Adapted SIFT Descriptor for Improved Near Duplicate Retrieval

Afra'a Ahmad Alyosef and Andreas Nürnberger
Department of Technical and Business Information Systems, Faculty of Computer Science, Otto von Geruicke University Magdeburg, Magdeburg, Germany

Keywords: Image Near Duplicate Retrieval, SIFT Descriptor, RC-SIFT 64*D*, Feature Extraction.

Abstract: The scale invariant feature transformation algorithm (SIFT) has been designed to detect and characterize local features in images. It is widely used to find similar regions in affine transformed images, to recognize similar objects or to retrieve near-duplicates of images. Due to the computational complexity of SIFT based matching operations several approaches have been proposed to speed up this process. However, most approaches lack significant decrease of matching accuracy compared to the original descriptor. We propose an approach that is optimized for near-duplicate image retrieval tasks by a dimensionality reduction process that differs from other methods by preserving the information around the keypoints of any region patches of the original descriptor. The computation of the proposed Region Compressed (RC) SIFT−64*D* descriptors is therefore faster and requires less memory for indexing. Most important, the obtained features show at the same time a better retrieval performance and seem to be even more robust. In order to prove this, we provide results of a comparative performance analysis using the original SIFT−128*D*, reduced SIFT versions, SURF−64*D* and the proposed RC-SIFT−64*D* in image near-duplicate retrieval using large scale image benchmark databases.

1 INTRODUCTION

Finding similar images that show the same scene, but have been taken with slightly different conditions (so-called "near duplicate images") is still a very challenging task, even though its a very fundamental problem in many real world tasks. In the literature (e.g., (Xu et al., 2010), (Chum et al., 2007)) one can meanwhile find several benchmark databases containing sets of near duplicates that can be used to study the performance of algorithms and similarity measures that have been designed for the task of finding near duplicate images. The sets of near duplicate images provided slightly differ by quite diverse factors as noise, blurring, compression rates, lighting conditions or camera viewpoint. Therefore, these collections nicely cover characteristics of real world collections and tasks, e.g., such as the daily business of media agencies to filter and sort huge amounts of images that have been taken by different photographers during fashion fairs or political events. The main underlying problems here are data redundancy, e.g., having two or more of copies of the same image in a collection; view configuration issues, i.e., detecting near duplicate images which are taken using different configuration of a camera or different cameras; problems of copyright infringement, e.g., the task of detecting manipulated (fabricated) images and problems of time offsets, i.e., finding similar images of the same scene taken minutes or hours apart.

The first step in image near-duplicate retrieval (NDR) is to extract the features of an image. The goal is to represent images by means of one or more kinds of their distinct characteristics. One of the most used approaches for finding the local features in the field of NDR (Auclair et al., 2006), (Chum et al., 2008), (Zhang and et al., 2004), (Chu et al., 2013) is the scale invariant feature transform algorithm (SIFT) (Lowe, 2004), because the SIFT features are invariant to scale and rotation variation and perform robustly even if the images differ in perspective, noise, and illumination (Lowe, 2004). Furthermore, approaches to index and quantize SIFT descriptors have been proposed to improve the performance of the matching process, especially in the image NDR tasks (see e.g., (Jiang et al., 2015), (Nistèr and Stewènius, 2006), (Auclair et al., 2006), (Chum et al., 2008)).

In this work, we propose a method to further improve the performance of SIFT by reducing the dimensionality of SIFT−128*D* descriptors to 64*D*. Thereby we tackle two issues: First, especially in image NDR field the reduced SIFT−64*D* speeds up the process of image matching by decreasing the memory and time complexity of the indexing and match-

Alyosef, A. and Nürnberger, A.
Adapted SIFT Descriptor for Improved Near Duplicate Retrieval.
DOI: 10.5220/0005694800550064
In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016)*, pages 55-64
ISBN: 978-989-758-173-1
Copyright © 2016 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

ing process. Second, the reduced SIFT improves the robustness and accuracy of the matching process. The performance of the proposed approach is tested by conducting extensive experiments using established benchmark datasets. The experiments show that the reduced SIFT$-64D$ improves the performance of SIFT in image NDR tasks.

The remainder of this paper is organized as follows. Section 2 gives an overview of prior work related with the SIFT algorithm and image NDR algorithms. Section 3 details the proposed method to reduce SIFT descriptors. Section 4 presents the settings of our experiments and Section 5 discusses the results of experiments. Finally, Section 6 draws conclusions of this work and discusses possible future work.

2 RELATED WORK

Due to the robustness of the SIFT descriptor against different kinds of image deformation, it has been widely used in image NDR (Auclair et al., 2006), (Chum et al., 2008), image classification (Nistèr and Stewènius, 2006) and diagnosis of tumors in medical images (Jiang et al., 2015).

