedings of the 1999 IEEE signal processing society workshop neural networks for signal processing IX. pp 41–48
32. Muja M, Lowe D (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: Proceedings of the 2009 international conference on computer vision theory and applications (VISAPP). pp 331–340
33. Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 2. pp 2161–2168
34. Philbin J, Isard M, Sivic J, Zisserman A (2010) Descriptor learning for efficient retrieval. In: Proceedings of the 11th European conference on computer vision conference on Computer vision (ECCV). pp 677–691
35. Powell M (1964) An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput J 7:155–162
36. Raginsky M, Lazebnik S (2009) Locality-sensitive binary codes from shift-invariant kernels. In: Proceedings of the 22nd annual conference on neural information processing systems (NIPS). pp 1509–1517
37. Robertson D, Cipolla R (2004) An image-based system for urban navigation. In: Proceedings of the 2004 British machine vision conference (BMVC). pp 819–828
38. Sattler T, Leibe B, Kobbelt L (2011) Fast image-based localization using direct 2d-to-3d matching. In: Proceedings of the 2011 IEEE international conference on computer vision (ICCV). pp 667–674
39. Schindler G, Brown M, Szeliski R (2007) City-scale location recognition. In: Proceedings of the 2007 IEEE conference on computer vision and pattern recognition (CVPR). pp 1–7
40. Shao H, Svoboda T, Tuytelaars T, Van Gool L (2003) Hpat indexing for fast object/scene recognition based on local appearance. In: Proceedings of the 2003 international conference on image and video retrieval (CIVR). pp 71–80

41. Smith R, Cheeseman P (1986) On the representation and estimation of spatial uncertainty. Int J Robot Res (IJRR) 5(6):56–68
42. Snavely N, Seitz S, Szeliski R (2006) Photo tourism: exploring photo collections in 3d. ACM Transit Graph 25(3):835–846
43. Steinhoff U, Dusan O, Perko R, Schiele B, Leonardis A (2007) How computer vision can help in outdoor positioning. In: Proceedings of the 2007 European conference on ambient intelligence (AmI). pp 124–141
44. Strecha C, Bronstein A, Bronstein M, Fua P (2012) LDAHash: improved matching with smaller descriptors. IEEE Trans Patt Anal Mach Intell (TPAMI) 34:66–78
45. Tola E, Lepetit V, Fua P (2010) Daisy: an efficient dense descriptor applied to wide-baseline stereo. IEEE Trans Patt Anal Mach Intell (TPAMI) 32(5):815–830
46. Wang H, Yan S, Xu D, Tang X, Huang T (2007) Trace ratio vs. ratio trace for dimensionality reduction. In: IEEE conference on computer vision and pattern recognition (CVPR). pp 1–8
47. Wang M, Gao Y, Lu K, Rui Y (2013) View-based discriminative probabilistic modeling for 3d object retrieval and recognition. IEEE Trans Image Process (TIP) 22(4):1395–1407
48. Weiss Y, Torralba A, Fergus R (2008) Spectral hashing. In: Proceedings of the 22nd annual conference on neural information processing systems (NIPS). pp 1753–1760
49. Wendel A, Irschara A, Bischof H (2011) Natural landmark-based monocular localization for mavs. In: Proceedings of the 2011 IEEE international conference on robotics and automation (ICRA). pp 5792–5799
50. Winder S, Hua G, Brown M (2009) Picking the best daisy. In: Proceedings of the 2009 IEEE conference on computer vision and pattern recognition (CVPR). pp 178–185
51. Xiao J, Chen J, Yeung D, Quan L (2008) Structuring visual words in 3d for arbitrary-view object localization. In: Proceedings of the 10th European conference on computer vision (ECCV). pp 725–737
52. Xuan K, Zhao G, Taniar D, Safar M, Srinivasan B (2011) Voronoi-based multi-level range search in mobile navigation. Multimed Tools Appl (JMTA) 53(2):459–479
53. Yagnik J, Strelow D, Ross DA, Lin RS (2011) The power of comparative reasoning. In: Proceedings of the 2011 IEEE international conference on computer vision (ICCV). pp 2431–2438
54. Yang Y, Nie F, Luo J, Zhuang Y, Pan Y (2012) A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. IEEE Trans Patt Anal Mach Intell (TPAMI) 34:723–742
55. Yang Y, Zhuang Y, Wu F, YH, P (2008) Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. IEEE Trans Multimed (TMM) 10:437–446
56. Yu S, Yang Y, Hauptmann A (2013) Harry potter's marauder's map: localizing and tracking multiple persons-of-interest by nonnegative discretization. In: Proceedings of 2013 IEEE conference on computer vision and pattern recognition (CVPR)
57. Zhang W, Kosecka J (2006) Image based localization in urban environments. In: Proceedings of the 3rd international symposium on 3D data processing, visualization, and transmission (3DPVT). pp 33–40



