Near-Duplicate Retrieval: A Benchmark Study of Modified SIFT Descriptors

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
{afraa.ahmad-alyosef,andreas.nuernberger}@ovgu.de

Abstract. Local feature detectors and descriptors are widely used for image near-duplicate retrieval tasks. However, most studies and evaluations published so far focused on increasing retrieval accuracy by improving descriptor properties and similarity measures. There has been almost no comparisons considering the modification of the descriptors and the impact on accuracy *and* performance, which is especially of interest for interactive retrieval systems that require fast system responses. Therefore, we evaluate in this paper accuracy and performance of variations of SIFT descriptors (reduced SIFT versions, RC-SIFT$-64D$, the original SIFT$-128D$) and SURF$-64D$ in two cases: Firstly, using benchmarks of various sizes. Secondly, using one particular benchmark but extracting varying amounts of descriptors. Another aspect that has been almost neglected in previous benchmarks is the combination of different affine transformations in near-duplicate images. A problem that many real-world systems have to face. Therefore, we provide in addition results of a comparative performance analysis using benchmarks generated by combining several image affine transformations.

1 Introduction

Finding near-duplicate images is still a very challenging task, due to the various scenarios in which near-duplicate images could have been created: using different cameras or slightly different positions; different camera settings or lenses; different lighting conditions; post processing of images using image processing software, may be even to hide illegal use of copyrighted material. Therefore, the features and similarity models used to find near-duplicate images have to be quite robust.

The image near-duplicate retrieval process can be divided into several stages depending on the used techniques and the goal of the retrieval task that should be supported. However, the first step is to represent images by means of one or more kinds of expressive features. The goal is to reduce the amount of processed information. The scale invariant feature transform (SIFT) provides keypoints and descriptors that are used in many NDR approaches [6,8,9,11]. This is mainly due to its invariance to scale and rotation variation and its robust performance even

A. Fred et al. (Eds.): ICPRAM 2016, LNCS 10163, pp. 121–138, 2017.
DOI: 10.1007/978-3-319-53375-9_7

if the images differ in perspective, noise, and illumination [1]. The huge amount of descriptors that are required to represent a large scale image dataset and the high dimensionality of these descriptors imposes strong demands on memory and computing power in order to support near-duplicate retrieval tasks. To reduce the amount of extracted data, we proposed a method in [22] to compress the region around the SIFT descriptor. This compression leads to a decrease in time and memory usage of feature indexing and matching. We showed in [22] that the region compressed SIFT (RC-SIFT) descriptors are invariant to affine transformation change and perform robust as the original SIFT features to viewpoint change, scale change and blurring change. In this work, we evaluate the performance of the RC-SIFT$-64D$ [22] descriptor in solving near-duplicate retrieval tasks in two cases: Firstly, for benchmarks of various sizes. Secondly, when a specific benchmark is used but descriptor databases of various sizes are extracted from images. After that, the robustness of the RC-SIFT$-64D$ descriptor is evaluated by various combinations of image affine transformations.

The remainder of this paper is organized as follows. Section 2 provides a short definition of near-duplicate images. Section 3 gives an overview of prior work related with the SIFT algorithm and image NDR algorithms. Section 4 details the proposed method to produce the region compressed SIFT descriptor. Section 5 presents the settings of our experiments and the measures used to describe the performance. Section 6 discusses the results of experiments. Finally, Sect. 7 draws conclusions of this work and discusses possible future work.

2 Near-Duplicate Images

To clarify the meaning of near duplicate images, we define first briefly the concept of exact duplicate images: Two images are considered as exact duplicate iff there is no difference between both of them [7], i.e. all corresponding pixels are identical. Two images are defined to be near-duplicates (ND) [7,10] if they show the same scene (the same object) but they differ (slightly) in some properties that can be represented by affine transformations (such as noise, blurring, compression, contrast etc.) or time conditions (lighting or illumination conditions) or the images are even taken from different perspective. Unfortunately, so far the range of transformations in which images are still considered near-duplicates is not yet clearly defined in the literature. Moreover, the evaluation of NDR algorithms is still challenging and focuses mostly on comparisons of rankings or performance.

3 Related Works

The SIFT detector and descriptor has been shown superior performance to several other low dimensional descriptors [25]. Therefore, it has been widely used in image near-duplicate retrieval field [8,9], image classification [2] and processing medical images [3] i.g., checking the existence of cancerous growth.

To accelerate the feature indexing process several methods have been proposed to reduce the length of the original SIFT descriptor vector. This is achieved either by ignoring some patches of the original descriptor [13] to get $96D$, $64D$ and $32D$ descriptors or by employing principle component analysis to obtain $64D$ SIFT descriptors [21]. This approach is in need of an off-line training stage to compute the suitable eigenvalue vector. The issue of extracting variable amounts of SIFT features is addressed in [26] by pruning the extracted features based on their contrast property. In [22], we proposed a method to compress the descriptor without the need for a training stage and without ignoring any part of the region around the keypoint. The details of this method are also described in Sect. 4.

To accelerate the features matching process, various methods have been proposed to structure, index or quantize SIFT features. In [1] the best-bin-first algorithm based on a $kd-$tree has been used to speed up the process of matching in 128 dimensional space. However, this method is not appropriate for large scale feature databases due to the required time for backtracking through the tree which leads to decreased $kd-$tree efficiency. To overcome this problem, direct clustering specifically, $k-$means clustering have been used in [14–17], to group the SIFT descriptors into k groups. The obtained cluster centers construct a bag of words; each descriptor is assigned to its closest word in this bag. In this way, images are represented in form of vectors of bag of words. The concept of bag of words is extended in [4,5] and combined with further training steps to improve the retrieval of relevant scenes or objects. In [20] a bag of words is built to construct image vectors. The dimensionality of these vectors is jointly optimized and reduced by applying principle component analysis. A vocabulary tree and the inverted file concept are constructed based on hierarchical $k-$means clustering in [2,3] to refine the splitting of features into groups. In [27] retrieval performance is improved through re-ranking the retrieved images based on the scale and orientation properties of the extracted features.

The next subsection gives an overview of the SIFT detector and descriptor algorithm to simplify the description of the region compressed SIFT descriptor later on.