In (Khan et al., 2011) the SIFT descriptor have been reduced to $96D$, $64D$ and $32D$ by ignoring the contribution of some regions around the keypoints when building the descriptors. It has been mentioned in the original SIFT$-128D$ algorithm that ignoring some values of SIFT descriptor lead to decreased matching performance. However, in (Khan et al., 2011) SIFT$-96D$, $64D$ have shown robust performance of image matching as the original SIFT$-128D$ across a substantial range of affine transformation and addition of noise. However, the performance of the original SIFT$-128D$ is still better than the performance of SIFT$-96D$, $64D$ descriptors in case of additional noise and illumination change (Khan et al., 2011). In (Ke and Sukthankar, 2004) principle component analysis is used to obtain the $64D$ SIFT descriptors. In (Jègou et al., 2010) the SIFT descriptors are aggregated to represent the images in form of vectors after that, principle component analysis is applied to jointly optimize the dimensionality reduction of these vectors. However, both approaches require a training stage for the specific image collection.

To optimize the use of SIFT features in the matching process, various techniques are suggested to structure, index and quantize the descriptors in suitable form for further processing. In (Lowe, 2004) the $kd-$tree has been used to structure the SIFT descriptor in $128D$ space and to speed up their matching process. However, the efficiency of $kd-$tree decreases for high dimensional data because of the request time for backtracking through the tree. In (Li et al., 2014), (Yang and Newsam, 2008), (Grauman and Darrell, 2005), (Grauman and Darrell, 2007), SIFT descriptors have been quantized by splitting them into k groups using the $k-$means clustering algorithm. In this case a specific number of clusters is determined and the descriptors are indexed by their closest centers. The produced cluster centers form a bag of words and the images are presented in form of vectors of these bag of words. In (Li et al., 2014), (Yang and Newsam, 2008), (Grauman and Darrell, 2005), (Grauman and Darrell, 2007) the concept of bag of words is used in further training steps to optimize the matching process. In (Jiang et al., 2015), (Nistèr and Stewènius, 2006) hierarchical $k-$means clustering is employed to build a vocabulary tree. Each leaf node of the tree is associated with an inverted file which contains the indexes of images that have at least one descriptor passed through a specific path in the tree. In our work, the concept of a vocabulary tree is applied to index the descriptors of several kinds. In the next subsection the details of compressing the SIFT descriptors are explained.

3 REGION COMPRESSED SIFT DESCRIPTOR FOR NDR

To motivate and describe our suggested modifications of the SIFT descriptor, we will firstly explain briefly the working mechanism of the SIFT detector and descriptor (Lowe, 2004).

3.1 SIFT$-128D$ Descriptor

The detection of SIFT features can be achieved in four main stages specified as follow: scale space extrema detection, keypoint localization, orientation computation and keypoint descriptor computation.

In the first stage image scale space is built by downsampling and blurring the input image several times. The blurring is achieved by convolving the input image with multiple-scale Gaussian filters. After that, the difference of neighbors in the scale space is computed to form difference of Gaussian (DoG) images. In the second stage, SIFT keypoints are determined by finding the local maxima and minima in DoG images. The stability of keypoints is verified against contrast change and edge response and the unstable keypoints are rejected. In the third stage, the dominant orientation is determined and assigned to each keypoint.

In the final stage a highly distinctive descriptor is computed at each keypoint. The SIFT descriptor is extracted from the region around the keypoint which is called region of interest (RoI). The RoI is rotated around a keypiont relative to the dominant orientation. Afterwards, a $n \times n$ orientation histogram is created over the RoI. For each bin in the histogram r orientations are assigned, so that the descriptor has three dimensions and $n \times n \times r$ element. The size of the SIFT descriptor is controlled by the width of the orientation histogram n and the number of orientation bins r. In the original SIFT algorithm (Lowe, 2004), it has been shown that the best matching results are reported when $n = 4$ and $r = 8$, i.e., when a descriptor of $4 \times 4 \times 8 = 128$ element is constructed. Fig. 1(a) presents the way in which the SIFT$-128D$ descriptor is constructed.

However, the high dimensionality of SIFT descriptor ($128D$) increases the sparsity of descriptors and this may affect the accuracy of descriptor indexing in image NDR (for a discussion of problems related to high dimensional data indexing and clustering, see e.g., (Steinbach et al., 2003)). Therefore, in this work we aim to compress the dimensionality of the SIFT descriptor. In the next subsection, we explain our approach to compress SIFT descriptor.

3.2 Region Compressed SIFT Descriptor

To increase the efficiency of descriptor indexing, i.e., speeding up the process and reducing the amount of stored data in NDR, we propose a method to compress the dimensionality of the SIFT descriptor from $128D$ to $64D$. We achieve this by extracting first the SIFT features in the same way as in the original SIFT algorithm (Lowe, 2004) (as described in Section 3.1). Afterwards, the descriptors are computed over all pixels in RoI with specific location, gradient and orientation with respect to the corresponding keypoint. The descriptor is computed in form of a $3D$ histogram centered at the keypoint. In the original SIFT algorithm, this descriptor has the dimensions $4 \times 4 \times 8$. The values of these three dimensions indicate how the keypoint can be shifted to each allowed position in RoI in vertical and horizontal locations that is 4×4 locations. For each location 8 directions are allowed between 0^o and 360^o. In contrast to the reduction method presented in (Khan et al., 2011) through ignoring some patches of RoIs, we suggest in this work that for each two possible horizontal shifting in the same direction with respect to the keypoint, only one vertical shifting is available so that, for the all possible horizontal shifting (i.e., four horizontal shifting) in all directions only two vertical shifting exists. For each of this (4×2) locations eight directions are assigned. In this way we reduce the amount of the possible change of the SIFT descriptor when the RoI is modified. Moreover, the number of altered bins in the RoI histogram decreases. As a result we obtain $4 \times 2 \times 8$ histogram i.e., $64D$ SIFT descriptor. We call our method for extracting and compressing SIFT descriptor "Region Compressed SIFT" (RC-SIFT). The histogram at each keypoint can be presented by a triplet of elements H_y, H_x and H_θ where:

$$H_y = y - \frac{N_y - 1}{2} \tag{1}$$

$$H_x = x - \frac{N_x - 1}{2} \tag{2}$$

$$H_\theta = \frac{2\pi}{N_\theta} \tag{3}$$

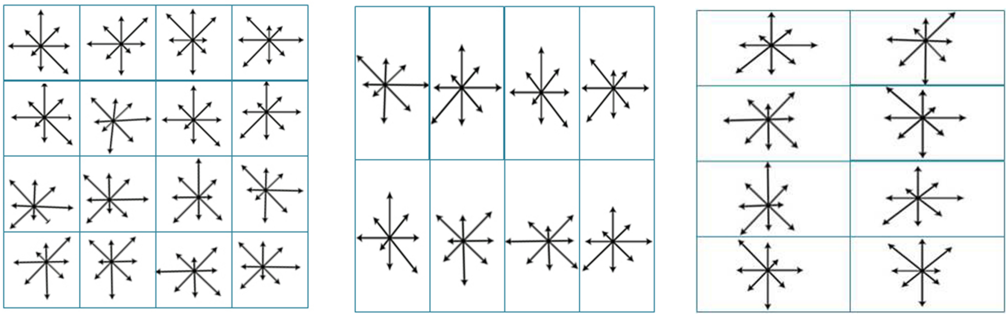
where N_y and N_x define the number of bins in H_y and H_x, respectively. The values of y and x are defined as $y = 0, \ldots, N_y - 1$, and $x = 0, \ldots, N_x - 1$. N_θ is the number of orientations in each bin of the histogram and θ is defined as $\theta = 0, \ldots, N_\theta - 1$.

In this work, we perform the experiments for $N_y = 2$, $N_x = 4$ and $N_\theta = 8$ to get the descriptor of the form $4 \times 2 \times 8$. We refer to this descriptor as RC-SIFT$-64D$(R) (see Figure 1(b)). Afterwards, the experiment is applied for $N_y = 4$, $N_x = 2$ and $N_\theta = 8$ to obtain the descriptor of the form $2 \times 4 \times 8$. we refer to this descriptor as RC-SIFT$-64D$(C) (see Figure 1(c)). After that, experiments are performed for $N_y = 2$, $N_x = 2$ and $N_\theta = 8$ to get RC-SIFT$-32D$ and finally for $N_y = 2$, $N_x = 2$ and $N_\theta = 4$ to get RC-SIFT$-16D$.

In this way the compressed SIFT descriptor preserves the size of RoI around a keypoint, i.e., contrary to the method suggested in (Khan et al., 2011), no region around the keypoint is ignored. In the next step the efficiency of RC-SIFT$-64D$, RC-SIFT$-32D$ and RC-SIFT$-16D$ are evaluated against the performance of the original SIFT$-128D$, SURF$-64D$ and SIFT$-64D$ suggested in (Khan et al., 2011) for image near-duplicate retrieval.

3.3 SIFT Descriptors Indexing with a Vocabulary Tree

The straightforward way to match SIFT features is exhaustive search, which can be achieved by matching each feature of a given query image with all features in the feature database. However, the exhaustive search of SIFT features is extremely time consuming especially for large scale image databases



(a) 4×4 computed orientation histogram array in SIFT$-128D$.

(b) 2×4 computed orientation histogram array in RC-SIFT$-64D$(R).

(c) 4×2 computed orientation histogram array in RC-SIFT$-64D$(C).

Figure 1: Comparison between SIFT$-128D$ and RC-SIFT$-64D$ descriptors. We refer to the compression of forms $4\times2\times8$ and $2\times4\times8$ as RC-SIFT$-$64(R)and RC-SIFT$-$64(C), respectively. These symbols are used in the all presented tables.

which produce a huge amount of features. To overcome this problem, hashing functions (Auclair et al., 2006), (Chum et al., 2008), direct clustering (Chum et al., 2007), (Zhang et al., 2013) and hierarchical clustering (Jiang et al., 2015), (Nistèr and Stewènius, 2006) have been adapted to quantize and index SIFT descriptors. In our study, a vocabulary tree and inverted files are used as described in (Jiang et al., 2015), (Nistèr and Stewènius, 2006) to index SIFT descriptors. The vocabulary tree is built by applying the $k-$means algorithm on the entire descriptor database which split them into k clusters where each cluster consists of a set of descriptors closest to a particular center. This process is applied recursively on each cluster to build a vocabulary tree of depth L and k^L leaf nodes. The tree nodes present cluster centers and are referred to as "visual word" (Jiang et al., 2015). The leaf nodes in the tree are represented by inverted files. Each inverted file contains the indexes of the images that they represent with at least one descriptor at a particular leaf node. The inverted files of the leaf nodes are concatenated to get the inverted files of inner and root nodes. These inverted files strongly speed up the matching process. Moreover, the inverted files help to adapt weights for the branches of the tree.