Guoyu Lu is currently a PhD student and Research Assistant in Video/Image Modeling and Synthesis Lab, University of Delaware. He was a student of European Master in Informatics (EuMI) double degree program. He obtained Master degree in Computer Science at University of Trento and Master degree in Media Informatics at RWTH Aachen University. He was a visiting scholar in Auckland University of Technology KEDRI group. His research interest includes Computer Vision, Multimedia Retrieval, and Machine Learning.

Nicu Sebe is with the Department of Information Engineering and Computer Science, University of Trento, Italy, where he is leading the research in the areas of multimedia information retrieval and human-computer interaction in computer vision applications. He was involved in the organization of the major conferences and workshops addressing the computer vision and human-centered aspects of multimedia information retrieval, among which as a General Co-Chair of the IEEE Automatic Face and Gesture Recognition Conference, FG 2008, ACM International Conference on image and Video Retrieval (CIVR) 2007 and 2010, and WIAMIS 2009 and as one of the initiators and a Program Co-Chair of the Human-Centered Multimedia track of the ACM Multimedia 2007 conference. He is the general chair of ACM Multimedia 2013 and was a program chair of ACM Multimedia 2011. He has served as the guest editor for several special issues in IEEE Computer, Computer Vision and Image Understanding, Image and Vision Computing, Multimedia Systems, and ACM TOMCCAP. He has been a visiting professor in Beckman Institute, University of Illinois at Urbana-Champaign and in the Electrical Engineering Department, Darmstadt University of Technology, Germany. He is the co-chair of the IEEE Computer Society Task Force on Human-centered Computing and is an associate editor of Machine Vision and Applications, Image and Vision Computing, Electronic Imaging and of Journal of Multimedia. He is a senior member of IEEE and of ACM.



Congfu Xu is an associate professor in the College of Computer Science, Zhejiang University. He finished his Master and PhD in the department of Computer Science & Technology, Zhejiang University. His research interest includes Information Fusion, Data Mining, Artificial Intelligence, Recommender Systems, and Sensor Networks.

Chandra Kambhamettu is currently a Professor in the Department of Computer Science, University of Delaware, Newark, where he leads the Video/Image Modeling and Synthesis (VIMS) group. From 1994-1996, he was a Research Scientist at the NASA Goddard Space Flight Center (GSFC). His research interests include video modeling and image analysis for biomedical, remote sensing, and multimedia applications. He is best known for his work in motion analysis of deformable bodies, for which he received the NSF CAREER award in 2000. He has published over 200 peer-reviewed papers, supervised ten Ph.D. students and several Masters students in his areas of interest. Dr. Kambhamettu received the Excellence in Research Award from NASA in 1995 while at GSFC. He has served as Area Chair, and has been technical committee member for leading computer vision and medical conferences. He has also served as Associate Editor for the journals Pattern Recognition and Pattern Recognition Letters and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.