Performance Comparison of Various Feature Detector-Descriptor Combinations for Content-based Image Retrieval with JPEG-encoded Query Images

Jianshu Chao, Anas Al-Nuaimi, Georg Schroth and Eckehard Steinbach

Institute for Media Technology, Technische Universität München
Munich, Germany
{jianshu.chao, anas.alnuaimi, schroth, eckehard.steinbach}@tum.de

***Abstract*—We study the impact of JPEG compression on the performance of an image retrieval system for different feature detector-descriptor combinations. The VLBenchmarks retrieval framework is used to compare a total of 60 detector-descriptor combinations for a dataset with JPEG-encoded query images. Our results show that among all tested detectors, the Hessian-Affine detector leads to the most robust performance in the presence of strong JPEG compression. Additionally, we compare the retrieval gains of the different detector-descriptor pairs after processing the JPEG-encoded query images with different deblocking filters. The results illustrate that for the MSER, MFD and WαSH detectors, the retrieval results benefit from two of the deblocking approaches at low bit rate irrespective of what descriptor the detectors are combined with. The same two deblocking filters are found to increase the retrieval performance for the MROGH descriptor when combined with most of the tested detectors.**

I. Introduction

Feature extraction is a fundamental operation for many computer vision applications, such as content-based image retrieval (CBIR), object classification, mobile visual search [1] and mobile visual location recognition [2]. For large-scale image retrieval, the database at the server side can consist of millions of images, resulting in huge memory requirements. Also, a growing number of consumer electronics products, such as smartphones and tablets, are equipped with digital cameras. Thus, more and more mobile visual search applications, such as Google Goggles [3], Kooaba [4] and IQ Engines [5] are emerging. All these approaches rely on a client-server architecture. Hence, network transmission and low latency requirements are additional issues. Working with compressed images saves storage space and allows for fast data transmission over mobile networks.

The JPEG standard is widely used and dominates the consumer electronics world and the world wide web due to its good compression ratio and computational simplicity. Thus, features often need to be extracted from JPEG-encoded images in the computer vision area. Methods to improve the JPEG algorithm to decrease the impact of compression on the feature quality have been developed in [6], [7], [8]. In this paper, we are interested in the impact of low bit rate JPEG compression on the performance of image retrieval system. For several features proposed in the literature, the impact of JPEG compression on the feature matching performance [9], [10], [11], [12] has already been evaluated. The evaluation is usually based on the criteria and dataset provided by Mikolajczyk et al. [13], which focuses on the *repeatability* or *matching score* between the uncompressed and compressed image pair. The question of how much the compression artifacts influence the performance of a real image retrieval system has, to the best of our knowledge, not yet been addressed.

This study differentiates itself from other works by involving many of the currently available detectors and descriptors, as opposed to limiting the study to a few of the most established features. Moreover, we opt for a system-level comparison making sure to study the impact of image compression on the overall performance of a retrieval system. We also investigate possible improvements in retrieval by applying different deblocking filters on the JPEG-encoded query images. Another important feature of this study is its easy reproducibility as we use a publicly available benchmarking framework, the VLBenchmarks by Lenc, Glushan and Vedaldi [14], and the freely available Oxford Buildings image dataset [15].

The remainder of this paper is organized as follows. In Section II the evaluation framework will be explained, including the evaluation criteria, the dataset and the studied detectors and descriptors. In Section III, implementation details are given and three deblocking approaches are presented. In Section IV our experimental results are presented and discussed. Section V concludes the paper.

II. Evaluation framework

In this section, the applied evaluation criteria are explained. Also, the used dataset is introduced as well as the detectors and descriptors that are evaluated.