In (Nistèr and Stewènius, 2006) the $L1-$norm and $L2-$norm have been used to compute the similarity between images. The $L1-$norm tends to give better matching results (Nistèr and Stewènius, 2006). In our evaluation, we use both the $L1-$norm and $L2-$norm in order to compute the similarity between normalized query and database vectors by traversing each vector in a vocabulary tree as described in Eq. 4 (Nistèr and Stewènius, 2006).

3.4 Complexity of Vocabulary Tree for SIFT$-64D$ and SIFT$-128D$

After building the vocabulary tree (see Section 3.3), the tree is used for image matching. The complexity of this process is computed assuming that descriptors are represented as character data-type. A D dimensional descriptor vocabulary tree of depth L and k^L leaf nodes need a memory of $O(Dk^L)$. Specifically, a $128D$ descriptor tree requires $O(128k^L)$ whereas, a $64D$ descriptor tree requires only $O(64k^L)$. Moreover, the time complexity of building a vocabulary tree is affected by the dimensionality of descriptors. Considering that total number of nodes in the vocabulary tree is given as $\Sigma_L^{i=1} k^i = \frac{k^{L+1}-k}{k-1} \approx k^L$ (see also (Nistèr and Stewènius, 2006)), the time complexity of the vocabulary tree for a $D-$ dimensional descriptor database is given as $O(DNTk^L)$, where k is the number of initial clusters, T is the iteration of algorithm and N is the number of all descriptors of a given image database. Based on this, the time complexity of building a tree for a descriptor database of dimensionality $D = 128$ is $O(128NTk^L)$ whereas the time complexity for the RC-SIFT$-64D$ descriptors is $O(64NTk^L)$. So it is just a linear decrease but the suggested RC-SIFT$-64D$ descriptors obviously speeds up the indexing process and reduce the required memory for processing. Table 1 presents that the indexing time needed by RC-SIFT$-64D$ and SIFT$-64D$ (Khan et al., 2011) is about the halve time needed by SIFT$-128D$. The presented results are computed using a vocabulary tree of depth $L = 4$ and initial centers $k = 10$.

In the next section, the performance of the suggested RC-SIFT ($64D$, $32D$ and $16D$) descriptors are evaluated against the performance of the original SIFT$-128D$, SIFT$-64D$ and SURF$-64D$ based on typical near-duplicate retrieval tasks.

Table 1: The computation time needed to perform the indexing for both SIFT$-128D$, RC-SIFT$-64D$ and SIFT-$64D$ (Khan et al., 2011) using a standard processor(Intel(R) Core(TM)i5-2500 CPU) and a Matlab implementation. Desc-No refers to the number of descriptor vectors.

Method	Time(sec)	Desc-No
SIFT-128D	3396.86	$2,095,545$
RC-SIFT-64D	1639.82	$2,095,545$
SIFT-64D	1588.38	$2,095,545$

4 EVALUATION

In the image near-duplicate retrieval field, the performance of the RC-SIFT$-64D$, RC-SIFT$-32D$ and RC-SIFT$-16D$ is verified against the SIFT$-128D$, the SURF$-64D$ descriptor and the SIFT$-64D$ (Khan et al., 2011) descriptor mentioned in 2. The performance is measured on a large scale image databases using the vocabulary tree for feature indexing and $L1-$norm to achieve the image NDR task. The vocabulary trees are constructed as described in Subsection(3.3) for each kind of the used descriptors separately. In our experiment the initial number of clusters is $k = 10$.

To perform the NDR task, the similarity between normalized query vectors q_img and database vectors d_img is computed by traversing each vector in a vocabulary tree and it is given as (Nistèr and Stewènius, 2006):

$$s(q\_img, d\_img) = \left\| \frac{q\_img}{\|q\_img\|} - \frac{d\_img}{\|d\_img\|} \right\| \tag{4}$$

The normalization can be in any desired norm. In our experiment $L1-$norm and $L2-$norm.

In this work we used our own implementation of SIFT algorithm using some "Opencv" functions. SURF descriptors are computed by means of Opencv functions. The vocabulary tree is constructed using Matlab functions and VLFeat library. Moreover, we implement the SIFT$-64D$ described in (Khan et al., 2011) based on our implementation of SIFT algorithm by ignoring some patches of descriptors as it is described in (Khan et al., 2011).

Furthermore, the experiment is performed on benchmark databases as used in (Nistèr and Stewènius, 2006) (the dataset can be download from the website (Nistèr and Stewènius,)). These databases consist more than $10,000$ of indoor/outdoor images for about $2,500$ different scenes. The images of each scene differ through a combination of changes (additional blurring, scale change, rotation change, illumination decrease/increase, additional noise, viewpoint change) and another conditions (i.e., appear new objects and occlusion of objects).

