Performance Assessment of Feature Detector-Descriptor Combination

A. M. M. Madbouly[1], M.Wafy[2], Mostafa-Sami M. Mostafa[3]
[1] Helwan University, Mathematics Department, Faculty of Sciences, Cairo, Egypt.
[2,3]Helwan University, Faculty of Computers and Information, Cairo, Egypt.

Abstract

Features detection and description among multiple images are widely used in many applications, e.g., feature matching, object categorization, 3D construction, image retrieval and object recognition. This paper evaluates combination performance of different feature detectors and descriptors. It will compare performance of detectors and descriptors combination on images under rotate, scale constraints and distortion such as illumination on different scene (bedroom, industrial and CALsuburb). An experimental result shows MinEigen detector has best result in number of detected key-points when handle rotate, scale and illumination and not affected with scene. SURF without external detector is the best when handle rotate and scale constraint in different levels and scene. FAST/SURF and Harris/FREAK are best combined against illumination distortion in different levels. This review introduces a brief introduction for providing a new research in feature detection field to find appropriate method according to their condition.

Keywords: *local feature, detectors, descriptors Component, FREAK, SURF, BRISK, MSER, MinEigen.*

1. Introduction.

Local features detectors and descriptors play an important part in many applications like mapping, text recognition, license plate recognition and content based image retrieval, image registration [1], object recognition [2], object categorization [3], texture classification [4], robot localization [5], and video shot retrieval [6].

There are many researches that build new fast and robust detector (SIFT[7], SURF[8],Fast [9], BRISK [10]and descriptors SIFT[7],SURF [8], BRISK[10], Harris[11], FREAK [12], MinEigen, MSER,HOG).

Local features can be utilized into two different methods. First method includes three steps: feature detection, feature description, and feature matching. Second method is bag-of-features [13] and hyperfeatures [14] that includes feature detection, feature description, feature clustering, and frequency histogram construction for image representation. A local feature extraction is composed of feature detector and a feature descriptor. Feature detectors (Moravec's corner detector) [15], that search for the local maximum and minimum intensity changes, e.g. Harris and Stephens [16],

2. Feature detectors and Descriptors.

Feature detection is an essential step in feature description. It finds points and regions to use it as descriptors of features. Most of detectors can be classified into two types, corner detectors and region detectors.

2.1 SIFT (Scale Invariant Feature Transform)

SIFT was proposed by Lowe et al. [7], it was invariant to scale, 3D camera viewpoint, rotation and partially invariant to change in illumination.

SIFT consists of the following steps: -

1) Key-point detect

It is not possible to use the same window to detect key-points of different size. LoG (Laplace of Gaussian) can be used as blob detection to detect corner points of different scale (detect blob in various sizes due to change in variance). Variance works as a scale parameter; for example, Gaussian kernel with low variance gives high value for small corner while Gaussian kernel with high variance fits well with large corner. Local maxima across scale and space can be found from the list of coordinates of points (x, y) and variance (v) of these coordinates, i.e. There is a potential key-point in these coordinate (x, y) at variance (v) scale. SIFT used DoG (Difference of Gaussian) to avoid LoG limitation (it needed a lot of computation that consumed time and memory). Difference of Gaussians that w an approximation of LoG. It is computed as the difference of Gaussian blurring of an image with two different variances. This process is done for different octaves of the image in Gaussian pyramids as in the figure below.

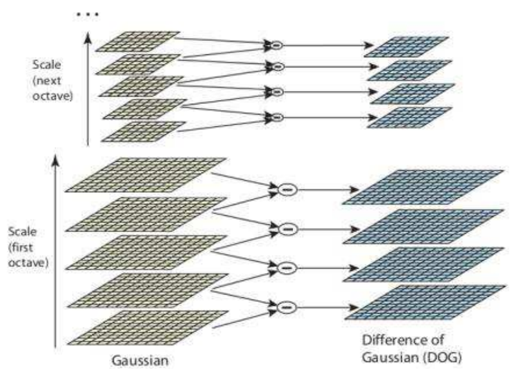


Fig. 1. Shows how DoG computed.