A. Evaluation Criteria

The evaluation of features is typically performed either at the image-level [13], [9], [10], [11], [12] or using a system-

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/\$31.00

level evaluation framework [15], [16]. The former approach is normally based on only a few images, has the ability to compare many features in a short time and is independent of a specific application [17]. However, it cannot reflect the actual performance for specific applications. The repeatability and the matching score [13] are the most important criteria in this context for evaluating detectors and descriptors, respectively. System-level feature evaluation, on the other hand, comprises a large number of images and is typically performed for just one specific feature or one specific application, e.g., image retrieval or object classification. Similar to the evaluation strategy in [18], [19], [20], we use a simple image retrieval system to evaluate different detector-descriptor combinations. Our results hence illustrate the impact of low bit rate JPEG-encoded images on the actual retrieval performance. The experimental results in Section IV show that, depending on the actual detector and descriptor employed, this impact can be quite different. Particularly, we use the open source software VLBenchmarks [14] which implements the retrieval system in [16]. The benchmark determines the mean Average Precision (mAP) which is obtained by averaging the mean precision under the precision-recall curve over all query images.

B. Dataset

For the following experiments we use the Oxford Buildings dataset [15]. It consists of 5062 database images and a query set of 55 images taken from 11 distinct locations in the city (5 query images for each location). It should be noted that the images are already compressed with JPEG, although at high quality. The evaluated features in our work are extracted from the Y component, so all images are first transformed to YCbCr. In our experiments these gray scale (Y component) images are referred to as *original* images.

C. Feature Detectors and Descriptors

In the literature, various detectors and descriptors have been proposed and the most popular and important features are evaluated in our experiments. The *Hessian-Affine* detector, *Harris-Affine* detector [21] and the *Maximally Stable Extremal Regions* (MSER) detector [22] outperform other detectors in most of the experiments of Mikolajczyk [13]. Two other widely-used detectors, the *Difference of Gaussian* (DoG) [23] and *Speeded-Up Robust Features* (SURF) [24], are also included in our experiments. *STAR* (based on [25]) uses bi-level filters to approximate the Laplacian which is found to outperform other detectors and can be computed in real time. The *BRISK* [10] detector which is based on the *FAST* [26] detector is also compared in our experiments, since various detectors derived from FAST are widely used in many applications due to their short extraction time. We also include the recently proposed *Medial Feature Detector* (MFD) [27] and *WαSH* [28] detectors in our evaluation. The *Kaze* [29] detector which detects features in a nonlinear scale-space is shown to perform better than other classic detectors in [29] and is hence also included in the evaluation.

As descriptors, SIFT, PCA-SIFT [30], *Shape context (SC)* [31], *GLOH* [13] and SURF are tested. In addition, the recently proposed descriptors *MROGH* [11] and *LIOP* [12] seem to be strong competitors based on the results they show in their comparison to the more classic descriptors. The VLBenchmarks [14] that we use deals with real valued descriptors, thus, binary descriptors (e.g. the *BRISK* descriptor) are currently not included in our experiments.

III. Implementation details

In this section, we first provide relevant implementation details. Then, the tested deblocking approaches for improving the retrieval performance are presented.

Detector and descriptor combinations Typically, several implementations for a detector or descriptor exist. We find that the original implementations by the authors typically perform better than 3rd party implementations and hence use the original implementations whenever available. In addition, the number of features influences the retrieval performance. In general, the more features are detected, the better the results the retrieval system produces. In our experiments, the feature extraction speed is not considered. The parameters of each detector are tuned to generate approximately 500 features per image for the 55 query images. Table I shows the noteworthy parameters for each detector and gives the source of the implementation that is used in our work. Note that the DoG parameters of the authors' implementation cannot be tuned, thus we use *VlFeatSift* for our experiments.

TABLE I
Selected parameters for the studied detectors.

