The Effect of SIFT Features Properties in Descriptors Matching for Near-duplicate Retrieval Tasks

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*

Keywords: SIFT Descriptor, RC-SIFT 64*D*, Feature Truncating, Properties of the SIFT Features, Image Near Duplicate Retrieval.

Abstract: The scale invariant feature transformation algorithm (SIFT) has been widely used for near-duplicate retrieval tasks. Most studies and evaluations published so far focused on increasing retrieval accuracy by improving descriptor properties and similarity measures. Contrast, scale and orientation properties of the SIFT features were used in computing the SIFT descriptor, but their explicit influence in the feature matching step was not studied. Moreover, it has not been studied yet how to specify an appropriate criterion to extract (almost) the same number of SIFT features (respectively keypoints) of all images in a database. In this work, we study the effects of contrast and scale properties of SIFT features when ranking and truncating the extracted descriptors. In addition, we evaluate if scale, contrast and orientation features can be used to bias the descriptor matching scores, i.e., if the keypoints are quite similar in these features, we enforce a higher similarity in descriptor matching. We provide results of a benchmark data study using the proposed modifications in the original SIFT$-128D$ and on the region compressed SIFT (RC-SIFT$-64D$) descriptors. The results indicate that using contrast and orientation features to bias feature matching can improve near-duplicate retrieval performance.

1 INTRODUCTION

The scale invariant transformation algorithm (SIFT) is one of the most used feature extraction algorithms in various research to recognize similar objects, classify images, retrieve relevant images from an image database and more specifically to solve near-duplicate retrieval (NDR) tasks. The importance of the SIFT algorithm comes from the invariance of its features against various kind of image affine transformation and their robustness to viewpoint change, blurring and scale change.

The extraction of the SIFT features as described in (Lowe, 2004) results in huge amount of descriptors that are required to represent a set of images. These descriptors are high dimensional vectors (each vector contains 128 elements (Lowe, 2004)). Using such high dimensional vectors and large scale sets of images impose strong demands on memory and computing power in order to support near-duplicate retrieval tasks. Therefore, methods have been proposed, see e.g. (Khan et al., 2011) and (Alyosef and Nürnberger, 2016) to reduce the dimensionality of SIFT descriptors. This reduction decreases the time of processing and the usage of memory when SIFT features are indexed and matched. The region compressed SIFT (RC-SIFT) descriptors (Alyosef and Nürnberger, 2016) are also invariant to affine transformation change and perform as robust as the original SIFT features to viewpoint change, scale change and blurring change (Alyosef and Nürnberger, 2016). However, we still have two major issues to discuss with respect to the original SIFT and the RC-SIFT features. The first issue is determining an appropriate method to truncate the number of extracted SIFT features from a set of images. This issue is important because the number of extracted SIFT features of an image database is not stable. Moreover, there is no rule to determine the accepted SIFT features when only a specific number of feature is required in a study. Therefore, we suggest in this work to use the scale and contrast properties of the SIFT features to rank the extracted features and then truncate the appropriated number of accepted features based on these properties. The second issue is the matching process of descriptors of two images. The standard method of comparing the SIFT features is to compute the distance between their descriptors i.e. the smaller the

Alyosef, A. and Nürnberger, A.
The Effect of SIFT Features Properties in Descriptors Matching for Near-duplicate Retrieval Tasks.
DOI: 10.5220/0006250607030710
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 703-710
ISBN: 978-989-758-222-6

distance between descriptors is the greater is the similarity between them. The other information like locations, orientations, scales and contrasts of features are not used in the matching step. Therefore, we suggest a method to involve the scale, contrast and orientation properties in the matching step to determine which of them play an important role in solving near-duplicate retrieval tasks. We perform a benchmark study using the original SIFT$-128D$ features (Lowe, 2004) and the region compressed SIFT$-64D$ features (Alyosef and Nürnberger, 2016) to determine whether the influence of these properties is equivalent for both of SIFT$-128D$ and RC-SIFT$-64D$ features.