3.1 SIFT$-128D$ Descriptor

As described in [1] the original SIFT detector and descriptor algorithm consists of four major stages: scale invariant peak detection, feature localization, orientation assignment and descriptor construction. In the first stage the locations and scales of interest points (called keypoints) are identified. This is achieved by building a Gaussian pyramid and searching for the local maxima or minima in the difference of Gaussian (DoG) images. The second stage determines the location of the candidate keypoints and rejects the keypoint that have low contrast or are poorly localized on an edge. The third stage assigns the dominant orientation for each keypoint based on the properties of its local image patch. In the final stage keypoint descriptor is computed based on the local gradient and orientation data of a patch around a keypoint. This descriptor is built in form of $n \times n$ array of orientation histogram. For each bin in this histogram r orientations are

assigned, so that each descriptor has $n \times n \times r$ element. The size of descriptor is determined by the width of a histogram n and the number of orientations r. The standard length of the SIFT descriptor [1] is 128 elements. Figure 1(a) shows the final form of the SIFT$-128D$ descriptor around a keypoint.

Since, the sparsity of descriptors may increase as the dimensionality of the SIFT descriptor increase [12] and this may affect the accuracy of descriptor indexing in image NDR, we proposed an approach [22] to compress the dimensionality of the SIFT descriptor. In the next section, we describe this approach in detail.

4 Region Compressed SIFT Descriptor

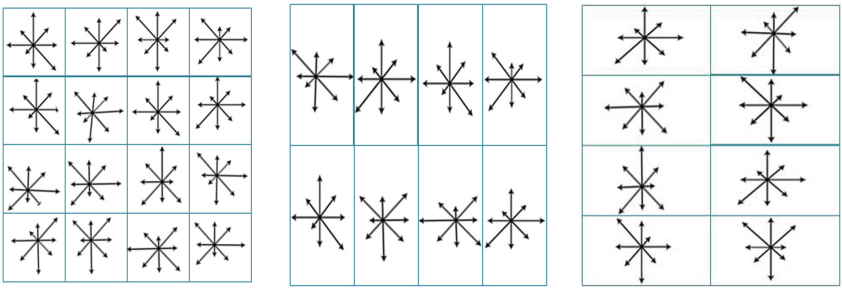
In [22], we proposed an approach to compress the dimensionality of the original SIFT descriptor from $128D$ to $64D$ without ignoring any part of the local patch around a keypoint and without the need for a training stage. This approach aims to reduce the usage of memory and the amounts of processed data. Moreover, it improves the retrieval task in the near-duplicate retrieval field. To achieve this, SIFT features are first extracted in the same way described in [1] (see Sect. 3.1). After that, the SIFT local descriptor is computed over a local image region around each keypoint. The original SIFT descriptor has the dimensionality of $4 \times 4 \times 8$ and it is computed in form of three dimensional histograms centered at the keypoint. This gradient orientation histogram explain that a keypoint may be located at any allowed position in the local patch around a keypoint in vertical and horizontal location (i.e. 4×4 locations). For each location eight directions are assigned. We proposed in [22] 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 all possible horizontal shiftings (i.e., four horizontal shiftings) in all directions only two vertical shifting exists. For each of this (4×2) locations eight directions are assigned. As a result we obtain $4 \times 2 \times 8$ histogram i.e., $64D$ SIFT descriptor. We called 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 are the number of bins in H_y and H_x, respectively. The variables y and x are defined as $y = 0, ..., N_y - 1$, and $x = 0, ..., N_x - 1$ and N_θ defines the number of orientations in each bin of a histogram and its values: $\theta = 0, ..., N_\theta - 1$. The best performance [22] is found when $N_y = 2$, $N_x = 4$ and $N_\theta = 8$ and when $N_y = 4$, $N_x = 2$ and $N_\theta = 8$. These two forms of the RC-SIFT$-64D$ descriptor are presented in Fig. 1(b) and (c) respectively. Contrary



(a) The $4 \times 4 \times 8$ SIFT descriptor.

(b) The $2 \times 4 \times 8$ RC-SIFT descriptor.

(c) The $4 \times 2 \times 8$ RC-SIFT descriptor.

Fig. 1. The Different forms of the SIFT descriptor. (a) presents the original SIFT$-128D$ descriptor. Whereas (b) and (c) show the RC-SIFT$-64D$ descriptors of forms $4\times2\times8$ (referred as RC-SIFT-64(R)) and $2\times4\times8$ (referred as RC-SIFT-64(C)) respectively. The symbols RC-SIFT-64(R) and RC-SIFT-64(C) are used in the all presented tables.

to the methods proposed in [13,21], which ignore some parts of SIFT descriptor or need for a training stage as described in Sect. 3, we compressed the SIFT descriptor without ignoring any region of the local patch around a keypoint and without the need for an off-line training stage.

In the following, we present an extensive benchmark study to verify the performance of the RC-SIFT$-64D$ in solving near-duplicate retrieval tasks in the following scenarios:

- Various benchmarks: Check the performance using various image databases. We apply our experiments on UKbench [2] and Caltech-Buildings benchmark [24] (see Sect. 6).
- Benchmarks of various sizes: Verify the performance using benchmarks of various sizes produced from UKbench benchmark (see Subsect. 6.1).
- Descriptor databases of various sizes: Evaluate the performance for a variable number of extracted features (see Subsect. 6.2).
- Combination of image affine transformations: The robustness of the RC-SIFT descriptor is verified when a combination of affine transformations applied to images. The following combination are applied in this work: illumination and rotation changes (see Subsect. 6.3), illumination change and adding noise (see Subsect. 6.3) and combination of adding noise and rotation (see Subsect. 6.3).
- Combination of Blurring and image affine transformations: The robustness of the RC-SIFT and all other proposed descriptors is verified against combinations of blurring and affine transformations (see Subsect. 6.4).

5 Evaluation

The performance of the RC-SIFT$-64D$ descriptor is compared to the original SIFT$-128D$, the SURF$-64D$ [18] and the SIFT$-64D$ [13] descriptors mentioned in Sect. 3 by solving different image near-duplicate retrieval tasks. To

achieve this, large scale image benchmarks of different sizes and resolutions are used. In the following subsections the used image benchmarks and the evaluation measures are described.

5.1 Benchmark Datasets

In this work, the experiment is performed on two different benchmark datasets. The first benchmark is UKbench [2] (this dataset can be download from [28]). From this benchmark various image datasets of sizes (10200, 6000, 4000 and 2000) are formed as described in Subsect. 6.1. The resolution of these images is 640×480. This benchmark consist of indoor/outdoor images of different scenes in groups of four images for each scene. The images of each scene vary in one or more of the following conditions: view point, scale, lightness, appear new objects and occlusion of objects. The second benchmark is the Caltech-Buildings [24,29] image dataset which contains 250 images for 50 different buildings around the Caltech campus.(i.e. in groups of five images for each building taken at different perspectives and scales). Moreover, this benchmark contains of high resolution image (i.e. the resolution of each image is 2048×1536).

5.2 Evaluation Measures

To evaluate the performance of the proposed descriptors, the descriptors of each kind are firstly indexed using the vocabulary tree concept as described in [22]. In our experiment the initial number of clusters is $k = 10$. The similarity between two images is computed by traversing each normalized vector of the query image q_img in the vocabulary tree of the database images db_img and it is given as [2]:

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

All implementations are build using windows platform and Visual C++ programing language with "Opencv" functions. Matlab functions are used to apply combination of image affine transformations and blurring. The Matlab library VLFeat is employed to index the extracted descriptors.

The results of the experiments are evaluated by computing the *recall* value. 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:

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

The mean recall MR for a set of query images is computed as

$$MR = \frac{1}{Q} \sum_{q=1}^{Q} Recall(q) \tag{6}$$

Where Q is the total number of query images. To measure the amount of difference between the MR and the recall value of each query image, the variance of the recall values VR is computed as:

$$VR = \frac{1}{Q}\sum_{q=1}^{Q}\left(Recall(q) - MR\right)^2 \tag{7}$$

However, the computation of the recall ignores the ranking of the relevant images in the results. Therefore, we compute the mean average precision MAP which characterizes the relation between the relevant images and their ranking in the results [23] and it is defined as:

$$MAP = \sum_{q=1}^{Q}\frac{Ap(q)}{Q} \tag{8}$$

where $Ap(q)$ is the average precision for image q and is given as:

$$AP(q) = \frac{1}{n}\sum_{i=1}^{n} p(i) \times r(i) \tag{9}$$

where $r(i) = 1$ if the i^{th} retrieved image based on the query image q is one of the relevant images and $r(i) = 0$ otherwise, $p(i)$ is the precision at the i^{th} element.

6 Result and Analysis

The results of the SIFT$-64D$ and the original SIFT$-128D$ are evaluated in different cases using various kinds of image benchmarks as described in the following.

6.1 UKbench Benchmark

From this benchmark [2] we construct four image datasets of different size to test the robustness of the RC-SIFT descriptors in solving the task of image near-duplicate retrieval. For the experiment, 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. The constructed benchmarks have the sizes 10200, 6000, 4000 and 2000 images and they are referred as $UKBench10$, $UKBench6$, $UKBench4$ and $UKBench2$, respectively. The features and descriptors are extracted using the original SIFT$-128D$, SURF$-64D$, SIFT$-64D$ [13] and our RC-SIFT$-64D$(R) and RC-SIFT$-64D$(C) descriptors. After that, the descriptors of each kind are indexed separately using a vocabulary tree of depth $L = 4$ and initial clusters $k = 10$. To achieve the retrieval task, the distance between a query image and database images is computed as described in Eq. 4 using the $L1-$norm and $L2-$norm. However, in our experiment the $L1-$norm

obtains better results than the $L2-$norm. Therefore, we present the results obtained when the $L1-$norm is used. A query image is retrieved if its corresponding images in a database appear in the top three, ten or fifty retrieved images.

Table 1 summarizes the results of all proposed descriptors using benchmarks of various sizes. In this table a query image is retrieved if its relevant images in the benchmark appear in the top three retrieved images. It shows that the RC-SIFT$-64D$ obtained slightly better results than SIFT$-128D$. The values of variance are small for all descriptors but the smallest values are found for SURF-$64D$ and SIFT-$64D$ [13]. The best mean average precision is found for the RC-SIFT$-64D$ and then for the SIFT$-128D$ descriptors. Tables 2 and 3 present the performance of various descriptors when the belonging images appear in the top ten or fifty results, respectively. In the both cases the best performance is shown by RC-SIFT$-64D$ and SIFT$-128D$.

The results presented in Tables 1, 2 and 3 show that, if the mean recall increase the variance values increase for both SURF-$64D$ and SIFT-$64D$ [13]. Whereas, for both of RC-SIFT$-64D$ and SIFT$-128D$ the variance of recall decrease as the mean recall value increases. Table 4 provides a qualitative comparison between all proposed descriptors. For this example it shows that the best results are found when the RC-SIFT$-64D$ is used. However, there are of course other examples where the SIFT$-128D$ preforms best. Moreover, we note in many cases that despite the equivalent recall results of SIFT$-128D$ and RC-SIFT$-64D$ descriptors, the RC-SIFT$-64D$ obtains better mean average precision values than the SIFT$-128D$ descriptor. Table 5 presents an example of the results 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$-128D$.

Table 1. The retrieval performance of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-$64D$ using benchmarks of various sizes $UKBench10$, $UKBench6$, $UKBench4$ and $UKBench2$, each of them contains images of various scenes with groups of four images belong to the same scene. The first image of each scene is used as a query image. The mean recall MR, the variance of recall VR and mean average precision MAP are computed in percent based on the top three retrieved images. The symbols RC-SIFT$-64D$(R) and RC-SIFT$-64D$(C) are used to refer the compression of forms $4 \times 2 \times 8$ and $2 \times 4 \times 8$, respectively.

Method	$UKBench10$			$UKBench6$			$UKBench4$			$UKBench2$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-$128D$	49.3	15.1	47.5	55.3	14.4	53.5	53.1	14.3	51.3	51.6	13.4	**49.7**
SURF-$64D$	24.3	13.2	22.9	26.3	12.3	24.6	25.0	11.1	23.4	26.1	11.2	25.5
SIFT-$64D$	27.2	11.2	25.2	29.9	11.5	27.9	27.1	10.9	25.2	25.6	10.0	24.0
RC-SIFT-$64D$(R)	**50.7**	14.8	**48.8**	**57.1**	13.7	**55.2**	**54.5**	13.6	**52.7**	**54.9**	12.5	**53.1**
RC-SIFT-$64D$(C)	**49.9**	14.6	**47.9**	**56.3**	14.0	**54.1**	**53.1**	13.7	**51.7**	**51.8**	12.8	**49.7**

Table 2. The retrieval performance of SIFT−128D, SIFT−64D, SURF−64D and our RC-SIFT-64D using benchmarks of various sizes ($UKBench10$, $UKBench6$, $UKBench4$ and $UKBench2$), each of them containing images of scenes with groups of four images belong to the same scene. The first image of each scene is used as a query image. The MR, VR and MAP are computed based on the top ten retrieved images.

Method	$UKBench10$			$UKBench6$			$UKBench4$			$UKBench2$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-128D	58.7	15.2	50.1	64.8	13.7	57.3	**62.3**	14.1	**54.9**	61.0	13.3	**53.3**
SURF-64D	30.2	14.7	23.3	34.2	14.6	28.3	31.9	13.7	24.2	33.7	12.9	28.2
SIFT-64D	36.2	14.0	28.2	39.0	14.3	31.2	35.4	13.5	28.1	30.1	12.3	26.6
RC-SIFT-64D(R)	**60.7**	14.8	**52.7**	**67.1**	13.1	**59.4**	**64.6**	13.4	**57.0**	**64.9**	12.7	**57.4**
RC-SIFT-64D(C)	**59.2**	14.9	**50.6**	**65.1**	13.5	**58.0**	62.0	13.6	54.5	**61.4**	12.8	**53.5**

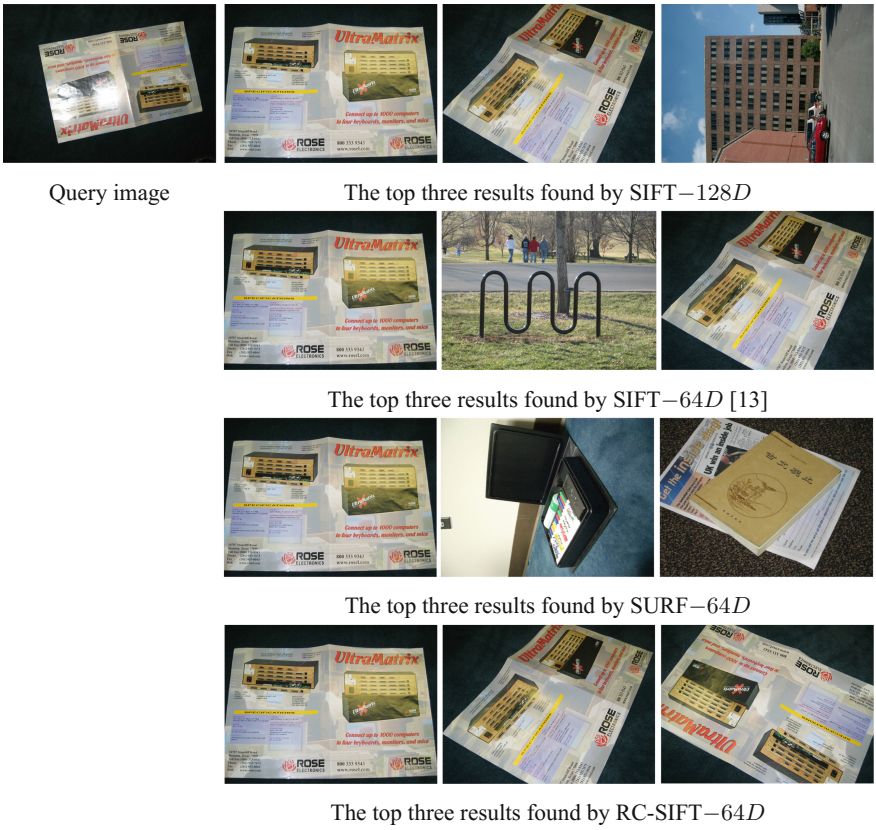
Table 3. The retrieval performance of SIFT−128D, SIFT−64D, SURF−64D and our RC-SIFT-64D using benchmarks of various sizes $UKBench10$, $UKBench6$, $UKBench4$ and $UKBench2$, each of them contains images of scenes with groups of four images belong to the same scene. The MR, VR and MAP are computed based on the top fifty retrieved images and the task is to retrieve the belonging to the same scene images in the top fifty results.

Method	$UKBench10$			$UKBench6$			$UKBench4$			$UKBench2$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-128D	69.4	13.0	51.2	75.0	11.1	58.4	73.0	11.5	56.0	72.4	11.5	**54.5**
SURF-64D	45.1	15.6	25.7	50.8	14.9	30.0	47.0	15.0	26.8	47.2	14.0	26.9
SIFT-64D	49.1	15.1	29.4	52.0	14.8	32.3	47.9	14.9	29.2	46.3	14.1	28.0
RC-SIFT-64D(R)	**72.2**	11.8	**53.9**	**77.6**	9.8	**60.6**	**75.5**	10.3	**58.1**	**76.1**	9.6	**58.6**
RC-SIFT-64D(C)	**70.2**	13.0	**52.1**	**75.6**	10.9	**59.0**	**73.1**	11.3	**56.2**	**72.7**	11.0	54.9

6.2 Caltech-Buildings Benchmark

In this case, because of the high resolution of images of this benchmark [24], we determine three different threshold to extract different numbers of descriptors from the images. The used number of features in this experiment are 2500, 1000 and 500 and they are referred as $Caltech$–2500, $Caltech$–1000 and $Caltech$–500, respectively. We compute the performance of all proposed descriptors in solving the task of image near-duplicate retrieval with the three descriptors databases of different sizes. For the experiment, we select the first image of each scene as a query image while the remaining four images of each scene are used as a basic database for retrieval task. In this experiment a vocabulary tree of depth $L = 3$ and initial clusters $k = 10$ is used. In addition, the $L1 - norm$ is used to normalize the vectors of images. Table 6 presents the results of all descriptors when the related images appear in the top four results. It shows a comparable performance of the RC-SIFT−64 and the SIFT−128 descriptors. However, Tables 7 and 8 present a little bit enhancement in the performance of the RC-SIFT−64 compared to the SIFT−128 descriptor. Moreover, the results show that the performance of the SIFT−64 and the SURF−64 descriptors for this benchmark

Table 4. Performance comparison between all proposed methods in solving the image near-duplicate retrieval task. The results presentthat RC-SIFT−64D shows the best performance.



is better than their performance for the benchmarks constructed based on the UKbench benchmark.

In the next step the robustness and invariant properties of the SIFT−128, SIFT−64, SURF−64 and RC-SIFT−64 are verified against a combination of different kinds of image transformations and blurring.

6.3 Combination of Image Affine Transformations

Various experiments are accomplished to verify the robustness of the original SIFT−128D, SIFT−64D, SURF−64D and RC-SIFT−64 descriptors against combinations of image transformations in the field of image NDR. In this work we discuss the following kinds of combinations: a combination of illumination increase or decrease with rotation change, illumination increase or decrease with

Table 5. Equivalent performance of the SIFT−128D and the RC-SIFT−64D descriptor but different ranking of the retrieved results. In this example RC-SIFT−64D presents better raking of the results than SIFT−128D.

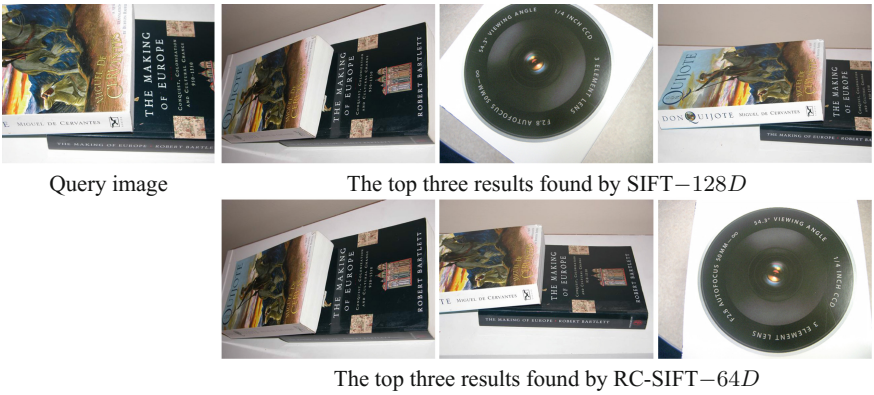


Table 6. The retrieval performance of SIFT−128D, SIFT−64D, SURF−64D and RC-SIFT-64D when various number of features are extracted from the images (i.e. 500, 1000 and 2500 features for each image). This is done using the Caltech-Buildings. A query image is retrieved when one or more of its related images is obtained in the top four results. The results are presented in percent.

Method	$Caltech - Buil500$			$Caltech - Buil1000$			$Caltech - Buil2500$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-128D	**44.0**	8.2	39.9	**43.0**	6.5	**40.2**	**39.5**	8.2	**36.7**
SIFT-64D	39.5	8.3	35.4	38.2	6.5	34.1	36.7	7.7	31.9
SURF-64D	33.2	7.5	29.3	31.7	6.7	28.1	29.8	7.2	26.1
RC-SIFT64D(R)	**44.0**	7.8	**40.3**	42.8	7.2	39.8	**39.0**	8.4	**36.4**
RC-SIFT64D(C)	**44.0**	7.8	**40.1**	**43.3**	7.1	**40.4**	**39.0**	8.5	**36.4**

Table 7. The retrieval performance of SIFT−128D, SIFT−64D, SURF−64D and RC-SIFT-64D when the $Caltech - 2500$, $Caltech - 1000$ and $Caltech - 500$ benchmarks are used. A query image is retrieved if one or more of its related images is obtained in the top ten results.

Method	$Caltech - Buil500$			$Caltech - Buil1000$			$Caltech - Buil2500$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-128D	57.0	11.0	44.9	**53.0**	11.4	**43.6**	**48.5**	12.6	**40.3**
SIFT-64D	51.0	13.0	39.2	47.3	14.1	39.1	41.7	13.2	36.2
SURF-64D	39.7	12.8	33.0	34.2	14.5	26.7	33.8	12.6	23.8
RC-SIFT64D(R)	**58.2**	10.6	**45.6**	**53.0**	10.4	**43.8**	**49.0**	12.3	**40.6**
RC-SIFT64D(C)	**57.7**	10.8	**45.2**	**52.8**	10.6	**43.6**	**49.0**	12.6	**40.3**

Table 8. The retrieval performance of SIFT−128D, SIFT−64D, SURF−64D and RC-SIFT-64D when 500, 1000 and 2500 features are extracted from the Caltech-Buildings benchmark images. The performance is verified in the top fifty retrieved images.

Method	$Caltech - Buil500$			$Caltech - Buil1000$			$Caltech - Buil2500$		
	MR	VR	MAP	MR	VR	MAP	MR	VR	MAP
SIFT-128D	74.0	8.2	47.4	71.5	9.3	**45.3**	66.5	10.4	**43.0**
SIFT-64D	67.0	14.6	42.7	59.8	14.3	43.5	53.0	14.5	37.0
SURF-64D	50.4	14.9	40.3	48.9	14.0	35.3	48.5	14.9	25.6
RC-SIFT64D(R)	**75.0**	7.0	**47.6**	**73.5**	8.9	**46.0**	**67.0**	10.6	**43.1**
RC-SIFT64D(C)	**75.3**	7.0	**48.0**	**73.2**	8.7	44.7	**67.0**	11.0	**43.0**

adding noise and finally, rotation change with adding noise. To achieve this, the first 500 images of each scene of the UKbench [2] benchmark (referred as $UKbench5$) are picked. Afterwards, we convolve the images of the $UKbench5$ benchmark with a combination of different kinds of the image affine transformation. The descriptors are indexed using a vocabulary tree of depth $L = 3$ and initial centers $k = 10$. The similarity is computed using the $L1-$norm. A query image is considered to be retrieved if its corresponding database image appears in the top of the retrieved images.

Table 9. 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 illuminated and rotated benchmarks (generated from $UKbench5$). For each query image we check if its corresponding image in the used benchmark appears as the first retrieved image in the result. The results are presented for two levels of illumination increase (i.e. 50, 120) for each five rotation values are applied: $40^o, 135^o, 215^o, 250^o, 300^o$. The results are presented in percent.

Method	Illumination increase 50					Illumination increase 120				
	40^o	135^o	215^o	250^o	300^o	40^o	135^o	215^o	250^o	300^o
SIFT-128D	**76.2**	**78.0**	**78.0**	**78.0**	**76.0**	**31.0**	29.4	**30.0**	29.4	29.1
SIFT-64D	75.6	76.0	76.2	76.2	75.3	29.6	29.2	29.2	**29.6**	29.0
SURF-64D	75.5	76.9	76.9	76.9	**75.8**	28.9	29.2	29.0	29.1	28.7
RC-SIFT64 (R)	**75.9**	77.7	**77.8**	**78.0**	**76.0**	**31.0**	29.6	**29.6**	**29.6**	**29.3**
RC-SIFT64 (C)	75.8	**77.8**	**77.8**	75.7	75.7	**30.0**	29.2	29.5	29.5	**29.2**

Combination of Illumination and Rotation. To evaluate the robustness of the descriptors with respect to combinations of illumination and rotation changes, the illumination of the $UKbench5$ images is increased using the values $50, 70, 100, 120$ [13,22]. After that, the illuminated images are rotated at different angles in a clockwise direction (i.e. $40^o, 135^o, 215^o, 250^o, 300^o$) to generate 20 benchmarks each of them contains 500. To verify the robustness of the descriptors to illumination decrease and rotation, the values $30, 50, 70, 90$ are subtracted

Table 10. The performance evaluation of SIFT−128D, SIFT−64D, SURF−64D and our RC-SIFT-64D(R)and RC-SIFT-64D(C) in the case of combination of illumination decrease and rotation. For each query image the retrieval task is achieved if its corresponding image in the illuminated and rotated benchmark appears in the top of the result. The results are presented for two levels of illumination decrease (i.e. 30, 90) for each five rotation values are applied: $40^o, 135^o, 215^o, 250^o, 300^o$.

Method	Illumination decrease 30					Illumination decrease 90				
	40^o	135^o	215^o	250^o	300^o	40^o	135^o	215^o	250^o	300^o
SIFT-128D	**79.8**	**75.8**	**76.2**	**78.6**	77.3	**52.8**	49.0	**50.8**	48.8	47.3
SIFT-64D	78.6	**75.6**	**76.0**	**78.6**	77.5	52.6	**49.6**	**51.2**	50.3	47.6
SURF-64D	78.2	74.8	75.8	78.0	77.4	50.7	48.7	50.2	**50.8**	47.7
RC-SIFT64 (R)	**79.5**	**75.8**	**76.0**	**78.8**	**78.0**	**53.1**	**49.6**	**51.2**	**51.2**	**49.3**
RC-SIFT64 (C)	79.2	75.5	75.8	78.0	**77.6**	**52.8**	**49.2**	50.3	50.6	**48.7**

from all channels of the pixels of each image after that the same previous rotation angles are applied to generate 20 benchmarks too. Tables 9 and 10 show robust performance for all used rotation angles when small amount of illumination change is applied to images. However, these tables present a decrease of performance for all rotation angles when the illumination change increase. From these results we deduce that the increasing values of combination affect negatively the stability of the extracted descriptors. Moreover, the comparison of these results with the results of applying the illumination or the rotation change separately [22] clarify that the performance of all proposed descriptors decrease when the rotation and illumination changes are combined (for rotation change the performance is more than 92% and up to 100% for illumination change [2,22]).

Table 11. The performance of SIFT−128D, SIFT−64D, SURF−64D and our RC-SIFT−64D when a combination of salt and pepper noise and illumination increase is applied on $UKbench5$ images. The results are presented for two level of noise densities (i.e. 15% and 35%), for each the lightness increases using the values: $Li = 50$ and $Li = 120$). A query image is retrieved if its corresponding image in the used noised and illuminated benchmark appears in the top of the retrieved image. In this table SP and Li refer to the salt pepper noise and lightness change, respectively.

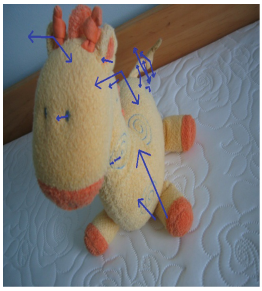
Method	Li 50	SP 15%	SP 15%		Li 120	SP 35%	SP 15%	
			Li 50	Li 120			Li 50	Li 120
SIFT-128D	**100**	82.6	**68.0**	32.8	**91.2**	20.2	**5.8**	3.0
SIFT-64D	**99.8**	82.2	**67.6**	32.6	**90.2**	15.2	5.0	**3.1**
SURF-64D	99.5	81.2	67.2	32.3	88.1	14.5	4.7	2.8
RC-SIFT-64D(R)	99.5	**83.4**	**67.6**	**34.7**	**90.2**	**20.8**	**6.2**	**3.4**
RC-SIFT-64D(C)	97.0	**83.1**	67.1	**34.2**	**90.2**	**20.5**	**5.8**	**3.1**

Table 12. The performance of SIFT−128D, SIFT−64D, SURF−64D and our RC-SIFT−64D using when a combination of salt and pepper noise and increasing darkness is applied on $UKbench5$ images. The results are presented for two level of noise densities (i.e. 15% and 35%), for each the darkness increases using the values $Dr = 50$ and $Dr = 120$. A query image is retrieved if its corresponding image in the used noised and illuminated benchmark appears in the top of the retrieved image. In this table SP and $Dr =$ refer to the salt pepper noise and lightness, respectively.

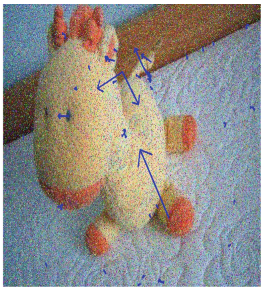
Method	Dr 30	SP 15%	SP 15%		Dr 90	SP 35%	SP 15%	
			Dr 30	Dr 90			Dr 30	Dr 90
SIFT-128D	**100**	82.6	**73.6**	**36.7**	**91.2**	20.2	8.0	6.2
SIFT-64D	**99.8**	82.2	73.0	**36.7**	**90.2**	15.2	8.0	5.8
SURF-64D	99.5	81.2	73.0	35.3	88.1	14.5	7.1	5.3
RC-SIFT-64D(R)	99.5	**83.4**	**73.4**	**37.0**	**90.2**	**20.8**	**10.2**	**7.0**
RC-SIFT-64D(C)	97.0	**83.1**	73.1	36.6	**90.2**	**20.5**	**9.6**	**6.8**

Combination of Noise and Illumination Change. To test the robustness of the proposed descriptors to the illumination change and added noise, salt and pepper noise with density of 15% and 35% is applied to $UKbench5$ images. After that, the brightness of noised images is increased using the values $50, 70, 100, 120$ [13,22] or decreased by subtracting the values $30, 50, 70, 90$ from all channels of the pixels of the image. As a result we obtain 16 benchmarks which contains various levels of additional noise and illumination change. Tables 11 and 12 present the performance of various descriptors in two cases: Firstly when adding noise and illumination change are applied separately and secondly in case of combination. These tables show a decrease in the performance of all presented descriptors in the case of combination. Moreover, the performance in the case of combination is always lower than the minimum performance obtained by applying the affine transformation separately. However, in case of using the salt and pepper noise with density of 35% all presented descriptors are not stable anymore. Figure 2 presents the difference between the locations of the extracted descriptors by the RC-SIFT−64D before and after applying a combination of illumination increase and noise. The most extracted features in Fig. 2(a) and (b) locate at comparable positions in the both images. Whereas, the locations of features in Fig. 2(a) differ from the those in Fig. 2(b). Therefore, the image in Fig. 2(b) appears as the first retrieved results of the query image Fig. 2(a) whereas, the image in Fig. 2(c) does not appear in the top retrieved results of the query image Fig. 2(a).

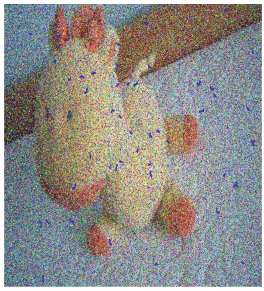
Addition of Noise and Rotation. A combination of adding noise and rotation is achieved by firstly adding the salt and pepper noise with density of 15% or 35% to the $UKbench5$ benchmark (the detail of adding noise is described in [22]). Secondly, the noised images are rotated at different angles in a clockwise direction (i.e. $40^o, 135^o, 215^o, 250^o, 300^o$) to generate ten benchmarks of



(a) An image of the $UKbench5$ benchmark (i.e. the query image).



(b) The image in 2(a) after applying a combination of illumination increase with value 50 and noise of density 15%.



(c) The image in 2(a) after applying a combination of illumination increase with value 50 and noise of density 35%.

Fig. 2. It is shown in (a) and (b), that many extracted descriptors (presented in blue) have the same locations in both images. Therefore, the image retrieval task is achieved successfully in this case. Whereas, the extracted descriptors in (a) and (c) are located in different positions thus, the benchmark image (c) does not appear in the top of the retrieved results. (Color figure online)

Table 13. The performance comparison of SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and our RC-SIFT-$64D$(R)and RC-SIFT-$64D$(C) in the case of applying salt and pepper noise and rotation to $UKbench5$ benchmark. For each query image the retrieval task is achieved if its corresponding image in the noised and rotated benchmark appears in the top of the result. The results are presented for two noise densities (i.e. 15%, 35%) for each five rotation values are applied: $40^o, 135^o, 215^o, 250^o, 300^o$. The results are presented in percent.

Method	Noise 15%					Noise 35%				
	40^o	135^o	215^o	250^o	300^o	40^o	135^o	215^o	250^o	300^o
SIFT-128D	**32.0**	**39.8**	**40.0**	**43.0**	37.6	**6.2**	**5.6**	5.4	**5.8**	5.4
SIFT-64D	**31.8**	**39.5**	**40.0**	42.0	36.8	**6.1**	**5.6**	5.6	5.2	5.0
SURF-64D	30.7	39.2	40.2	42.3	37.1	5.8	5.5	**5.9**	**5.8**	5.4
RC-SIFT64 (R)	31.7	39.2	**41.2**	**43.8**	**40.0**	6.0	**6.0**	5.5	**6.2**	**6.0**
RC-SIFT64 (C)	**32.0**	39.3	**41.2**	**43.0**	**40.0**	5.8	5.3	**5.7**	**6.2**	**6.2**

noised rotated images. Table 13 describes how the performance of all proposed descriptors decrease very strongly for a fixed rotation angle when the density of the added noise is increased.

6.4 Combination of Blurring and Affine Transformation

To study the effect of image blurring combination with various kinds of affine transformation on the performance of the original SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and RC-SIFT$-64D$, the $UKbench5$ benchmark images are firstly blurred by convolving the image with Gaussian filters using three variations i.e., $\sigma^2 = 5$, $\sigma^2 = 10$ and $\sigma^2 = 15$ (the process of fileting is described in [22]). After

that, the illumination of the blurred images is increased or decreased using the same values presented in Subsect. 6.3. The best Performance of near-duplicate retrieval is obtained when the Gaussian filter with $\sigma^2 = 5$ is used. This performance is below 25% for all proposed descriptors. However, when a Gaussian filter with $\sigma^2 = 10$ or $\sigma^2 = 15$ is applied the performance decreases to be not more than 16% or 13%, respectively. When a combination of image blurring and rotation change (the rotation values are: $40^o, 135^o, 215^o, 250^o, 300^o$) is applied, the successfully retrieved images are not more than 13% for all descriptors when $\sigma^2 = 5$. Whereas, the performance decreases to 8% or 4% when $\sigma^2 = 10$ or $\sigma^2 = 15$, respectively. A Combination of the Gaussian blur with the salt pepper noise retrieve successfully less than 15% of the applied query images when the density of noise is 15% and the blurring variation is $\sigma^2 = 5$. The number of retrieved images decreases to 10% or 8% when the blurring variation increases to $\sigma^2 = 10$ or $\sigma^2 = 15$, respectively. The results of combining Gaussian blur with different kinds of image affine transformations present that the performance of all proposed descriptors decreases strongly and the extracted descriptors become unstable when more blurring is added to the images.

7 Conclusion

In this work, we evaluated the performance of the RC-SIFT$-64D$ descriptor to solve the near-duplicate retrieval task in two cases: Firstly, for benchmarks of different size. Secondly, using the same benchmark but for different numbers of extracted features. The experiments show a slight improvement in matching results compared to the original SIFT$-128D$ when tested on various benchmark databases. Moreover, the RC-SIFT$-64D$ needs shorter time for indexing and less memory.

We also evaluate the robustness and stability of the original SIFT$-128D$, SIFT$-64D$, SURF$-64D$ and RC-SIFT$-64D$ against combinations of image affine transformations. The results show that all proposed descriptors are robust for combination of transformations with small changes. However, the stability of descriptors decreases when the amount of the combined transformations increases especially, in the case of combination with noise. When the image affine transformations are combined with blurring, the performance of all proposed descriptors decreases very strongly. So that in this case the extracted descriptors loose their robustness.

In the case of extracting variable amounts of features from the benchmark, the performance increase when the numbers of extracted features decrease. Therefore, we are going in the next step to study the factors that may help to reduce the amount of detected features but enhance the performance of descriptors in solving the near-duplicate retrieval task. Moreover, we aim to study if 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 [18].

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. J. Comput. Vis. **60**, 91–110 (2004)
2. Nistèr, D., Stewènius, H.: Scalable recognition with a vocabulary tree. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2161–2168 (2006)
3. Jiang, M., Zhang, S., Li, H., Metaxas, D.N.: Computer-aided diagnosis of mammographic masses using scalable image retrieval. IEEE Trans. Biomed. Eng. **62**, 783–792 (2015)
4. Jianchao, Y., Kai, Y., Yihong, G., Thomas, H.: Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
5. Zhang, C., Wang, S., Huang, Q., Liu, J., Liang, C., Tian, Q.: Image classification using spatial pyramid robust sparse coding. Pattern Recogn. Lett. **34**, 1046–1052 (2013)
6. Zhang, D.Q., Chang, S.F.: Detecting image near-duplicate by stochastic attribute relational graph matching with learning. In: Proceedings of the 12th Annual ACM International Conference on Multimedia (2004)
7. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR), pp. 549–556 (2007)
8. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-Hash and tf-idf weighting. In: British Machine Vision Conference, pp. 50.1–50.10 (2008)
9. Auclair, A., Vincent, N., Cohen, L.D.: Hash functions for near duplicate image retrieval. In: Applications of Computer Vision (WACV), pp. 1–6 (2009)
10. Xu, D., Cham, T., Yan, S., Duan, L., Chang, S.: Near duplicate identification with spatially aligned pyramid matching. IEEE Trans. Circ. Syst. Video Technol. **20**, 1068–1079 (2010)
11. Chu, L., Jiang, S., Wang, S., Zhang, Y., Huang, Q.: Robust spatial consistency graph model for partial duplicate image retrieval. IEEE Trans. Multimedia **15**, 1982–1996 (2010)
12. Steinbach, M., Ertoz, L., Kumar, V.: The challenges of clustering high dimensional data. In: Wille, L.T. (ed.) New Vistas in Statistical Physics-Applications in Econophysics, Bioinformatics, and Pattern Recognition, pp. 273–309. Springer, Heidelberg (2004)
13. Khan, N.Y., McCane, B., Wyvill, G.: SIFT and SURF performance evaluation against various image deformations on benchmark dataset. In: Digital Image Computing Techniques and Applications (DICTA), pp. 501–506 (2011)
14. Grauman, K., Darrell, T.: The pyramid match kernel: efficient learning with sets of features. J. Mach. Learn. Res. **8**, 725–760 (2007)
15. Grauman, K., Darrell, T.: Pyramid match kernels: discriminative classification with sets of image features. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 1458–1465 (2005)
16. Yang, Y., Newsam, S.: Comparing SIFT descriptors and Gabor texture features for classification of remote sensed imagery. In: Proceedings of the 15th IEEE on Image Processing, pp. 1852–1855 (2008)
17. Li, J., Qian, X., Li, Q., Zhao, Y., Wang, L., Tang, Y.Y.: Mining near duplicate image groups. Multimedia Tools Appl. **74**, 655–669 (2014). Springer Science and Business Media, New York

18. Steffen, J., Christian, H., Ahmad Alyosef, A., Tönnies, K., Nürnberger, A.: Rotational invariance at fixation points - experiments using human gaze data. In: Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods, pp. 451–456 (2012)
19. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). doi:10.1007/11744023_32
20. Jègou, H., Douze, M., Schmid, C., Pèrez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 3304–3311 (2010)
21. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Computer Vision and Pattern Recognition, no. 2, pp. 506–513 (2004)
22. Ahmad Alyosef, A., Nürnberger, A.: Adapted SIFT descriptor for improved near duplicate retrieval. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods, pp. 55–64 (2016)
23. Manning, C.D., Raghavan, P., Schütze, H.: Chapter 8: evaluation in information retrieval. Part of Introduction to Information Retrieval, pp. 151–175 (2009)
24. Aly, M., Welinder, P., Munich, M., Perona, P.: Towards automated large scale discovery of image families. In: Computer Vision and Pattern Recognition Second IEEE Workshop (CVPR), pp. 9–16 (2009)
25. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1615–1630 (2005)
26. Foo, J.J., Sinha, R.: Pruning SIFT for scalable near-duplicate image matching. In: Proceedings of the Eighteenth Conference on Australasian Database, pp. 63–71 (2007)
27. Jègou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European Conference on Computer Vision, pp. 301–317 (2008)
28. Nistèr, D., Stewènius, H.: Recognition Benchmark Images. http://www.vis.uky.edu/~stewe/ukbench/
29. Aly, M., Welinder, P., Munich, M., Perona, P.: Caltech-Buildings Benchmark. http://www.vision.caltech.edu/malaa/datasets/caltech-buildings/