Detectors	Parameters	Values	Source
Hessian-Affine	threshold	640	Authors [13]
Harris-Affine	threshold	2200	
MSER	es	1	
	mm	8	
DoG	PeakThresh	4.6	VLBen. [14]
	firstoctave	0	
SURF	threshold	56000	Authors [24]
STAR	responseThreshold	26	OpenCV [32]
BRISK	threshold	76	Authors [24]
MFD	ff	2	Authors [27]
WαSH	t	0	Authors [28]
Kaze	dthreshold	0.00175	Authors [29]

The executables of SIFT, PCA-SIFT, Shape context and GLOH descriptor are provided by [13]. The MROGH and LIOP descriptor calculation is done by using the executables provided by the authors. These implementations use a descriptor region which is three times larger than the detector region. The enlarged descriptor region is mapped into a circular region to obtain affine-invariance. The patch is then scaled into a patch of length 41 pixels and the feature descriptor is calculated. Therefore, these descriptors are comparable. In summary, in our experiment we evaluate 10 different feature detectors and combine them with 6 different descriptors which leads to a total of 10×6 detector-descriptor combinations to be

Fig. 1. From left to right: Original image, JPEG-encoded image with quality value 12 (PSNR = 28.4 dB), DIP deblocked image (PSNR = 27.8 dB), SA-DCT deblocked image (PSNR = 29.2 dB), REAPP deblocked image (PSNR = 28.9 dB).

TABLE II
mAP SCORES (%) FOR ORIGINAL QUERY IMAGES

Detectors	avg. #db	avg. #qry	SIFT	PCA-SIFT	GLOH	SC	MROGH	LIOP	SIFT*[14]	SURF	Kaze
Hess.-Aff.	926.7	496.1	**61.6**	53.8	59.8	56.1	48.2	46.4			
Harr.-Aff.	952.6	495.5	**56.6**	48.7	56.4	51.8	47.4	46.1			
MSER	775.7	461.9	**69.4**	57.5	66.1	64.5	50.7	51.5			
MFD	1002.2	486.6	**63.9**	44.1	58.0	59.3	41.6	43.6			
WαSH	454.5	216.4	**53.0**	39.5	46.3	48.1	31.5	37.1			
DoG	872.9	500.2	**55.6**	49.0	52.9	47.1	53.9	40.8	**59.2**		
SURF	851.2	507.6	**60.6**	53.4	59.0	54.0	53.1	46.1		57.6	
STAR	783.1	513.7	**56.6**	49.6	54.8	50.2	51.6	42.4			
BRISK	1006.4	506.7	**52.8**	41.2	49.9	48.1	41.2	36.2			
Kaze	915.6	509.8	**53.4**	46.8	52.3	47.1	49.3	39.3			52.7

evaluated. In addition, the SIFT (*VlFeatSift*), SURF (original implementation) and Kaze (original implementation) features, which detect keypoints and produce feature vectors at the same time, are compared.

In VLBenchmarks the number of nearest neighbors for each query descriptor during the retrieval is set to 100, which means that the nearest 100 descriptors for one query descriptor are returned. The parameters *OkImagesNum*, *JunkImagesNum*, *BadImagesNum* are all set to *inf*, thus, all images in the Oxford Buildings dataset are included in our experiments.

Deblocking approaches Roughly speaking, the post-processing approaches for blocking artifact removal can be divided into three categories: spatial domain, DCT frequency domain and hybrid filtering methods. [33] proposes a hybrid filtering method which performs a soft blending of an edge-preserving and a low-pass filtering of JPEG-encoded images. In our experiment this deblocking filter is called *DIP* approach. [34] proposes a pointwise shape-adaptive DCT (*SA-DCT* approach) for filtering blocking artifacts which is also used in our experiments. [35] averages shifted versions of the already compressed image to reduce coding artifacts, which is named as *REAPP* approach. For all these deblocking filters, the default settings are used in our experiments. Fig.1 shows an example of the zoomed-in query image *all_souls_000013*. These deblocking approaches target the improvement of visual quality, however, it has not been studied (to the best of our knowledge) whether they have a positive impact on the performance of a feature-based image retrieval system or not.

IV. Experimental results

A. Retrieval results for original query images

The mAP scores for the original query images are presented in Table II. The left part in the table shows the names of the detectors (column 1), the average numbers of features for the database images (column 2) and that of the query images (column 3). While the parameters of the detectors are tuned to produce approximately 500 features for each of the 55 query images, the number of descriptors in the database images varies significantly. It should be noted that the WαSH detector cannot detect a large number of features even when the parameter t is set to 0. From Hessian-Affine to WαSH in the upper half of the table, the detectors are affine-invariant and from DoG to Kaze in the lower half, they extract circular regions as keypoints. The columns SIFT - LIOP present the mAP scores of each detector-descriptor combination. For the affine-invariant features, the descriptors obtain affine invariance by mapping the regions into circular ones. Generally speaking, the affine-invariant detectors perform better than their circular counterparts. Among them the MSER performs the best, followed by MFD in our experiments. The authors of the recently proposed detectors, MFD, WαSH, Kaze show that their detectors outperform DoG, SURF, Hessian-Affine or STAR. However, the results for our dataset suggest that MSER is still the best one. There are several reasons for the different results obtained in our experiments. First, we find that different implementations of one specific detector produce quite different results, e.g. *VlFeatMser* in VLBenchmarks

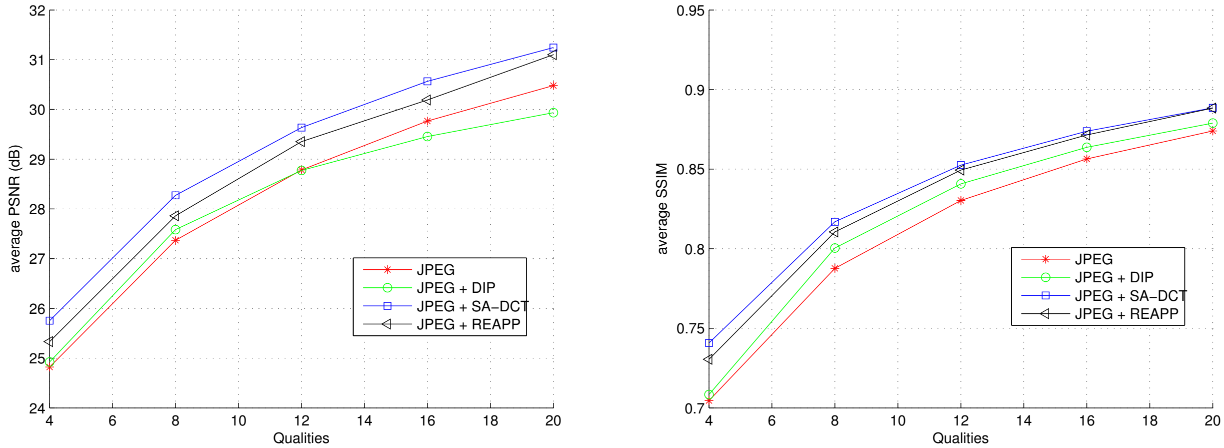

Fig. 2. PSNR and SSIM comparison for different deblocking filters

performs significantly worse than the one we use here. This is also true when using the OpenCV implementations of SIFT or SURF. This could be one important reason for the different rankings of the detectors in our results. Second, the parameters used in our experiments are different compared to other works. In addition, the image retrieval framework and its setup are different.

As for the descriptors, in [11], [12] and the evaluation work of [20] MROGH and LIOP perform the best. However, their evaluation work is based on the dataset of Mikolajczyk [13]. Compared to this dataset, the Oxford Buildings dataset used here contains more realistic image changes, such as lighting change, perspective change, scale change and camera photographic change. Under such severe conditions, the detected locations, scales of keypoints or pixel intensities vary to a large extent. Thus, the segmentation techniques in MROGH and LIOP are strongly affected, yielding poor performance. Surprisingly the classic SIFT stands the test of time and is superior to all other descriptors. The comparison of SIFT, SURF and Kaze features is presented in the right part of Table II. *SIFT** means the descriptors are extracted using *VlFeatSift*. Compared to the SIFT extraction provided by [9], its performance increases from 0.556 to 0.592 due to the fact that the radius of the region used for computing the descriptor is six times larger than the detected scale, rather than three times as used in [9]. The Kaze feature does not show superior results in our experiments.