The remainder of this paper is organized as follows. Section 2 gives an overview of prior work related with the SIFT and the RC-SIFT algorithms and image NDR algorithms. Section 3 details the proposed method to truncate the SIFT features and involve the scale, contrast and orientation properties in matching process. Section 4 presents the evaluation measures and the settings of our experiments. Section 5 discusses the results of experiments. Finally, Section 6 draws conclusions of this work and discusses possible future work.

2 RELATED WORK

The SIFT descriptor has been widely used in various fields of images retrieval, image near-duplicate retrieval (Auclair et al., 2006), (Chum et al., 2008), image classification (Nistèr and Stewènius, 2006b) and object recognition (Jiang et al., 2015) due to its robustness against different kinds of image transformation, viewpoint change, blurring and scale change. To overcome the problem of extracting high dimensional SIFT descriptors (the length of the original SIFT descriptor is 128) various methods have been suggested. A method to reduce the dimensionality of SIFT descriptor to $96D$, $64D$ or $32D$ is described in (Khan et al., 2011). This reduction is done by skipping the outside edges of the region around the keypoints to get $96D$ descriptor and then averaging the outside regions to obtain $64D$ descriptor. The reduced SIFT descriptors of forms SIFT$-96D$, $64D$ have shown robust performance against image affine transformation, viewpoint change and scale change. In (Ke and Sukthankar, 2004), Principle Component Analysis is employed to obtain $64D$ SIFT descriptors. This approach is in need of an off-line training stage to compute the eigenvalue vector for each image databases separately. In (Alyosef and Nürnberger, 2016), a method is proposed to compress the SIFT descriptor to get RC-SIFT$64D$, $32D$ or $16D$ descriptors.

An important issue when SIFT features are extracted of an image database is that the amount of the extracted features of various images is not invariant. This issue is addressed in (Foo and Sinha, 2007) and it has been solved by ranking the extracted SIFT features based on their decreased contrast values. After that, the list of features is pruned based on a specific number which determines the required number of features, so that, features which have low contrast are skipped. To accelerate the matching process many methods has been suggested to find the relation between the extracted features such as building a dictionary of features based on direct clustering as described in (Li et al., 2014), (Yang and Newsam, 2008), (Grauman and Darrell, 2005) and (Grauman and Darrell, 2007) or based hierarchical $k-$means clustering as described in (Jiang et al., 2015), (Nistèr and Stewènius, 2006b). Hashing functions are used in (Chum et al., 2008) and (Auclair et al., 2006) to reduce the amount of comparisons between the features of various images. In (Y. Jianchao and Thomas, 2009) and (Zhang et al., 2013) the sparse coding concept is used in a further step after applying feature clustering to accelerate the matching process and to improve the matching results. In the next section, we explain our suggested steps to involve the properties of SIFT features in the matching process.

3 IMAGE NEAR-DUPLICATE RETRIEVAL UNDER THE IMPACT OF FEATURE PROPERTIES

To explain the effect of involving the scale, contrast and orientation properties in solving the near-duplicate retrieval task, we give a short description of the way of feature extraction in both, original SIFT and RC-SIFT algorithms. After that, we describe our idea to truncate the list of SIFT and RC-SIFT features. Finally, we explain our suggested method to employ the scale, contrast and rotation in matching process.

3.1 The Concept of the SIFT$-128D$ & the RC-SIFT Detectors

The extraction of the original SIFT$-128D$ and the RC-SIFT features has the same initial steps. These steps begin by building the image space scale or a so called "image pyramid". This image pyramid contains octaves downsampled and scaled copies of an input image. Based on this pyramid the difference of Gaussian (DoF) pyramid is constructed. The minima and maxima locations are determined in the DoF images which they present the candidate keypoints. After that, the invariance of the candidates is verified by computing their contrast and the candidates that their contrast is lower than specific threshold are rejected. The scales of the keypoints are computed to determine their positions in the image pyramid. Afterwards, the dominant orientation of each keypoint is computed and used in the step of descriptor computation. In the case of obtaining more than one dominant orientation of the same keypoint new keypoints are created in the same location but with different orientations and different descriptors. In the original SIFT algorithm a descriptor of 128 elements is constructed by considering that a keypoint can take any place in a grid of 4×4 dimensions, for each dimension 8 different orientation are assigned. Whereas, the descriptor of the RC-SIFT$-64D$ is computed based on the suggestion that for each two possible shifting of the keypoint in the horizontal direction only one vertical shifting is possible (for more details see (Alyosef and Nürnberger, 2016)). In both of the SIFT$-128D$ and the RC-SIFT$-64D$ each keypoint is presented with a feature which contains the location of keypoint, its scale and orientation properties and its descriptor vector.