The results of the experiments are evaluated by computing the *recall* value. Since we always have a fixed number of relevant images and the comparison is done use a ranked list of fixed length (i.e., length of one, three or ten images as indicated in the respective tables), we omit precision since it is directly correlated to the recall values. Considering N_q is the number of relevant images to a specific query image in the database, N_{qr} the number of relevant images obtained in matching results, then the *recall* is defined as follows:

$$Recall = \frac{N_{qr}}{N_q} \tag{5}$$

5 RESULT AND ANALYSIS

The results of SIFT$-64D$ and SIFT$-128D$ are evaluated in different cases using various kinds of image databases.

5.1 Mixed Database

The database described in (Nistèr and Stewènius, 2006) is used for experiments. This database contains four different images of $2,550$ different indoor/outdoor scenes i.e., $10,200$ images in total. Moreover, it contains a mixed complex images for some scenes (i.e., image of the same scene but present different arrangement of objects, appear/disappear of some objects in addition to changes in lightness, contrast, sharpness, scale, and viewpoint conditions). To test the robustness of the RC-SIFT in image NDR field, we select the first image of each scene as a query image while the remaining three images of each scene are used as a basic database for retrieval task (i.e., $2,550$ query image and $7,550$ database image). The features and descriptors are extracted using the original SIFT$-128D$, SURF$-64D$, SIFT$-64D$ (Khan et al., 2011) (produced by ignoring some patches of the RoIs) and our RC-SIFT$-64D$, RC-SIFT$-32D$ and RC-SIFT$-16D$. After that, the descriptors of each kind are indexed separately using the vocabulary tree of depth $L = 4$ and initial clusters $k = 10$. To achieve the retrieval task, $L1-$norm and $L2-$norm are used to compute the distance between the query image and the database images as described in equation 4. However, in our experiment $L1-$norm obtains better results than $L2-$norm. A query image is considered to be retrieved if its corresponding images in the database appear in the top three or ten retrieved images. Table 2 summarizes the results of the all proposed kinds of descriptors. It shows that the RC-SIFT$-64D$ obtained slightly better results than SIFT$-128D$. However, it presents that the RC-SIFT$-64D$ obtains better results than the RC-SIFT$-32D$ and RC-SIFT$-16D$. We assume that the ignoring of some patches of the original descriptors affects the performance of SIFT in solving image NDR task therefore, the performance of RC-SIFT64D seems to be superior than the performance of the suggested SIFT$-64D$ (Khan et al., 2011). The performance of SURF descriptor seems to be low in solving image NDR task. In our evaluation, it seems to be related with the complexity of the scenes. This result is consistent with other studies, e.g., (Khan et al., 2011). However, in (Khan et al., 2011) only a subset of a benchmark database (Nistèr and Stewènius, 2006) (i.e., $2,500$ images) are used whereas, our experiment is applied on the whole images in the same database (i.e., $10,200$ images). The worst results are obtained when the dimensionality of RC-SIFT is compressed to $16D$. Figure 2 presents a comparison between the all proposed methods to achieve image NDR task. It shows that the best results are found when the RC-SIFT$-64D$ is used. However, there is another examples where the SIFT$-128D$ preforms the best. Moreover, we note in many cases that despite the equivalent results of SIFT$-128D$ and RC-SIFT$-64D$ obtain better ranking of the results than SIFT$-128D$. Figure 3 presents an example where the performance of SIFT$-128D$ and RC-SIFT$-64D$ is equivalent but the ranking of the results found by RC-SIFT$-64D$ is better than SIFT$-128D$.

The experiment is repeated for vocabulary trees of depth $L = 1,2,3,4$ and the best retrieval results for all proposed descriptors are obtained when $L = 4$. Therefore, we present the results only when a vocabulary tree of depth $L = 4$ is used.

In the next step the robustness and invariant properties of the RC-SIFT-64 are verified against different image transformations, blurring, scale change and viewpoint change. These invariant and robustness properties are compared for the all proposed descriptors. We don't verify the invariant and robustness properties of RC-SIFT-32 and RC-SIFT-16 because their performance in solving NDR task is low comparing with the other used descriptors.

5.2 Image Affine Transformations

To verify the robustness of our RC-SIFT against different kind of image transformations in the field of image NDR, the performance of all proposed descriptors is evaluated against rotation change, illumination increase or decrease and adding different kinds of noise. To achieve this, 500 images of different scene of the benchmark database (Nistèr and Stewènius, 2006) are picked to test the invariance properties of features. The setting of generating the transformed images are similar to the setting applied in (Khan et al., 2011). The descriptors are indexed using a vocabulary tree of depth $L = 4$ and initial centers $k = 10$. The similarity is computed using $L1-$norm. A query image is considered to be found in the database if its corresponding database image appear in the top of the retrieved images.

Table 2: The retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D, RC-SIFT-32D and RC-SIFT-16D using a large ground truth database ($7,650$ images) with groups of three images belong to the same scene and using a set of $2,550$ query images, each of them has three related images in the database. The recall is computed firstly based on the top three retrieved images and secondly using the top ten retrieved images. The symbols RC-SIFT$-64D$(R) and RC-SIFT$-64D$(C) are used to refer for the compression of forms $4\times2\times8$ and $2\times4\times8$, respectively.