SIFT searches for a local extrema over scale and space when it finds DoG of octaves. For instance; one pixel in an image is compared with its eight neighbours as well as nine pixels in a next scale and nine pixels in previous scales. Key-point considers being a potential key-point, if it has a best representation in that scale, as shown in the following figure:

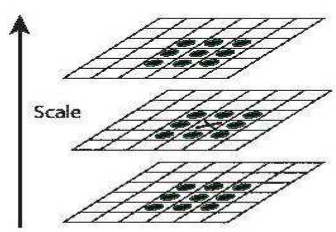


Fig. 2. Extrema Computation

Once potential key-point locations are found, they have to be refined to get more accurate results. SIFT uses Taylor series expansion of scale and space to get a more accurate location of extrema. If the intensity of this extrema is less than a threshold value, it will be rejected.

2) Orientation Assignment

Invariance to image rotation can be achieved by assigning the orientation to each key-point . A neighborhoods are chosen around the key-point location based on the scale, and the gradient magnitude and direction which is computed in that region. An orientation histogram with thirty six bins covering 360 degrees is constructed. It is weighted by the gradient magnitude and Gaussian weighted circular window with σ equal to 1.5 times the scale of a key-points. The orientation can be calculated by choosing the highest peak in the histogram and any peak above 80%.

It constructs key-points in a same location and scale, but in different directions. It contributes to stability of the matching.

3) Key-point Descriptor

Now the key-point descriptor is constructed. A 16x16 neighborhoods around the key-point are chosen. It is split into 16 sub blocks of 4x4 sizes. For each sub block, eight bin orientation histograms are constructed. So a total of 128 bin values are obtainable. It is represented as a vector to figure key-point descriptor. Moreover, several measures are chosen to accomplish robustness against illumination changes, rotation and etc.

4) Key-point Matching

Images key-points can be matched by identifying their nearest neighbors. However in some cases, the second closest match can be very near to the first. It may be happening because of noise or some other reasons. In that case, the ratio of closest distance to second closest distance is chosen. If it exceeds 0.8, they will be rejected. It removes around 90% of false matches while rejecting only 5% correct match.
SIFT match images by matching their key-points nearest neighbors. Sometimes it uses a ratio of the closest distance between key-point descriptor vectors.

There are many researchers who used SIFT in various application; for example, Mehrotra proposed an efficient indexing scheme for searching large iris biometric database that achieves invariance to similarity transformations which is based on SIFT Key-points[17]. Piccinini presented a novel approach for detecting and finding duplicate objects in pick-and-place applications under extreme conditions of occlusion, where standard appearance-based approaches are likely to be ineffective depending on mean shift and SIFT [18]. Zhou proposed a method for object tracking in real scenarios. It used SIFT to extract region of interest then perform mean shift to conduct similarity search via color histogram [19]. Y. Nig et al. proposed simplified SIFT because original SIFT has large amount of computation that makes it inappropriate for real time detection. New SIFT, firstly, uses geometric center of license plate characters as the feature points. Secondly, key direction is produced by PCA as an attempt to simplify feature detection step. Finally, it uses SVM classification algorithm. The algorithm requires some improvements for Chinas licenses plate detection [20]. M. Zahedi et al. presented a method that depended on SIFT to recognize licenses plate. Limitation of this method was performing preprocess (vertical edged detection) on input images that consuming time [21]. F. Silva et al. developed real time automatic vehicle license recognition based on SIFT descriptor. SIFT was used to compare key-points of template characters and extracted key-points of images [22].

Zhu presented an image registration algorithm named BP-SIFT, where key-point matching of SIFT descriptors was formulated as a global optimization problem and supplied a sub optimum solution using belief propagation (BP) [23].
Liao proposed expansion to SIFT descriptor for image retrieval and matching. Firstly, it normalizes elliptical neighboring region; secondly, it transforms to affine scale-space. Finally, it uses polar histogram orientation bin to improve SIFT descriptor [24].
Li proposed an algorithmic method based on the SIFT for multispectral images based on geometric algebra (GA). Firstly, he set a new representation of multispectral image including spatial and spectral information. Secondly, he presented a new method for getting the scale space of the multispectral image. Thirdly, he performed SIFT, computed geometry algebra that was based on difference of Gaussian images, detecting key-points. Finally, he featured points that can be detected and described [25]