Based on the way of computing the SIFT and the RC-SIFT features, we find that the scale, contrast and orientation factors play important roles in extracting, localizing and describing the features but they are not considered when the features of two images are matched together. Moreover, these factors are not considered to overcome the problem of extracting various numbers of features of different images.

In the following section we explain our idea to truncate the accepted features based on either the scale or the contrast properties to get almost the same number of features for all images in a database. In addition, we explain our method of involving scale, orientation and contrast in matching step to solve the near-duplicate retrieval tasks.

3.2 Truncate the List of Features based on Their Properties

The number of extracted features is not well defined by a formal rule, neither in the the original SIFT$-128D$ nor the RC-SIFT$-64D$. Therefore, we suggest in this work to compute and store scale and contrast properties for all extracted features for this purpose. In order to study the effect of different truncation, we order the features based on either decreasing contrast or decreasing scale (depending on the goal of the experiment (see Section 5)) and then we truncate the list of features using a predefined initial number of accepted features NF. We do not use the orientation property in this step because this property does not give any information about the robustness of features (like the contrast property) or where the features are found (like the scale property). In the final stage the dominant orientation of each feature is computed as described in Subsection 3.1 and new keypoints are created in the locations where more dominant orientations are found. So that, after applying these steps the number of the extracted features can be defined to be lesser than $NF + \varepsilon$ where ε denote the number of the new created features because of the dominant orientation.

3.3 Involve Feature Properties in Feature Matching Step

As discussed in Section 2, the standard matching of the original SIFT features and the RC-SIFT features is achieved by comparing only the descriptors i.e. the scale, contrast and orientation properties of features are ignored. In this study we analyze the effect of using these properties in the matching process i.e. we suggest that features that have very similar scale, contrast and orientation properties should be considered to be more similar than features that have quite different properties. The reason of this idea is that the properties of robustness, scale and directions of each region present important information which may improve the matching of two regions. Therefore, we analyzed this idea to determine whether these properties improve the performance of near-duplicate retrieval tasks and whether their influence are equivalent. To achieve this we start by extracting the SIFT$-128D$ and RC-SIFT$-64D$ features. These features are structured using hierarchical $k-$means clustering as described in (Alyosef and Nürnberger, 2016). Based on the hierarchical clustering a bag of words is constructed and employed to represent images in terms of vectors (see also (Jiang et al., 2015) and (Nistèr and Stewènius, 2006b)). To compare a query image with a database image the following steps are carried out:

- Weights definition: In this step weights related to contrast W_{cont}, scale W_{scl} and orientation W_{ori} properties are defined. These weights are necessary to involve the influence of various properties when descriptors are compared. In this work we define all used weights in terms of unique value W i.e. contrast, scale and orientation are given the same degree of importance.
- Properties criteria for matching: The weights

W_{ori}, W_{cont} and W_{scl} are given values in the range $]0,1[$ if the following relations are satisfied:

$$\Delta Ori = \left|Ori(f_q) - Ori(f_{db})\right| \leq thr_{ori} \quad (1)$$

$$\Delta Cont = \left|Cont(f_q) - Cont(f_{db})\right| \leq thr_{cont} \quad (2)$$

$$\Delta Scl = \left|log(Scl(f_q)) - log(Scl(f_{db}))\right| \leq thr_{scl} \quad (3)$$

where Ori, $Cont$ and Scl denote to orientation, contrast and scale respectively. thr_{ori}, thr_{cont} and thr_{scl} symbols refer to thresholds related with orientation, contrast and scale respectively. The values of these thresholds are determined heuristically.