B. Compression and deblocking

In order to examine the impact of JPEG compression on the retrieval results, we use the software from [36] to compress the 55 query images with quality values ranging from 4, 8, 12, 16 to 20. For each detector-descriptor combination, the mAP score is computed by averaging the results for the 55 JPEG-encoded query images encoded at the same quality level. Subsequently, the three different deblocking approaches are applied on the 55 query images and the corresponding mAP score is determined. First PSNR and SSIM are calculated to examine whether the deblocking approaches work and which one performs the best in terms of image quality. Fig. 2 shows that the *SA-DCT* approach performs the best followed by the *REAPP* approach for all quality values in terms of both PSNR and SSIM. The *DIP* approach performs worse than the normal JPEG at high bit rates in terms of PSNR, however, it is better for all quality values in terms of SSIM.

C. Comparison

First, we are interested in the robustness against strong JPEG compression. Since the SIFT descriptor results in the best image retrieval performance for the original query images, we focus on it first. From Table III it can be seen that the Hessian-Affine detector is quite robust. The mAP score for the Hessian-Affine detector drops from 0.616 to 0.547 which is the highest when compared to the other detectors at quality value 4. In contrast, the mAP score of the MSER detector is only 0.413 which indicates that it is strongly affected by the compression artifacts. This is due to the fact that blocking artifacts generate many unwanted keypoints during the process of the watershed segmentation algorithm for strongly JPEG-encoded images. For quality value 12 and better, the MSER detector again shows its superior performance. Second, the impact of the three deblocking approaches is compared. In Table III, the white, green and yellow backgrounds for the three deblocking approaches represent the same, increased and decreased mAP scores compared to the normal JPEG-encoded images, respectively. For MSER, the deblocking approach *REAPP* increases the performance at qualities 4, 8, 12 and 16, with a higher mAP gain, the higher the compression rate is. The performance of the MFD and WαSH detectors is also improved by the deblocking approaches *SA-DCT* and *REAPP* at low qualities 4 and 8. However, the deblocking approach *DIP* decreases the performance in most cases. For the other detectors, the performance decreases after deblocking. The results for quality value 20 are already close to the results for the original query images. Thus, the mAP scores for normal JPEG images, *DIP*, *SA-DCT* and *REAPP* deblocked images show no big difference. Third, due to limited space only the Shape context and MROGH descriptors are compared beside

TABLE III

mAP SCORES (%) FOR QUERY IMAGES COMPRESSED AT DIFFERENT QUALITIES

Detectors	SIFT				Shape context				MROGH			
Quality 4	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP
Hess.-Aff.	54.7	50.4	53.4	54.0	48.3	44.4	46.9	48.1	40.1	38.9	41.3	40.5
Harr.-Aff.	48.9	43.7	48.3	48.7	43.3	38.9	42.8	43.7	38.2	36.3	38.9	38.8
MSER	41.3	39.3	42.0	42.8	34.2	31.6	37.3	37.3	25.6	31.4	33.1	30.9
MFD	47.0	46.5	49.9	50.6	42.3	40.6	44.4	45.7	28.0	29.5	30.0	28.9
WαSH	45.2	45.1	46.4	45.7	40.9	39.9	40.7	41.0	25.4	27.1	25.9	25.9
DoG	29.7	29.5	27.5	24.8	23.9	23.9	24.4	22.5	33.3	32.8	36.0	33.6
SURF	47.4	45.3	48.2	47.7	41.8	39.4	41.4	41.3	40.3	40.6	43.9	42.4
STAR	38.9	34.6	35.7	34.4	32.0	28.6	28.9	27.6	36.3	35.0	38.8	38.0
BRISK	41.5	36.7	38.5	39.1	37.9	31.8	33.9	33.9	29.4	27.0	28.6	27.3
Kaze	37.7	36.5	36.6	37.0	32.7	32.7	32.1	31.7	36.3	36.7	40.1	38.3
Quality 8	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP
Hess.-Aff.	60.3	58.5	59.2	59.5	54.2	52.7	52.7	53.4	46.3	44.8	46.3	45.9
Harr.-Aff.	54.7	52.6	53.0	54.0	49.4	47.3	48.0	48.4	43.7	42.5	44.5	43.9
MSER	57.6	57.7	58.1	59.1	49.9	49.4	50.6	52.0	39.2	41.2	42.0	41.4
MFD	56.7	56.9	56.7	57.6	52.0	51.6	52.0	53.3	34.3	35.1	35.1	35.0
WαSH	50.8	50.1	50.9	51.2	45.8	45.6	45.8	46.6	29.5	30.3	30.0	30.4
DoG	40.6	40.0	37.9	38.3	34.7	33.8	33.2	34.0	45.9	45.6	46.7	45.2
SURF	56.4	54.3	54.4	56.3	49.6	47.4	48.0	48.5	50.0	49.0	51.1	50.0
STAR	48.2	45.3	46.2	45.7	41.7	37.6	39.3	38.9	45.2	45.1	46.8	45.7
BRISK	46.5	45.5	44.4	45.1	42.1	41.3	40.2	40.9	37.0	36.2	36.4	34.7
Kaze	47.4	46.2	46.4	47.8	41.9	40.9	39.7	41.9	46.2	45.8	47.1	46.6
Quality 12	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP
Hess.-Aff.	61.2	61.5	60.5	61.2	55.8	55.2	54.8	55.6	46.9	46.2	46.9	46.5
Harr.-Aff.	55.9	55.6	55.3	55.8	50.4	49.6	50.1	50.7	45.5	45.0	45.7	45.3
MSER	63.3	63.0	62.8	63.9	55.8	55.4	55.4	56.4	44.1	44.5	46.0	45.3
MFD	59.3	57.9	58.4	59.2	54.7	54.1	53.5	55.4	37.5	36.5	37.9	37.8
WαSH	50.0	51.7	51.1	51.2	45.6	47.0	46.4	46.4	30.2	31.5	32.2	30.5
DoG	47.1	45.7	44.4	44.4	39.7	38.1	38.1	38.3	50.0	49.1	50.0	49.8
SURF	57.7	56.6	56.8	57.6	51.6	49.8	50.2	51.1	51.1	50.2	52.3	51.4
STAR	51.9	50.3	49.6	49.8	43.1	42.4	42.5	42.2	48.5	48.4	49.1	48.9
BRISK	49.4	49.2	48.7	49.1	45.0	45.2	43.0	43.8	39.2	39.1	37.9	37.7
Kaze	50.8	50.1	49.4	50.3	44.7	43.2	43.3	44.0	48.4	47.3	48.2	47.5
Quality 16	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP
Hess.-Aff.	60.9	60.9	60.5	60.5	55.4	55.1	54.9	54.7	47.3	47.1	47.5	47.2
Harr.-Aff.	55.5	55.9	55.3	55.9	50.3	49.9	49.9	50.1	45.8	45.4	46.2	45.9
MSER	65.5	64.9	65.5	65.9	58.8	57.3	58.1	58.6	46.1	44.6	46.0	47.0
MFD	61.0	59.4	60.1	60.2	57.0	55.8	55.8	56.0	38.9	37.9	38.5	39.1
WαSH	52.4	50.7	51.6	52.0	48.1	46.4	46.5	47.2	31.0	30.3	31.4	31.3
DoG	49.1	48.9	46.9	47.5	41.0	40.1	39.0	39.8	51.4	50.6	51.0	51.1
SURF	59.6	58.6	58.3	59.6	52.3	50.8	51.5	51.9	52.9	51.7	52.6	52.5
STAR	53.1	51.9	50.7	51.7	46.3	44.4	43.9	44.7	50.2	49.2	50.6	50.3
BRISK	50.2	50.1	49.0	48.9	45.8	47.0	44.2	44.0	39.7	39.2	39.2	38.6
Kaze	52.0	51.5	50.9	51.9	45.0	44.6	43.5	44.7	48.4	48.0	48.7	48.7
Quality 20	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP	Normal	DIP	SA-DCT	REAPP
Hess.-Aff.	61.3	62.1	61.0	61.5	55.8	55.9	55.2	56.1	47.7	47.2	47.7	47.5
Harr.-Aff.	56.9	56.6	56.6	56.6	52.0	50.6	51.3	51.5	46.3	45.3	46.2	46.0
MSER	67.6	65.7	66.7	66.8	61.4	58.9	60.7	60.9	47.7	45.7	47.7	47.7
MFD	61.2	60.5	60.1	61.0	57.6	56.5	57.1	57.2	39.3	38.3	39.3	39.4
WαSH	52.5	52.2	52.7	51.7	48.6	47.2	48.3	47.4	31.4	30.7	32.8	32.1
DoG	49.8	49.4	48.0	49.2	42.2	41.8	41.1	41.7	51.4	51.3	51.4	51.5
SURF	59.7	58.8	58.5	59.2	53.2	51.5	51.7	52.5	52.3	51.8	52.5	52.4
STAR	54.1	53.3	52.9	53.6	46.9	45.4	45.9	46.2	50.8	50.0	50.5	51.0
BRISK	51.0	50.3	50.4	49.4	46.6	47.1	45.9	45.0	40.2	40.2	40.1	39.1
Kaze	52.3	52.0	51.2	52.2	45.2	45.4	44.3	44.7	49.3	48.1	49.3	49.1