Method	Top 3 results	Top 10 results
SIFT-128D	0.4932	0.5797
SIFT-64D	0.2715	0.3534
SURF-64D	0.2432	0.2952
RC-SIFT64D(R)	**0.5067**	**0.6013**
RC-SIFT64D(C)	**0.4989**	**0.5914**
RC-SIFT-32D	0.2892	0.3365
RC-SIFT-16D	0.2460	0.2878

5.2.1 Rotation Change

To verify the rotation invariance, the first 500 images of the benchmark database (Nistèr and Stewènius, 2006) are rotated at different angels in a clockwise direction to generate 500 database images for each angle. Results of NDR task are summarized in Table 3. The results shows that all proposed descriptors (SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D) are rotation invariant. For a big rotation change the results shows that RC-SIFT-64D perform a little bit better than the other proposed descriptors.

5.2.2 Addition of Noise

To test noise invariance, three types of noise are applied for the first 500 images of the database (Nistèr and Stewènius, 2006). These types are: Gaussian noise, salt and pepper noise and multiplicative noise. The noise is added to the images using the following settings: Gaussian white noise with $\sigma^2 = 0.1$ and

(a) Query image

(b) The top three results found by SIFT$-128D$

(c) The top three results found by SIFT$-64D$ (Khan et al., 2011)

(d) The top three results found by SURF$-64D$

(e) The top three results found by RC-SIFT$-64D$

Figure 2: An example of the results of all proposed methods to perform the image NDR task. In this example RC-SIFT$-64D$ shows the best matching results.

Table 3: The performance comparison of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D(R)and RC-SIFT-64D(C) using a ground truth database (500 images) and a set of 500 query images, each of them has one rotated image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result. The experiment is repeated for the rotation values: $\{40^o, 135^o, 215^o, 250^o\}$.

Method	40^o	135^o	215^o	250^o
SIFT-128D	**0.934**	**0.926**	**0.934**	0.918
SIFT-64D	0.928	0.924	0.926	**0.928**
SURF-64D	0.933	**0.926**	**0.933**	0.920
RC-SIFT64 (R)	**0.931**	0.924	0.930	**0.924**
RC-SIFT64 (C)	0.928	0.924	0.928	0.923

$\sigma^2 = 0.2$, salt and pepper noise with density of 15% and 35% and multiplicative white noise with mean 0 and $\sigma^2 = 0.04$. The performance of the all used

(a) Query image

(b) The top three query results by SIFT$-128D$

(c) The top three query results by RC-SIFT$-64D$

Figure 3: Comparison of retrieval results of SIFT$-128D$ and RC-SIFT$-64D$ descriptors. This example shows an equivalent performance of the RC-SIFT$-64D$ and SIFT$-128D$ in retrieving the belonged images to the same scene. However, RC-SIFT$-64D$ presents better raking of the results than SIFT$-128D$.

descriptor in this work are presented in Tables 4, 5 and 6. These results show that performance all proposed descriptors decrease very strongly when the ratio of noise incease (see Tables 4 and 5). However, in case of using the salt and pepper noise RC-SIFT-64D obtains better results than the other descriptors even though when the ratio of noise increases.

Table 4: The performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT$-64D$ using a ground truth database (500 images) and a set of 500 query images, each of them has one Gaussian noised image in the database. The experiment is repeated for $\sigma^2 = 0.1$ and $\sigma^2 = 0.2$. For each query image we check if its corresponding database image appear as the first retrieved image in the result.

Method	$\sigma^2 = 0.1$	$\sigma^2 = 0.2$
SIFT-128D	**0.684**	**0.356**
SIFT-64D	0.648	0.290
SURF-64D	0.644	0.352
RC-SIFT-64D(R)	**0.682**	0.352
RC-SIFT-64D(C)	0.679	**0.354**

5.2.3 Illumination Change

The illumination invariance is verified in the cases of increase and decrease the brightness of the 500 test

Table 5: The performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D using a ground truth database (500 images) and a set of 500 query images, each of them has one salt and pepper noised image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result. The experiment is performed for two level of noise density (i.e., 15% and 35%).

Method	15%	35%
SIFT-128D	0.826	0.202
SIFT-64D	0.822	0.152
SURF-64D	0.812	0.145
RC-SIFT-64D(R)	**0.834**	**0.208**
RC-SIFT-64D(C)	**0.831**	**0.205**

Table 6: The performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D using a ground truth database (500 images) and a set of 500 query images, each of them has one multiplicative noise noised image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result.

Method	σ^2	Recall
SIFT-128D	0.04	**0.98**
SIFT-64D	0.04	0.822
SURF-64D	0.04	0.801
RC-SIFT-64D(R)	0.04	**0.972**
RC-SIFT-64D(C)	0.04	0.970

images. This is done by adding or subtracting a value of all pixel's channels(i.e., the channels red, green and blue of each pixel are incremented equally). The values of pixel's channels are adjusted to be within the range $0-255$. The brightness effect is tested using the values $\{50, 70, 100, 120\}$ and the darkness effect is test using the values $\{-30, -50, -70, -90\}$. Results of NDR tasks summarized in Tables 7 and 8 describe that all kinds of used descriptors perform well for illumination increase and decrease.