2.2 Speeded-Up Robust Features (SURF)

SURF [8] (Speeded-up robust features) is considered to be a scale and rotation invariant detector and descriptor. It is based on the same principles and steps of SIFT, but it uses a different scheme. It works much faster without loss of invariant to rotation and different type of noise. SURF consists of the following steps:

1) Key-point *detection*

It uses blob detectors based on a Hessian matrix. The detection of Hessian is defined as the maximal local change around the area. In SIFT, Lowe approximated LoG with DoG for finding scale and space but SURF approximates LoG with 9x9Box Filter as showed in figure below.

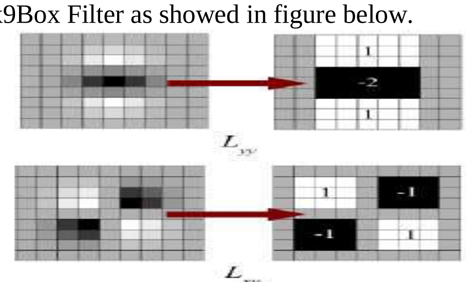


Fig. 3. Shows Box Filter approximate LoG.

2) Key-point Description

The SURF descriptor depends on the similar properties of SIFT. It determines orientation that is based on information in a circular area around the interest point. It extracts feature descriptor from a square area that aligned to select orientation.

SURF was used in many applications; for instance, Kang et. al. used SURF in face recognition [26]. Zhiheng et. al. built a motion tracking method that combined features of the mean shift and SURF. It was based on computing the orientation and proportion of SURF features of previous and current frames to realize a scale and orientation changing tracking [27]. Lee et. al. presented a novel image retrieval method (applicable to mobile environment) that worked on color images. It clustered extracted SURF features with well-known dominant color descriptor [28].

Pan et. al. presented a real time object tracking method. He tried to improve SURF by reducing the computation complexity via the number of detected feature points, reducing repeated calculation and improve matching method [29]

Juan and Gwun presented a panorama image stitching method. Firstly, it extracted feature descriptor using SURF. Secondly, it used KNN to get matching pairs and erase the mismatch couples by RANSAC (Random Sample Consensus), then; it adjusted the images by bundle adjustment and estimated the accurate homograph matrix. Finally, it blended images via multi band blending [30]

Huang et. al. proposed a modified SURF descriptor named I-SURF. It modified SURF descriptor by considering the boundary effect of the adjacent sub regions, and introduced the index vector to accelerate matching process [31]

Sig Do presented hardware architecture system for real time object tracing that depends on SURF IP [32].

Huiqing modified matching step in SURF by using KNN (to improve matching time) and combined it with SUSAN algorithm [33].

Fan et. al. proposed Color-SURF descriptor that combined local kernel color histograms and Haar wavelet responded to construct the feature vector. So the descriptor was a two elements vector. In image matching step, SURF descriptor was first compared, then, the unmatched points were computed by Bhattacharyya distance between their local kernel color histograms [34]

Li et. al. built a face recognition method based on SURF descriptor. It used information around the neighborhood of the sub-block by masking with 3X3 window templates and then constructed the descriptor that could get better discriminative power [35].

Du et. al. proposed an invariant object recognition method (SSURF) based on SURF applied on robot visual recognition. It reduced consuming time for interest search [36].

Wang et. al. introduced an adaptive image stitching method based on SURF. It adaptively found the appropriate uniform distribution radius that was based on image complexity and utilized this radius to wipe off amounts of unnecessary interest points. The remaining interest points would be distributed well in the image and much less than the number of interest points detected by the original SURF [37].

2.3 Fast Retina Key-point (FREAK)

FREAK [12] tried to mathematically simulate human vision process by using a retinal sampling grid. In addition, higher density point is located at the center. These points at the center drop potentially as coming closer to the edge. It uses different Gaussian kernel sizes for every sampling point.

FREAK evaluated forty three weighted Gaussian at the location that around the key-point. The pixels were concentrated near the key-point on the circular region (simulate retinal pattern). It used cascade for comparing these pairs.

Krizaj used FREAK descriptor for 3D face recognition [38]. Whiten et. al. presented a local spatiotemporal descriptor for action recognition depending on FREAK descriptor. It built a short string. The first byte was FREAK descriptor. The remaining bytes strengthened the motion model by building a binary string through local motion patterns [39].

2.4 Features from accelerated segment test (FAST)

FAST [9] was considered to be corner detector that was used to extract key-points. It did not need a lot of computation so it was faster than many other detectors like SIFT and Mineigen. Fast used Bresenheim algorithm to evaluate every circle around feature point to detect features.

3. Experiments

This paper aims to evaluate the performance of combining between detectors and descriptors (FREAK, SURF and FREAK) on various constraints types (rotate, scale and illumination).

Experiment will work on database images [40] of sizes 200 X 276, 247 X 220 and 220 X 330.

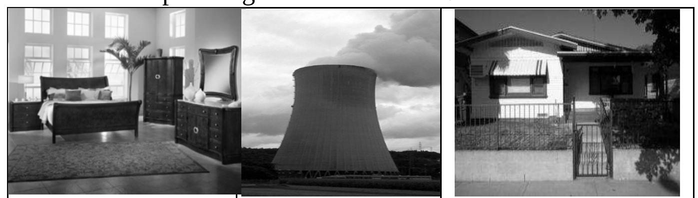
It consists of three tests that are used (SURF, FAST, BRISK, Harris and MinEigen) as detectors. Firstly, test one uses FREAK as descriptor. Secondly, test two uses SURF as a descriptor. Finally, test three is use BRISK as a descriptor as shown in the following table.

Table 1: shows combination of detectors and descriptors

Test number	Detector / Descriptor
Test 1	detector/FREAK
Test 2	detector/SURF
Test 3	detector/BRISK

It was performed on Laptop Lenovo 3000 C100 that has the following contents: Intel® Pentium® M Processor 1.73 GHz 1.73 GHz, 2MB cash: and RAM 2GB. The code was written in Matlab R2014a on windows 7 professional (32 bit).

Test aims to evaluate performance of descriptors against different constraints (rotation, scale and illumination) under different scene (bedroom, industrial and CALsuburb). Figure 2 show example image from each scene.



(a) (b) (c)

Fig.2. (a) represents sample of bedroom, (b) represents sample of industrial and (c) represents sample of CALsuburb.

The experiment will evaluate the number of key-points that detectors can capture them and time that need to detect and extracted these features.

Experiment will divided into three tests as shown in table 1.

Firstly, experiment will evaluate the detectors by number of captured key-points against various noises in different scenes.

The following table's shows number of detected key points and time need to capture them.

Table 2: shows detected key-points and time needed for bedroom scene.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*
SURF	115	142	147	147	190	0.09	0.09	0.12	0.12
FAST	134	198	161	161	179	**0.01**	**0.01**	**0.01**	**0.01**
BRISK	80	104	95	95	96	0.02	0.02	0.02	0.02
Harris	137	223	91	227	240	0.08	0.09	0.09	0.09
MinEigen	**491**	**571**	**386**	**571**	**677**	0.08	0.11	0.11	0.11

Table 3: illustrate number of detected key-points and detecting time of industrial scene

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*
SURF	92	106	99	121	213	0.08	0.09	0.10	0.10
FAST	68	101	93	93	121	**0.01**	**0.01**	**0.01**	**0.01**
BRISK	45	62	60	52	65	0.02	0.02	0.03	0.02
Harris	83	155	47	211	196	0.08	0.14	0.12	0.11
MinEigen	**421**	**350**	**213**	**400**	**455**	0.09	0.13	0.13	0.13

Table 4: CALSUBURB detected key-points and Time of detection.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*	*Rotate 25*	*Rotate 45*	*Rotate 75*	*Rotate 100*
SURF	202	214	202	227	258	0.12	0.15	0.14	0.12
FAST	257	313	284	292	297	**0.02**	**0.01**	**0.01**	**0.01**
BRISK	161	172	170	174	175	0.04	0.04	0.03	0.03
Harris	261	344	201	401	387	0.14	0.16	0.14	0.12
MinEigen	**799**	**874**	**688**	**964**	**1016**	0.12	0.17	0.17	0.16

Previous result shows that MinEigen detector is the most precious one and FAST detector is the fasttest detector.

In following tables experiment studies effect of scale constraint in different scene.
Scale 1.2, 1.4, 1.7 and 2 means zoom in image by 20%, 40%, 70% and 100% respectively.

Table 5: Detected Key-points and Time Needed for Bedroom Scene

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF	152	174	229	271	152	0.08	0.07	0.11	0.15
FAST	119	132	130	119	119	**0.01**	**0.01**	**0.01**	**0.01**
BRISK	91	108	115	121	91	0.02	0.02	0.02	0.02
Harris	153	181	218	248	153	0.08	0.08	0.10	0.14
MinEigen	**617**	**780**	**1021**	**1265**	**617**	0.10	0.11	0.13	0.19

Table 6: Illustrate Number of Detected Key-points and Detecting Time of Industrial Scene

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF	114	135	158	191	114	0.09	0.08	0.10	0.14
FAST	57	57	55	46	57	**0.01**	**0.02**	**0.01**	**0.01**
BRISK	45	48	53	54	45	0.02	0.02	0.02	0.02
Harris	88	101	113	136	88	0.09	0.09	0.12	0.16
MinEigen	**520**	**615**	**810**	**1001**	**520**	0.11	0.14	0.15	0.21

Table 7: CALSUBURB detected Key-points and Time of Detection.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*	*Scale 1.2*	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF	257	324	465	567	257	0.12	0.12	0.16	0.22
FAST	243	247	219	183	243	**0.01**	**0.01**	**0.02**	**0.02**
BRISK	174	188	197	191	174	0.03	0.02	0.03	0.03
Harris	313	378	461	534	313	0.11	0.11	0.15	0.21
MinEigen	**1047**	**1390**	**1969**	**2456**	**1047**	0.16	0.16	0.22	0.30

Table 8: Shows detected key-points and Time needed for bedroom scence.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*
SURF	152	174	229	271	152	0.08	0.07	0.11	0.15
FAST	119	132	130	119	119	**0.01**	**0.01**	**0.01**	**0.01**
BRISK	91	108	115	121	91	0.02	0.02	0.02	0.02
Harris	153	181	218	248	153	0.08	0.08	0.10	0.14
MinEigen	**617**	**780**	**1021**	**1265**	**617**	0.10	0.11	0.13	0.19

Table 9: Illustrate Number of Detected Key-points and Detecting Time of Industrial Scene.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*
SURF	114	135	158	191	114	0.09	0.08	0.10	0.14
FAST	57	57	55	46	57	**0.01**	**0.02**	**0.01**	**0.01**
BRISK	45	48	53	54	45	0.02	0.02	0.02	0.02
Harris	88	101	113	136	88	0.09	0.09	0.12	0.16
MinEigen	**520**	**615**	**810**	**1001**	**520**	0.11	0.14	0.15	0.21

Table 10: Display CALSUBURB detected key-points and Time of Detection.

Detectors	Detected feature Points					Detected feature Points time			
	Original image	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*	*Illumination 1*	*Illumination 2*	*Illumination 3*	*Illumination 4*
SURF	257	324	465	567	257	0.12	0.12	0.16	0.22
FAST	243	247	219	183	243	**0.01**	**0.01**	**0.02**	**0.02**
BRISK	174	188	197	191	174	0.03	0.02	0.03	0.03
Harris	313	378	461	534	313	0.11	0.11	0.15	0.21
MinEigen	**1047**	**1390**	**1969**	**2456**	**1047**	0.16	0.16	0.22	0.30

Previous tables shows that MinEigen detector is precocious than other detection but it is not the fast like FAST detector.
According to large number of detected key-points as shown in previous tables, we will select strongest twenty key-points to be able to notice if matching is true or not by eye.
Experiment finds that SURF is the best when used as detector and descriptor than combining detectors (Harris, FAST, BRISK, MinEigen) with descriptors BRISK and FREAK at bedroom scene in different levels of rotation, scale and illumination as shown in following tables.

Table 11: shows average number of matched features of bedroom scene under different levels of constraints.

Detectors/ Descriptor	Detected feature Points			
	Rotate 25	*Rotate 45*	*Rotate 75*	*Rotate 100*
SURF/FREAK	5.60	5.80	8.20	8.40
SURF/SURF	**9.40**	**7.40**	**12.40**	**11.60**
FAST/BRISK	9.20	10.20	14.00	10.40
	Scale 1.2	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF/FREAK	8.40	8.40	6.20	6.20
SURF/SURF	**10.20**	**10.60**	**8.40**	**8.80**
FAST/BRISK	2.20	4.20	2.60	3.60
	Illumination 1	*Illumination 2*	*Illumination 3*	*Illumination 4*
SURF/FREAK	8.40	8.40	6.20	6.20
SURF/SURF	**10.20**	**10.60**	**8.40**	**8.80**
SURF/BRISK	2.20	4.20	2.60	3.60

Experiment also tries to study if the scene effect on result or not, so we will examine a number of matched features with industrial scene and found SURF without external detector is the best combination at rotate and scale constraints.
FAST/SURF and Harris/FREAK is best combined against illumination distortion in different levels
According to following table that shows number of matched features, SURF without external detector is the best combination when handle rotate and scale constraints in different levels. Harris/FREAK AND FAST/SURF is best combination when handle illumination constraints in different levels.

Table 12: illustrate industrial scene average matched features under different level of constraints.

Detectors/ Descriptor	Detected feature Points			
	Rotate 25	*Rotate 45*	*Rotate 75*	*Rotate 100*
SURF/FREAK	7.80	6.60	6.00	6.40
SURF/SURF	**12.20**	**12.20**	**10.40**	**9.60**
FAST/BRISK	3.60	3.00	2.80	2.80
	Scale 1.2	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF/FREAK	8.40	8.40	6.20	6.20
SURF/SURF	**10.20**	**10.60**	**8.40**	**8.80**
FAST/BRISK	2.20	4.20	2.60	3.60
	Illumination 1	*Illumination 2*	*Illumination 3*	*Illumination 4*
Harris/FREAK	**18.00**	13.20	**12.20**	10.80
FAST/SURF	17.60	**15.20**	11.20	**11.20**
FAST/BRISK	13.00	11.00	8.20	8.20

In industrial scene, the best combination between detector and descriptors is SURF without external detector in different levels of rotate, scale and illumination.

The following tables illustrate the best combination between detectors and descriptors and between various types of distortion, according to number of matched features.

Table 13: CALsuburb matched features under different level of rotate, scale and illumination distortion.

Detectors/ Descriptor	Detected feature Points			
	Rotate 25	*Rotate 45*	*Rotate 75*	*Rotate 100*
FAST/FREAK	5.40	6.80	9.20	9.80
FAST /SURF	8.20	9.40	12.00	11.20
FAST /BRISK	**12.00**	**11.20**	**15.20**	**11.20**
	Scale 1.2	*Scale 1.4*	*Scale 1.7*	*Scale 2*
SURF/FREAK	9.40	7.20	5.40	5.80
SURF/SURF	**11.20**	**10.60**	**8.40**	**7.20**
SURF/BRISK	3.60	3.20	2.80	2.80
	Illumination 1	*Illumination 2*	*Illumination 3*	*Illumination 4*
Harris/FREAK	14.00	11.60	9.80	9.00
FAST/SURF	**16.80**	**13.40**	**12.00**	**11.00**
FAST/BRISK	11.00	8.40	7.60	7.20

4. Conclusion.

The main purpose of this paper is to find the best combination between detectors and descriptors against different type of distortion (illumination) , rotate, scale constrains and in different levels. For this purpose, experiment studies the effect of combining detectors and descriptors on different scenes to find best combination. It figures out if a detector is based on scene or not. Experiment results show that a MinEigen detector has a best result in number of detected key-points when handle rotate, scale and illumination. That means it is not affected by scene.

SURF descriptor is built to be invariant to scale and rotate.

It is invariant to scale by sampled over a window that is proportional to the window size with that it was detected.
SURF is invariant to rotate finding the dominant direction of the feature and rotating the sampling window to align with that angle. It divides rotated neighborhood up to 16 sub squares, each one is divided into 4 squares. FREAK tends to fasten matching so it is not as good as SURF.

SURF without external detector is the best combined when handle rotate and scale constrains in different levels and scenes. FAST/SURF and Harris/FREAK is best combined against illumination distortion in different levels.

.

References

[1] J. Bauer, H. Bischof, A. Klaus, K. Karner, “Robust and fully automated image registration using invariant features,” ISPRS, DVD Proc, pp. 12–23,2004.

[2] A.Berg, T. Berg, J. Malik,” Shape matching and object recognition using low distortion correspondence,” in: Proceedings of the IEEE International Conference Computer Vision Pattern Recognition, vol. 1, pp. 26–33, 2005 .

[3] G. Dorko, C. Schmid, “Selection of scale-invariant parts for object class recognition,” in: Proceedings of the IEEE International Conference on Computer Vision, vol. 1, pp. 634–639, 2003.

[4] S. Lazebnik, C. Schmid, J. Ponce, “A sparse texture representation using local affine regions,” IEEE Trans. Pattern Anal. Mach. Intell. , vol. 27 (8),pp. 1265–1278,2005.

[5] S. Se, D. Lowe, J. Little, “Vision-based mobile robot localization and mapping using scale-invariant features,” in: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2051–2058, 2001.

[6] J. Sivic, F. Schaffalitzky, A. Zisserman, “Object level grouping for video shots,”Int. J. Comput. Vis., vol. 67 (2),pp. 189–210, 2006.

[7] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol.60(2), pp. 91–110,2004.

[8] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, “SURF: Speeded Up Robust Features, “Computer Vision and Image Understanding, vol. 110(3), pp. 346–359,2008.

[9] Guo, Lisha, L.Junshan, Z.YingHong, and Z.Tang. “A novel Features from Accelerated Segment Test algorithm based on LBP on image matching.” In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, pp. 355-358, 2011.

[10] Leutenegger, Stefan, M.Chli, and R.Siegwart, “BRISK: Binary robust invariant scalable keypoints.”, In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548-2555. IEEE, 2011.

[11] C. Harris and M. Stephens, “A combined corner and edge detector,” in Proceedings of the 4th Alvey Vision Conference, pp. 147–151,1988.

[12] A. Alahi, R. Ortiz, P. Vandergheynst,” FREAK: Fast Retina Keypoint,” In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[13] E. Nowak, F. Jurie, B. Triggs, “Sampling strategies for bag-of-features image classification,” in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3954, pp. 490–503, 2006.

[14] Agarwal, Ankur, and B. Triggs. “Hyperfeatures–multilevel local coding for visual recognition,” In Computer Vision–ECCV 2006, pp. 30-43. Springer Berlin Heidelberg, 2006.

[15] H. Moravec, “Towards automatic visual obstacle avoidance,” in: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 584, 1977.

[16] A. Johnson, M. Hebert, “Object recognition by matching oriented points, ‘in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 684–689, 1997.

[17] Mehrotra, Hunny, Banshidhar Majhi, and P.Gupta. "Robust iris indexing scheme using geometric hashing of SIFT keypoints." Journal of Network and Computer Applications 33, no. 3, pp.300-313 , 2010.

[18] Piccinini, Paolo, A.Prati, and R.Cucchiara. "Real-time object detection and localization with SIFT-based clustering." *Image and Vision Computing* 30, no.8 ,pp. 573-587, 2012.

[19] Zhou, Huiyu, Y.Yuan, and C.Shi. "Object tracking using SIFT features and mean shift." Computer vision and image understanding 113, no. 3,pp. 345-352, 2009.

[20] Y. Nig, L. Tun and Z. Hong "Robust-description-method-sift-features-license-plate-characters," Technology journal , 2011.

[21] M. Zahedi, S. Salehi, "License Plate Recognition System Based on SIFT Features," Procedia Computer Science, Vol. 3, 2011.

[22] F. Silva, A. Artero, M. Paiva and R. Barbosa , "ALPRS - A New Approach For Licenses Plate Recognition Using The SIFT Algorithm," An International Journal (SIPIJ) Vol.4, No.1, 2013.

[23] Zhu, Yingxuan, S.Cheng, V.Stanković, and L.Stanković. "Image registration using BP-SIFT." Journal of Visual Communication and Image Representation 24, no. 4, pp. 448-457, 2013.

[24] Liao, Kaiyang, Guizhong Liu, and Youshi Hui. "An improvement to the SIFT descriptor for image representation and matching." *Pattern Recognition Letters*34, no. 11 (2013): 1211-1220.

[25] Li, Yanshan, Weiming Liu, Xiaotang Li, Qinghua Huang, and Xuelong Li. "GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra." Information Sciences (2013).

[26] Kang, Minku, Wonkook Choo, and Seungbin Moon. "Improved face recognition algorithm employing SURF descrpitors." In *SICE Annual Conference 2010, Proceedings of*, pp. 2511-2513. IEEE, 2010.

[28] Lee, Yong-Hwan, Hyochang Ahn, and Sang-Burm Rhee. "Efficient Image Retrieval Using Advanced Clustering SURF." In Network-Based Information Systems (NBiS), 2012 15th International Conference on, pp. 749-753. IEEE, 2012.

[29] Pan, Jie, Wenjie Chen, and Wenhui Peng. "A new moving objects detection method based on improved SURF algorithm." In Control and Decision Conference (CCDC), 2013 25th Chinese, pp. 901-906. IEEE, 2013.

[30] Juan, Luo, and Oubong Gwun. "SURF applied in panorama image stitching." InImage Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on, pp. 495-499. IEEE, 2010.

[31] Huang, Hui, Lizhong Lu, Bin Yan, and Jian Chen. "A new scale invariant feature detector and modified SURF descriptor." In Natural Computation (ICNC), 2010 Sixth International Conference on, vol. 7, pp. 3734-3738. IEEE, 2010.

[32] Do, Yong-Sig, and Yong-Jin Jeong. "A new area efficient SURF hardware structure and its application to Object tracking." In TENCON 2013-2013 IEEE Region 10 Conference (31194), pp. 1-4. IEEE, 2013.

[33] Huiqing, Zhang, and Gao Lin. "Image registration research based on SUSAN-SURF algorithm." In Control and Decision Conference (2014 CCDC), The 26th Chinese, pp. 5292-5296. IEEE, 2014.

[34] Fan, Peng, Aidong Men, Mengyang Chen, and Bo Yang. "Color-SURF: A surf descriptor with local kernel color histograms." In Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on, pp. 726-730. IEEE, 2009.

[35] Li, Haiyan, Tingrong Xu, Jie Li, and Lixiao Zhang. "Face recognition based on improved surf." In Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications, pp. 755-758. IEEE Computer Society, 2013.

[36] Du, Mingfang, Junzheng Wang, Jing Li, Haiqing Cao, Guangtao Cui, Jianjun Fang, Ji Lv, and Xusheng Chen. "Robot robust object recognition based on fast SURF feature matching." In Chinese Automation Congress (CAC), 2013, pp. 581-586. IEEE, 2013.

[37] Z.Wang, F.Yan ; Y.Zheng,"An adaptive uniform distribution surf for image stitching," Image and Signal Processing (CISP), 2013 6th International Congress on ,Vol 2,pp 1-10,2013

[38] J Krizaj, V Struc, S Dobrisek, D.Marcetic, and S.Ribaric. "SIFT vs. FREAK: Assessing the usefulness of two key-point descriptors for 3D face verification." In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on, pp. 1336-1341. IEEE, 2014.

[39] C.Whiten, R.Laganiere, and G. Bilodeau. "Efficient action recognition with mofreak." In *Computer and Robot Vision*

[27] Zhou, Zhiheng, Xiaowen Ou, and Jing Xu. "SURF feature detection method used in object tracking." In Machine Learning and Cybernetics (ICMLC), 2013 International Conference on, vol. 4, pp. 1865-1868. IEEE, 2013.

(CRV), 2013 International Conference on, pp. 319-325. IEEE, 2013.

[40] http://qixianbiao.github.io/Scene.html.