- Features matching: For each query image vector $v(q)$ and database vector $v(db)$, if the elements $v_i(q)$ and $v_i(db)$ satisfy that $v_i(q) > 0$ and $v_i(db) > 0$ then the distance between of them is computed as the average of the following three distances:

$$d_{ori}(v_i(q), v_i(db)) = W_{ori}\left|v_i(q) - v_i(db)\right| \quad (4)$$

$$d_{cont}(v_i(q), v_i(db)) = W_{cont}\left|v_i(q) - v_i(db)\right| \quad (5)$$

$$d_{scl}(v_i(q), v_i(db)) = W_{scl}\left|v_i(q) - v_i(db)\right| \quad (6)$$

where the value of one is assigned to W_{ori}, W_{cont} or W_{scl} if the relations 1, 2 or 3 are not satisfied.

- Image matching: Depending on the previous steps the distance between a query vector $v(q)$ and a database vector $v(db)$ is computed as:

$$d(v(q), v(db)) = \frac{1}{N_q N_{db}} \sum Average(d_{ori}, d_{cont}, d_{scl}) \quad (7)$$

where N_q and N_{db} present the number of extracted features of the query and database image respectively.

The steps of feature truncation and involving properties in the matching step are clarified in flowchart as given in Figure 1. Based on these steps the scale, orientation and contrast properties are involved in the matching process. In the following section we discuss results of our study on the influence of using these properties to solve the near-duplicate retrieval task.

4 EVALUATION

The performance of the SIFT$-128D$ and the RC-SIFT$-64D$ features is studied to solve the image near-duplicate retrieval task when the scale, contrast and orientation properties are involved in feature selection and feature matching steps. To achieve this, two different image databases are used of different sizes and resolutions. In the following we describe the evaluation measures and the image databases.

4.1 Evaluation Measures

To evaluate the effect of involving the properties of features (i.e. scale, contrast and orientation) in matching process to solve the near-duplicate retrieval task, we extract the original SIFT$-128D$ and the RC-SIFT$-64D$ features of images. After that, we rank and truncate the list of features based on the contrast or scale properties. Afterwards, the descriptors are indexed and the vectors of images are constructed using the hierarchical k-mean clustering. The similarity between a query vectors $v(q)$ and database vectors $v(db)$ is computed by applying the relation 7. In case of involving the properties separately the relation 7 becomes:

$$d(v(q), v(db)) = \frac{1}{N_q N_{db}} \sum d_p \quad (8)$$

where d_p is d_{ori}, d_{cont} or d_{scl}. The results are evaluated by computing the mean recall value as follows:

$$MR = \frac{1}{Q} \sum_{q=1}^{Q} Recall(q) \quad (9)$$

where Q is the total number of query images and $Recall(q)$ is the recall related with a query image q and is defined as:

$$Recall = \frac{N_{qr}}{N_q} \quad (10)$$

where 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.

To measure how the results of individual query images differ from the mean recall, we compute the variance of the recall values VR as:

$$VR = \frac{1}{Q} \sum_{q=1}^{Q} (Recall(q) - MR)^2 \quad (11)$$

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 and it is defined as:

$$MAP = \sum_{q=1}^{Q} \frac{Ap(q)}{Q} \quad (12)$$

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

$$AP(q) = \frac{1}{n} \sum_{i=1}^{n} p(i) \times r(i) \quad (13)$$

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

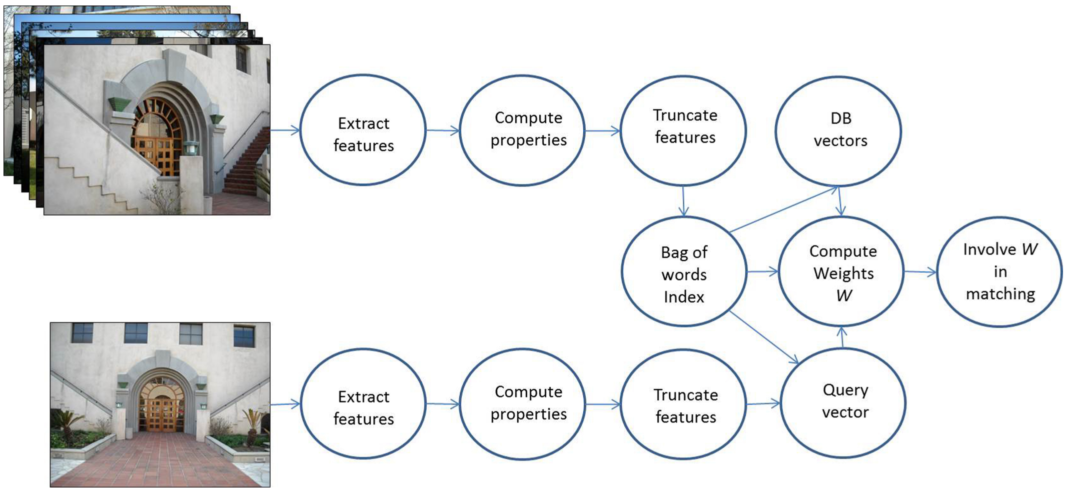

Figure 1: The flowchart of feature truncating and matching when the properties of features are employed.

4.2 Benchmark Sets

To study the influence of involving the scale, contrast and orientation properties of the SIFT features in feature selection and matching steps we choose two image databases that have been used in the state of art studies. These image databases contain indoor/ outdoor images of various scenes in groups of four or five images for each scene. The images of each scene differ in view point, scale, lightness or combination of more than one of these conditions. The first image database is the Caltech-Buildings (Aly et al., 2011) which contains 250 images for 50 different buildings around the Caltech campus. The images of this database have high resolution (the resolution of each image is 2048×1536 pixels). The second image database is UKbench (Nistèr and Stewènius, 2006b) (this database can be download from (Nistèr and Stewènius, 2006a). This image database contains about $10,000$ images of resolution 640×480 pixels. We apply our study on two different image databases to verify whether the content and properties of images affect the results of study.

Table 1: The retrieval performance of SIFT$-128D$ when the lists of features are ranked and truncate based on their *scale property*. The mean recall is computed based on the top four ($MR4$) and then top ten ($MR10$) retrieved images of the *Caltech-Buildings* database. The mean recall is calculated as given in relation 9.

Descriptors properties			SIFT$-128D$	
Scale	Contrast	Orientation	MR4	MR10
			40.02	49.5
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	35.0	47.0
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**41.0**	**52.50**
		$\Delta Ori<\pi/8$	**42.50**	**54.0**

5 RESULT AND ANALYSIS

The results of the SIFT$-128D$ and the SIFT$-64D$ algorithms are evaluated using the Caltech-Buildings and UKbench databases in two cases. Firstly when the extracted lists of features are ranked and truncated depending on the scale property. Secondly, when they are ranked and truncated based on the contrast property. In the empirical study we notice that the sets of extracted features in both cases are not equivalent when we suggest to consider only the top NF extracted features i.e. the position of features in the ranked list differ when the used property for ranking differs. Moreover, we notice that the new created features after using the dominant orientation (see Subsection 3.2) is $\varepsilon \leq \frac{NF}{3}$ so that, the total number of extracted feature is not more than $NF + fracNF3$. We determine the value of NF depending on the resolution of images and using a region adaptive approach.

For the Caltech-Buildings database, due to the high resolution of images of this benchmark (Aly et al., 2011), huge amount of features may extracted

Table 2: The performance of RC-SIFT$-64D$ using *Caltech-Buildings* database when the features are ranked and truncate based on their *scale property*. The used symbols are explained in Table 1.

Descriptors properties			RC-SIFT-64	
Scale	Contrast	Orientation	MR4	MR10
			40.70	50.06
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	35.0	47.50
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**41.60**	**53.72**
		$\Delta Ori<\pi/8$	**43.0**	**55.10**

Table 3: The *mean average of precision* (Eq. 12) and the *variance of recall* (Eq. 11) of SIFT$-128D$ when the lists of features are ranked and truncate based on their *scale property*. The *MAP* and *VR* are computed based on the top four retrieved images of the *Caltech-Buildings* database.