5.3 Image Blurring

To test the robustness of descriptors against image blurring, three blurred image databases are generated using the first 500 images of the benchmark database (Nistèr and Stewènius, 2006) and using three different values of Gaussian filter i.e., $\sigma^2 = 5$, $\sigma^2 = 10$ and $\sigma^2 = 20$. Table 9 shows that the performance degrade most clearly when the ratio of blurring increase (i.e., when the value of σ^2 increases). However, for a small amount of blurring the descriptors seem to be invariant. For a big amount of blurring, RC-SIFT-64D is superior in matching images.

Table 7: The retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D using a ground truth database (500 images) and a set of 500 query images, each of them has one brightened image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result. The results are checked for the following brightness values: $\{50, 70, 100, 120\}$.

Method	50	70	100	120
SIFT-128D	**1.00**	**1.00**	**0.968**	**0.912**
SIFT-64D	**0.998**	**0.998**	0.942	**0.902**
SURF-64D	0.995	0.993	0.932	0.881
RC-SIFT64(R)	0.995	0.995	**0.956**	**0.902**
RC-SIFT64(C)	0.970	0.970	**0.956**	**0.902**

Table 8: The retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D using a ground truth database (500 images) and a set of 500 query images, each of them has one darkened image in the database. The performance is presented using the darkness values: $\{-30, -50, -70, -90\}$. For each query image we check if its corresponding database image appear as the first retrieved image in the result.

Method	-30	-50	-70	-90
SIFT-128D	**1.00**	**0.992**	**0.958**	0.810
SIFT-64D	**0.998**	**0.992**	**0.956**	0.802
SURF-64D	0.997	0.992	0.947	0.822
RC-SIFT64(R)	**1.00**	**0.990**	0.955	**0.822**
RC-SIFT64(C)	0.997	0.988	0.955	**0.820**

Table 9: Comparison of retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-64D using a ground truth database (500 images) and a set of 500 query images, each of them has one blurred image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result. The experiment is repeated for different level of blurring using $\sigma = 5$, $\sigma = 10$ and $\sigma = 20$.

Method	$\sigma = 5$	$\sigma = 10$	$\sigma = 20$
SIFT-128D	**0.922**	0.426	0.358
SIFT-64D	0.836	0.368	0.330
SURF-64D	**0.859**	0.347	0.298
RC-SIFT-64D(R)	0.830	**0.366**	**0.386**
RC-SIFT-64D(C)	0.834	**0.368**	**0.388**

5.4 Scale Change

The robustness of all proposed descriptors is verified against scaling change by selecting 500 different scenes of the benchmark database (Nistèr and Stewènius, 2006) for which there are two images taken at different scales. Some of the selected images have additional viewpoint change as well. The first image of each scene is used as a query image and the second one is used as an available image in the image database. Table 10 shows that both SIFT$-128D$ and RC-SIFT$-64D$ perform consistent in the case of scale change. Moreover, it presents that the SURF$-64D$ descriptors perform the worst in this case.

Table 10: Comparison of retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-$64D$ using a ground truth database (500 images) and a set of 500 query images, each of them has one salt and pepper noised image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result.

Method	Recall
SIFT-128*D*	**0.801**
SIFT-64*D*	0.763
SURF-64*D*	0.533
RC-SIFT-64*D*(R)	**0.801**
RC-SIFT-64*D*(C)	**0.801**

5.5 Perspective Change

To test the invariance of descriptors against perspective change. 500 different scenes of the benchmark database (Nistèr and Stewènius, 2006) are selected for which there are two images taken at different viewpoint angles. The first image of each scene is used as a query image and the other one is used as an image in the database. The results are presented in Table 11, which describes that contrary to the other kinds of changes, the robustness of all proposed descriptors against perspective change decrease. But SIFT$-128D$ and RC-SIFT-$64D$ still have the best performance.

Table 11: Comparison of retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-$64D$ using a ground truth database (500 images) and a set of 500 query images, each of them has one salt and pepper noised image in the database. For each query image we check if its corresponding database image appear as the first retrieved image in the result.

Method	Recall
SIFT-128*D*	0.626
SIFT-64*D*	0.602
SURF-64*D*	0.439
RC-SIFT-64*D*(R)	**0.720**
RC-SIFT-64*D*(C)	**0.717**