the SIFT descriptor. For low qualities, both *SA-DCT* and *REAPP* have a positive impact on the MROGH descriptor in most cases. This can be explained by the fact that the *SA-DCT* and *REAPP* approaches remove many strange pixels caused by compression artifacts in patches which can be viewed in Fig.1. Afterwards, the following processes of sorting and segmentation according to pixel intensities in MROGH algorithm are more correct. The *SA-DCT* and *REAPP* approaches have a positive impact on MSER, MFD and WαSH detectors for low bit rate query images as well.

V. Conclusion

In this paper, we first evaluate the image retrieval performance for different combinations of well-established as well as recently proposed detectors and descriptors using the VLBenchmarks. We find that, the SIFT descriptor combined with the MSER detector outperforms even recently proposed feature types. In a second evaluation we quantify the effect of JPEG compression on the retrieval performance. Finally, we apply deblocking filters on the compressed query images for compression artifact removal and measure the change in retrieval performance for a wide range of detector-descriptor combinations. We find that although the deblocked images improve in terms of visual quality, the retrieval performance improves only in the case of high compression and mostly when the MSER detector is used irrespective of the descriptor, or when the MROGH descriptor is used irrespective of the detector. The results also show that the retrieval performance typically suffers when a deblocking filter is used on high-quality JPEG queries. It is also found that only the *SA-DCT* and *REAPP* deblocking approaches can generally yield retrieval performance improvements.

Acknowledgment

This work has been supported by a PhD grant from the China Scholarship Council for Jianshu Chao.

References

[1] B. Girod, V. Chandrasekhar, D. M. Chen, N. M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 61–76, Jul. 2011.

[2] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach, "Mobile visual location recognition," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 77–89, Jul. 2011.

[3] Google Goggles. http://www.google.com/mobile/goggles/.

[4] Kooaba. http://www.kooaba.com/.

[5] IQ Engines. https://www.iqengines.com/.

[6] J. Chao and E. Steinbach, "Preserving SIFT features in JPEG-encoded images," in *Proc. IEEE International Conference on Image Processing, Brussels, Belgium*, Sep. 2011.

[7] L. Duan, X. Liu, J. Chen, T. Huang, and W. Gao, "Optimizing JPEG quantization table for low bit rate mobile visual search," in *Proc. Visual Communications and Image Processing*, San Diego, CA, USA, Nov. 2012, pp. 1–6.

[8] J. Chao, H. Chen, and E. Steinbach, "On the design of a novel JPEG quantization table for improved feature detection performance," in *Proc. IEEE International Conference on Image Processing*, Melbourne, Australia, Sep. 2013.

[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Computer Vision (ICCV)*, Barcelona, Spain, Nov. 2011.

[11] B. Fan, F. Wu, and Z. Hu, "Rotationally invariant descriptors using intensity order pooling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2031–2045, Oct. 2012.

[12] Z.Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proc. Int. Conf. Computer Vision (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 603–610.

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005.

[14] K. Lenc, V. Gulshan, and A. Vedaldi, "Vlbenchmarks," http://www.vlfeat.org/benchmarks/, 2012.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Int. Conf. Compter Vision and Pattern Recognition*, Minneapolis, US, Jun. 2007, pp. 1–8.

[16] H. Jegou, M. Douze, and C. Schmid, "Exploiting descriptor distances for precise image search," 2011, Technical Report 7656, INRIA.

[17] Modern features: advances, applications, and software. https://sites.google.com/site/eccv12features/slides.

[18] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, Jul. 2007.

[19] A. L. Dahl, H. Aanæs, and K. S. Pedersen, "Finding the best feature detector-descriptor combination," in *Proc. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011, pp. 318–325.

[20] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *International Conference on Pattern Recognition (ICPR)*, Tsukuba Science City, Japan, Nov. 2012.

[21] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal on Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. British Machine Vision Conf. (BMVC)*, Cardiff, UK, Sep. 2002, pp. 384–396.

[23] D. Lowe, "Distinctive image feature from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[24] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. European Conf. Computer Vision (ECCV)*, Graz, Austria, May 2006, pp. 404–417.

[25] M. Agrawal, K. Konolige, and M. R. Blas, "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching," in *Proc. European Conf. Computer Vision (ECCV)*, 2008, pp. 102–115.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. on Computer Vision (ECCV)*, Graz, Austria, May 2006, pp. 430–443.

[27] Y. Avrithis and K. Rapantzikos, "The medial feature detector: Stable regions from image boundaries," in *Proc. Int. Conf. Computer Vision (ICCV)*, 2011, pp. 1724–1731.

[28] C. Varytimidis, K. Rapantzikos, and Y. Avrithis, "WαSH: Weighted α-Shapes for Local Feature Detection," in *Proc. European Conf. Computer Vision (ECCV)*, 2012, pp. 788–801.

[29] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features," in *Proc. European Conf. Computer Vision (ECCV)*, 2012, pp. 214–227.

[30] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Oct. 2004, pp. 506–513.

[31] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 509–522, 2002.

[32] OpenCV 2.4.3. http://opencv.willowgarage.com/wiki/.

[33] T. Pham and L. van Vliet, "Blocking artifacts removal by a hybrid filter method," in *Proc. 11th Annual Conf. of the Advanced School for Computing and Imaging, ASCI, Delft*, 2005, pp. 372–377.

[34] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, pp. 1395–1411, May 2007.

[35] A. Nosratinia, "Enhancement of JPEG-compressed images by re-application of JPEG," *Journal of VLSI Signal Processing*, vol. 27, pp. 69–79, Feb. 2001.

[36] "Independent JPEG Group," http://www.ijg.org/.