Descriptors properties			SIFT$-128D$	
Scale	Contrast	Orientation	MAP	VR
			37.50	9.47
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	32.12	8.50
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**37.75**	**9.19**
		$\Delta Ori<\pi/8$	**38.62**	**8.81**

Table 4: The *mean average of precision* and the *variance of recall* of RC-SIFT$-64D$ when the lists of features are ranked and truncate based on their *scale property* using *Caltech-Buildings* database. The used symbols are explained in Table 3.

Descriptors properties			RC-SIFT-64	
Scale	Contrast	Orientation	MAP	VR
			37.97	9.49
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	32.08	8.6
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**38.34**	**9.12**
		$\Delta Ori<\pi/8$	**39.56**	**8.23**

of images therefore, we determine $NF = 1600$ to be the number of accepted features. In case of ranking the features based on their decreasing scale, Tables 1 and 2 present the mean recall of the SIFT-128 and the RC-SIFT-64 algorithms receptively. These tables show that the best performance of the both SIFT-128 and RC-SIFT-64 is achieved when we consider the orientation property and ignore the scale and contrast properties. The worst results are obtained when both scale and contrast are involved in matching process. For the orientation, we determine the orientation threshold to be $thr_{ori} \leq \frac{\pi}{8}$ but we check this threshold in different direction to consider the possible viewpoint changes. For the scale and contrast properties we test different values for the thr_{scl} and thr_{cont} and the best performance for the SIFT-128 and the RC-SIFT-64 is found when $thr_{scl} \leq 0.1$ and $thr_{cont} \leq 0.1$. In case of satisfying one of the relations 1, 2 or 3 the value $W = 0.9$ is assigned to the corresponding weight. We test another values for the weights in the range $]0,1[$ but we got the best performance when the value 0.9 is used. Tables 3 and 4 present the mean average of precision and the variance of recall of the SIFT-128 and the RC-SIFT-64 respectively. They show that the best mean average of precision is obtained when the best mean recall is obtained too. Tables 1, 2, 3 and 4 describe how the variance of recall decreases when the mean of recall increases.

The resolution of images in the UKbench (Nistèr and Stewènius, 2006a) database is not high (it is only 640×480) therefore, we determine the number of extracted features per images as $NF = 500$. The SIFT-128 and the RC-SIFT-64 features are extracted. After that, they are ranked and truncated based on the decreasing value of the scale property. Tables 5 and 6 present the performance of the SIFT-128 and the RC-SIFT-64 descriptors when the UKbench database is used. They explain that

Table 5: The performance (Eq. 9) of SIFT$-128D$ when the lists of features are ranked and truncate based on their *scale property*. The mean recall is computed based on the top three ($MR3$) and then top ten ($MR10$) retrieved images of the UKbench database.

Descriptors properties			SIFT$-128D$	
Scale	Contrast	Orientation	MR3	MR10
			49.30	58.70
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	44.03	53.0
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**50.35**	**59.82**
		$\Delta Ori<\pi/8$	**52.20**	**63.0**

the best mean recall for the SIFT-128 and the RC-SIFT-64 descriptors is obtained when the scale and contrast properties are skipped. We do not present the mean average of precision and variance of recall for this database because they are equivalent to the results presented in Tables 3 and 4.

When the list of features are ranked and truncated based the contrast properties, Tables 7, 8 present the results for both SIFT-128 and RC-SIFT-64 when the Caltech-Buildings database is used. The best performance is obtained when the scale and orientation properties are skipped and then when only the scale property is skipped. Equivalent results are obtained for the UKbench database when the list of features are ranked based on the contrast property.

The previous results explain that, in case of ranking the features based on the scale property the best performance is achieved when both of scale and contrast properties are ignored in matching process. Whereas, in case of ranking the features based on the contrast property the best performance is achieved when the scale and orientation properties are skipped and only the contrast property is involved in the matching process.

Table 6: The performance of RC-SIFT$-64D$ using *UKbench* image database when the lists of features are ranked and truncate based on their *scale property*. The used symbols are explained in Table 5.

Descriptors properties			RC-SIFT-64	
Scale	Contrast	Orientation	MR3	MR10
			50.70	60.70
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	46.0	55.05
		$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	**52.40**	**63.0**
		$\Delta Ori<\pi/8$	**54.8**	**66.38**

Table 7: The retrieval performance of SIFT$-128D$ when the *Caltech-Buildings* database is used. The lists of features are ranked and based on their *contrast property*. The used symbols are explained in Table 1.

Descriptors properties			SIFT$-128D$	
Scale	Contrast	Orientation	MR4	MR10
			37.0	48.5
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori<5\pi/8$ or $\pi<\Delta Ori<9\pi/8$	36.5	46.0
	$\Delta Cont<0.1$		**37.5**	**47.5**
		$\Delta Ori<\pi/8$	**37.0**	**48.5**

6 CONCLUSION

In this work, we studied the role of the scale, contrast and orientation properties of the original SIFT and the RC-SIFT features in solving two issues. The first one is how to determine the set of accepted extracted features of each image in an image database when (almost) fixed number of features is required. We achieved this by ranking and truncating the obtained lists of features based on their decreasing scale or contrast properties. The number of accepted features depends on the resolution of images in a database and is determined using a region adaptive approach. In addition, we found out that dissimilar sets of features are extracted from the same set of images when the ranking and truncation criteria differ. Based on these sets of features we studied the second issue that is whether involving of the scale, contrast and orientation prop-

Table 8: The results of RC-SIFT$-64D$ when the *Caltech-Buildings* database is used. The the lists features are ranked based on the *contrast property*. The used symbols are explained in Table 1.

Descriptors properties			SIFT$-128D$	
Scale	Contrast	Orientation	MR4	MR10
			37.8	49.0
$\Delta Scl<0.1$	$\Delta Cont<0.1$	$\Delta Ori<\pi/8$ or $\pi/2<\Delta Ori$ $<5\pi/8$ or/8 $\pi<\Delta Ori$ $<9\pi/8$	36.70	46.62
	$\Delta Cont<0.1$		**39.0**	**50.80**
		$\Delta Ori<\pi/8$	**37.9**	**49.0**

erties in the matching process improves the performance of solving image near-duplicate tasks. Our benchmark studies indicated that using the contrast and orientation features improves recall. Moreover, we showed that using only the orientation property obtains the best performance when the features are ranked based on the scale property, whereas involving only the contrast property improves the performance when the list of features are ranked and truncated based on the contrast property.

In future study we aim to assign continuous values to the weights in relations 4 , 5 and 6 instead of using discrete values (as we did in this study in Sections 3.3 and 5) based on the difference between the scale, contrast and orientation properties of features. Moreover, we aim to study the effect of using the properties of features to improve the retrieval of specific kinds of image near-duplicates.

REFERENCES

Aly, M., Welinder, P., Munich, M., and Perona, P. (2011). Caltech-buildings benchmark. In *Available at http://www.vision.caltech.edu/malaa/datasets/caltech-buildings/*.

Alyosef, A. A. and Nürnberger, A. (2016). Adapted sift descriptor for improved near duplicate retrieval. In *Proc. of the 5th International Conference on Pattern Recognition Applications and Methods*, pages 55–64.

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

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

Foo, J. and Sinha, R. (2007). Pruning sift for scalable near-duplicate image matching. In *Proceedings of the Eighteenth Conference on Australasian Database*, pages 63–71.

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.

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*, pages 506–513.

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

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

Nistèr, D. and Stewènius, H. (2006a). Recognition benchmark images. In *Available at http://www.vis.uky.edu/∼stewe/ukbench/*.

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

Y. Jianchao, Y. Kai, G. Y. and Thomas, H. (2009). Linear spatial pyramid matching using sparse coding for image classificationl. In *In: Proc*.

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*, 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.