6 CONCLUSION

In this work, we consider the fact that "the sparsity of fixed amount of feature increase as their dimensionality increase" (Steinbach et al., 2003) to reduce dimensionality of the SIFT descriptor from $128D$ to $64D$. The goal of dimensionality reduction is to decrease the sparsity of SIFT descriptors, speeding up the indexing process and improve the performance of SIFT descriptors in the image NDR field. We verified in this work the performance of the RC-SIFT$-64D$ (for both horizontal and vertical compression), RC-SIFT$-32D$, RC-SIFT$-16D$ against the original SIFT$-128D$ to solve image NDR tasks using a benchmark which contains different kind of indoor/outdoor images. The experiments show a slight improvement in matching results when tested on benchmark databases. Moreover, the RC-SIFT$-64D$ needs shorter time for indexing and less memory than the original SIFT$-128D$. However, the performance of RC-SIFT$-32D$ and RC-SIFT$-16D$ decrease, due to the compression of descriptors information in both direction at once. The robustness and stability of our suggested RC-SIFT$-64D$ are verified against different kinds of image affine transformation, blurring change, scale change and viewpoint change. However, the results shows that the RC-SIFT$-64D$ descriptors are invariant to image affine transformation in some specific ranges. Moreover, it presents that RC-SIFT$-64D$ descriptors are robust against image blurring, scale change and perspective change. In addition, its descriptors are more robustness than the other presented descriptors against a big change in the rotation, some kinds of noise, big amount of blurring and viewpoint change. In the next step, we will attempt to improve the performance of RC-SIFT$-64D$ by adapting suitable weights inspired by the relation between features to improve the performance of matching in the field of image NDR.

In future work, we will also evaluate if the more robust performance of the RC-SIFT$-64D$ can be used in the field of human visual attention, e.g., as a more stable predictor for creating a saliency map of human gaze as discussed in a previous study (Steffen et al., 2012). Furthermore, the more efficient RC-SIFT$-64D$ approach may improve interactive image search when a large scale image collection is used as e.g., in (Low et al., 2014) and (Beecks and Seidl, 2009).

ACKNOWLEDGEMENTS

I would like to thank the state Saxony-Anhalt for the financial support of this work.

REFERENCES

Auclair, A., Vincent, N., and Cohen, L. (2006). Hash functions for near duplicate image retrieval. In *Applications of Computer Vision (WACV)*, pages 7–8.

Beecks, C. and Seidl, T. (2009). Visual exploration of large multimedia databases. In *Data Management and Visual Analytics Workshop*.

Chu, L., Jiang, S., Wang, S., Zhang, Y., and Huang, Q. (2013). Robust spatial consistency graph model for partial duplicate image retrieval. In *Multimedia, IEEE Transactions on*, pages 1982–1996.

Chum, O., Philbin, J., Isard, M., and Zisserman, A. (2007). Scalable near identical image and shot detection. In *Proc. CIVR*.

Chum, O., Philbin, J., and Zisserman, A. (2008). Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*.

Grauman, K. and Darrell, T. (2005). Pyramid match kernels: Discriminative classification with sets of image features. In *Proc. ICCV*.

Grauman, K. and Darrell, T. (2007). The pyramid match kernel: Efficient learning with sets of features. In *The Journal of Machine Learning Research*, pages 725–760.

Jègou, H., Douze, M., Schmid, C., and Pèrez, P. (2010). Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Jiang, M., Zhang, S., Li, H., and Metaxas, D. N. (2015). Computer-aided diagnosis of mammographic masses using scalable image retrieval. In *Biomedical Engineering, IEEE Transactions on*, pages 783–792.

Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *in: CVPR, issue 2*, page 506513.

Khan, N., McCane, B., and Wyvill, G. (2011). Sift and surf performance evaluation against various image deformations on benchmark dataset. In *Digital Image Computing Techniques and Applications (DICTA)*.

Li, J., Qian, X., Li, Q., Zhao, Y., Wang, L., and Tang, Y. Y. (2014). Mining near duplicate image groups. In *Springer Science and Business Media New York*.

Low, T., Hentschel, C., Stober, S., Sack, H., and Nürnberger, A. (2014). Visual berrypicking in large image collections. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: fun, fast, foundational*, pages 1043–1046. New York, NY : ACM.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. In *Journal of Computer Vision*, pages 91–110.

Nistèr, D. and Stewènius, H. Recognition benchmark images. In *available at http://www.vis.uky.edu/∼stewe/ukbench/*.

Nistèr, D. and Stewènius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168.

Steffen, J., Christian, H., Alyosef, A. A., Tönnies, K., and Nürnberger, A. (2012). Rotational invariance at fixation points - experiments using human gaze data. In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 451–456.

Steinbach, M., Ertoz, L., and Kumar (2003). The challenges of clustering high dimensional data. In *Wille LT, editor. New Vistas in Statistical Physics-Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag.

Xu, D., Cham, T., Yan, S., Duan, L., and Chang, S. (2010). Near duplicate identification with spatially aligned pyramid matching. In *IEEE Trans. Circuits and Systems for Video Technology*, pages 1068–1079.

Yang, Y. and Newsam, S. (2008). Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *Proceedings of the 15th IEEE on Image Processing, San Diego*, pages 1852–1855. USA.

Zhang, C., Wang, S., Huang, Q., Liu, J., Liang, C., and Tian, Q. (2013). Image classification using spatial pyramid robust sparse coding. In *Pattern Recognition letters*, pages 1046–1052.

Zhang, D. Q. and et al. (2004). Detecting image near-duplicate by stochastic attribute relational graph matching with learning. In *Proceedings of the 12th annual ACM international conference on Multimedia*.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS