New 2D Feature Descriptor Free from Orientation Compensation with k-Means clustering

Manel BENAISSA, Abdelhak BENNIA*

Department of electronic, University of Constantine 1, 325 Route de Ain El Bey, Constantine, Algeria

*Corresponding Author: Manel BENAISSA (email: menelbenaissa@gmail.com)
(Received: 01-November-2018; accepted: 28-December-2018; published: 31-December-2018)
DOI: http://dx.doi.org/10.25073/jaec.201824.211

Abstract. *In this paper, we propose two novel approaches in the field of feature description and matching. The first approach concerns the feature description and matching part, where we proposed an orientation invariant feature descriptor without an additional step dedicated to this task. We exploited the information provided by two representations of the image (intensity and gradient) for a better understanding and representation of the feature point distribution. The provided information is summarized in two cumulative histograms and used in the feature description and matching process. In the context of object detection, we introduced an unsupervised learning method based on k-means clustering. Which we used as an outlier pre-elimination phase after the matching process to improve our descriptor precision. Experiments shown its robustness to image changes and a clear increase in terms of precision of the tested descriptors after the pre-elimination phase.*

Keywords

Feature understanding; Feature description; Feature matching; object detection; k-means clustering.

1. Introduction

Feature description and image matching are two important problems in machine vision and robotics. An ideal feature descriptor should be robust to image transformations such as scale, illumination, rotation, noise and affine transformations. The ability to match corresponding points between two or more images of a scene is an important component of many computer vision tasks such as structure from motion [1], visual SLAM (simultaneous localization and mapping) [2], object recognition, image registration, 3D reconstruction and object tracking. Scale-invariant feature transform (SIFT) introduced by Lowe [3] is a successful approach in the field of feature detection and description, several variants and extension were proposed to improve its computational complexity like in [4], [5]. The speeded-up robust features (SURF) [6] is also based on local histograms of gradient orientations, which uses integral image representations to speed up the computation. Binary robust independent elementary feature (BRIEF) [7], is one of the proposed alternatives for SIFT which requires less complexity, with almost similar matching performance. Rublee et al. proposed the oriented FAST and rotated BRIEF (ORB) [8], the binary robust invariant scalable keypoints (BRISK) [9], and fast retina keypoint (FREAK) [10] are either

good examples. Binary descriptors are robust to local changes and computationally efficient since the Euclidean distance has been replaced by Hamming distance. However, they remain sensitive to small disturbance of sample point's locations added to the fact that pairwise intensity comparisons capture very limited information of a local image region. Moreover, most of the local binary features are handcrafted, that require strong prior knowledge and are heuristic. Shape based feature descriptors where also widely studied and used. Such as, DAISY [11], LIOP [12] and GSURF [13]. Several learning approaches have been proposed based on convolutional neural networks (CNNs). Such as AlexNet [14], VGG [15], GoogLeNet [16] and ResNet [17]. Recently, multiple combinations of the CNN's discriminative power with low-cost computational binary descriptors were proposed for multiple applications. Such as DeepBit [18], where compact binary descriptors are unsupervisedly learned and achieved the state-of-the-art of binary feature descriptors. Optimization approaches have been proposed in order to reach or outperform state-of –the-art binary feature descriptors such as in [19], where authors proposed an online adaptive binary descriptor, optimized for each image patch independently. Or in [20], where authors proposed a general-purpose learning to rank formulation that optimizes local feature descriptors for nearest neighbor matching. In [21], a context-aware local binary feature learning method has been proposed for face recognition applications. A supervised convolutional replacement of SIFT was proposed by [22], which is a pipeline with feature point detection, orientation estimation and feature description. In [23], authors proposed a self-supervised feature detector and descriptor that operates on full-sized images and jointly computes pixel-level interest point locations and associated descriptors in one forward pass. Even if the efficiency of learning methods isn't debatable, the need of an efficient and costly training phase in addition to the necessary availability of large annotated datasets in order to reach traditional state-of-the-art feature descriptors performances remain a drawback. Moreover, their important sensitivity to orientation changes is a real inconvenient for real word vision application. Therefore, an additional step dedicated to the orientation estimation is added to the pipeline in order to remedy to this inconvenient. Our main objective over this work is to propose an orientation invariant feature descriptor, which doesn't require any additional step dedicated to this task.

We propose an efficient histogram based orientation invariant feature descriptor, based on a simple combination of intensity and gradient images. Unlike state-of-the-art, our descriptor is free from the orientation estimation and compensation step, since its structure offers him the invariance property. Experimental results show the efficiency of our descriptor against rotation, JPEG compression, viewpoint, multiview and deformation changes. That makes it well adapted for Multiview or surveillance cameras. Moreover, we proposed an unsupervised outliers pre-elimination method in order to enhance the matching accuracy of our descriptor. Based on k-means clustering, this step was added in the context of object detection, after the matching process and before the RANSAC operation. We evaluated our descriptor on multiple datasets and compared it to the most used state-of-the-art feature descriptors and the obtained results show the efficiency of the proposed method.

This paper is organized as follow. Section 2 is dedicated to related work. A presentation of our descriptor in Section 3. Followed by experimental results in Section 4 and conclusions in Section 5.

2. Related work

2.1. Rotation invariant descriptors

An efficient estimation of a discriminative orientation is a critical step that affect the matching results as shown in [43]. However, it hasn't received consequent attention as in the case of feature detection or description. The estimation method introduced by SIFT is then actually the dominant solution for assigning an orientation to a feature point. A small improvement was introduced in ORB, by using the intensity centroid in order to speed-up the estimation process. The drawback of this method as pointed

out by [44] is its fragility against arbitrary positions, which negatively affect the descriptor performances [27]. Some rotation invariant descriptors have then been proposed in the literature such as, MROGH [45], which uses intensity order pooling with rotation invariant gradients. LIOP [12] applied another approach on the same way in order to aggregate the gradient information. Even if BRISK [9] and FREAK [10] claims to be rotation invariant, they remain dependent of the orientation estimation that is included in the descriptor extraction process. Learning-based descriptors still rely on the orientation estimation of the used feature detector. An approach has been proposed by [46], that used Deep Learning to predict stable orientations which result to significant gain over state-of-art. However, it still an additional step dedicated to independently estimating the feature point orientation.

2.2. Feature Matching

A correspondence between two patches or patch matching is an important process in many computer vision applications, such as multi-view reconstruction, object recognition or structure from motion. The SIFT approach and its variants [47], [48] are based on conventional distance, e.g Euclidian distance to measure the similarity between two patches. These methods remain largely dependent on human expertise and does not provide optimal solution. Recently, numerous learning based approaches have been proposed [34], [40] in order to adapt similarity functions for given datasets.

2.3. k-Means clustering with Outlier Removal

The main objective over data clustering is the identification of homogeneous clusters over a set of objects. Authors in [49] made an interesting work by summarizing several outliers detection techniques. In [50], Yu et al. proposed the OEDP k-means, where outliers are removed from dataset before applying the k-means algorithm. Authors in [51], proposed a method that choose initial centers that are not outliers by using two initialization methods. In [52], authors used an additional cluster to the k-means algorithm to eliminate outliers. The CHB-K-Means [61], detected outliers by using attribute-weighted matrix.

3. Proposed keypoint descriptor

We first used the MSER [53] detector in order to get the feature point positions. MSER is a blob detector, this last extract from the image a number of covariant regions, called MSER: an MSER is a connected stable component of a few sets of a gray level image. This one is based on the idea of taking regions that remain almost the same across a wide range of thresholds. So,

- All pixels below a given threshold are blank and all those are equal or above are black.
- If we are shown a sequence of thresholded images I_t. where the threshold is defined by t, a black image will appear first, then white spots corresponding to the minimum intensity will appear and then increase.
- These white dots will eventually merge, until the entire image is white.
- All of the connected components in the sequence are the set of all the extremal regions.

Optionally, elliptical frames are attached to MSERs by inserting ellipses into regions. These regions are retained as features for the descriptors. Word extremal refers to the property that all pixels within the MSER have (regions brilliant extremes) or inferior (dark extremal regions). For the description part, we took advantage of image intensity and gradient information to get a better understanding and description of the feature point and its surroundings distribution.

3.1. Description

Motivation

Our motivation was to find another representation of the selected patches in order to compare them without any orientation estimation

and compensation process. In this optic, we used two bi-dimensional histograms containing the intensities and the gradient magnitudes and orientations of the feature point surroundings. We made the choice of using bi-dimensional histograms in order to capture the intensity and gradient orientation change around the patch edges since only the position of the patch edges change in the case of orientation transformation. Considering the gradient magnitude as the best way to localize image edges, we used it in both intensities and gradient orientations histograms.

We chose to use a two-dimensional histogram of intensity and gradient to capture the most important intensity changes around the edges of the patch, as experiments show that better results in terms of pairing accuracy were obtained using two-dimensional histograms.

Histograms creation process

The first step of the histograms creation process is to compute the gradient magnitude I_m and gradient orientation I_θ images from the original image I, such as:

$$I_m(x,y) = \Big[(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2\Big]^{1/2} \quad (1)$$

$$I_\theta(x,y) = \tan^{-1}\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)} \quad (2)$$

Each keypoint and its surroundings in I, is described using three image patches, which are gradient orientations patch P_θ, gradient magnitudes patch P_m and intensities patch P_I, all centered at the keypoint position (x_i,y_i) and respectively extracted from I_θ, I_m and I.

We quantified the original values in the three patches as shown in Fig.1. Intensities and gradient magnitudes are ranged over five values (1-5), corresponding to the lowest until highest intensity values and from weakest to strongest edges in the case of gradient magnitudes. The gradient orientations are ranged over eight directions (22°-337°), with a shift of 45°.

The next step is the creation of the two cumulative histograms $H_{I,m}$ and $H_{\theta,m}$, which will constitute our descriptor such as:

- $H_{\theta,m}$ Contains the gradient orientations and magnitudes of P_θ andP_m.

- $H_{I,m}$ Contains the intensities and gradient magnitudes of P_I and P_m.

Fig.2 shows three examples of the obtained histograms from different regions of the moon surface. Intensities are ranged over the X-axis, gradient magnitudes on the Y-axis and Z-axis contain the cumulative votes of all the elements with the same (intensity, magnitude) values from the patches (P_I and P_m). The same goes for $H_{\theta,m}$, where the X-axis is now containing the gradient orientations.

By observing the intensities patch P_I^1, we can clearly see that it is composed of two dominant black, white and a small gray areas. P_m^1 contains one clear edge surrounded by a black region. This distribution is perfectly reflected in H^1, where the intensities are mainly distributed over two values which are one and five, with some votes for four representing the gray region. The gradient magnitudes are essentially distributed around one representing the black region and some votes are assigned to higher values corresponding to the unique edge in the patch.

In the case of P_I^2, intensities fluctuate between certain black and gray regions without a clear separation between them. P_m^2 is also composed of small fluctuant edges with different magnitudes. As in the first case, the votes of intensities and magnitudes are distributed over multiple small values in H^2, corresponding to the patch description presented above.

Finally, in P_I^3 we can clearly see that it is essentially composed of three gray, white and black regions. The patch of gradient magnitudes, P_m^3 is composed of several small edges. This patch description is well summarized in the last histogram H^3, where most of the votes are divided into three values namely three, four and five representing the three previous intensity regions. On the other hand, the gradient magnitudes are distributed over different low values between one and three.

Fig. 1: Patches content quantification and histograms creation process.

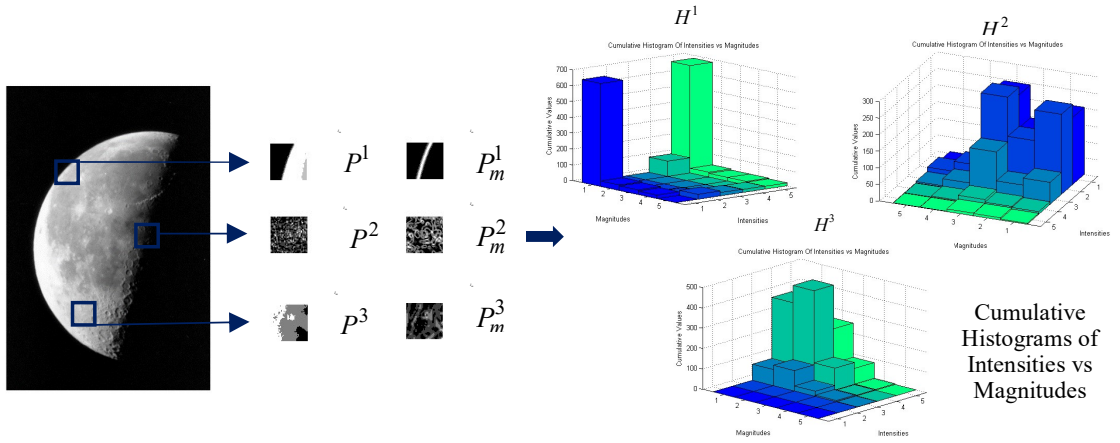

Fig. 2: Three Histograms obtained from different regions of the moon surface.

Orientation invariance property

By observing patches from the original image in Fig.3 and its rotated version. We remarked that the edges and intensities distribution is the same over the two patches, only their positions changed.

Our intuition was to say that, even if we perform a rotation on the original image. The edges distribution over it will remain unchanged and by the same, the intensities and gradient orientations distribution around them won't change either. This was confirmed by the obtained histograms from Fig.3 patches, where we can clearly see in Fig.4, that the obtained histograms of the feature points in the original and rotated images are the same.

3.2. Matching

In the matching part, our objective is to get the highest percentage of similarity between the selected keypoint pairs by comparing their histograms ($H^1_{\theta,m}$, $H^1_{I,m}$) and ($H^2_{\theta,m}$, $H^2_{I,m}$). Respectively obtained from the test I^1 and reference I^2 images. We performed a subtraction operation between the pairs of histograms in order to get their similarity scores. The resulting histograms are respectively given by, $H_{\theta,m}$ and $H_{I,m}$.

We consider that, if the j^{th} element in the resulting histograms $h^j_{\theta,m} \in H_{\theta,m}$ and $h^j_{I,m} \in H_{I,m}$ is less than its corresponding thresholded element in $H^1_{\theta,m}$ and $H^1_{I,m}$, such as: $h^j_{\theta,m} <^? Th * h^{1,j}_{\theta,m}$ and $h^j_{I,m} <^? Th * h^{1,j}_{I,m}$ with $Th < 1$, we add one to the matching scores (S_θ and S_p).

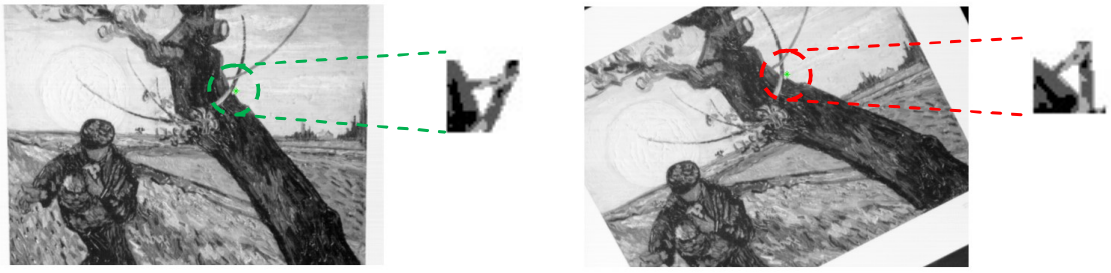

Fig. 3: A comparison of two patches surrounding the same feature point from the original image and its rotated version from VanGogh dataset.

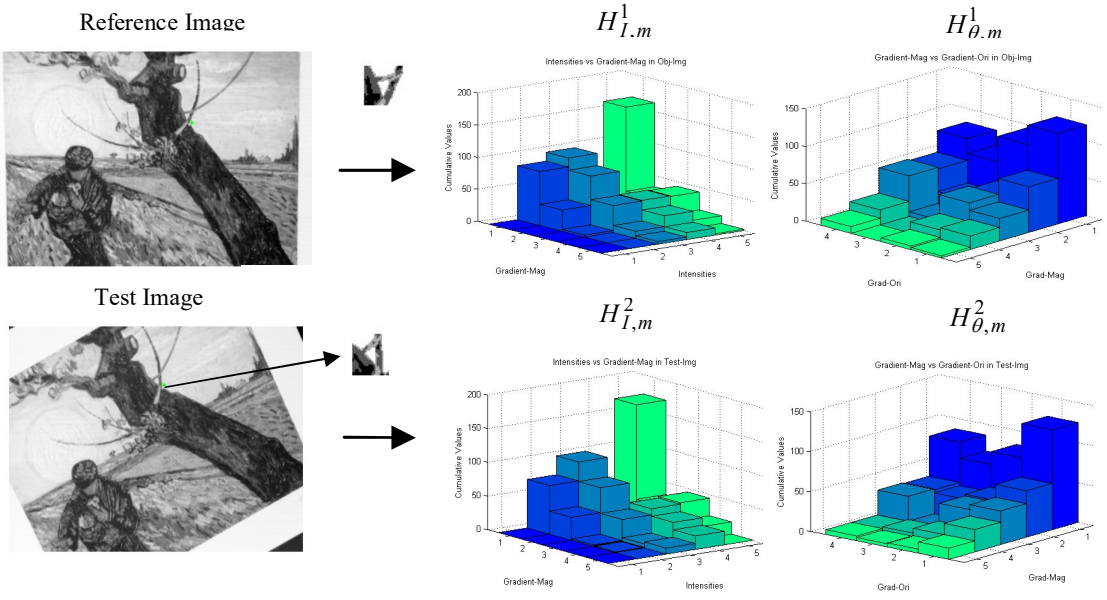

Fig. 4: Resulting histograms from two patches surrounding the same feature point in the original image and its rotated version.

In other terms, if we subtract the histograms ($H^2_{\theta,m}$, $H^2_{I,m}$) of the reference image from ($H^1_{\theta,m}$, $H^1_{I,m}$), the corresponding histograms of test image. A perfect match should correspond to resulting histograms ($H_{\theta,m}$ and $H_{I,m}$) with all zeros elements. Or, at least all the resulting elements ($h_{\theta,m}$ and $h_{I,m}$) are less than their corresponding thresholded elements ($h^1_{\theta,m}$, $h^1_{I,m}$) in the histograms of test image. The matching scores are then given by:

$$\begin{cases} S_I(\%) = \left[\sum_j (h^j_{I,m} < Th * h^{1,j}_{I,m})/N_I\right] * 100 \\ S_\theta(\%) = \left[\sum_j (h^j_{\theta,m} < Th * h^{1,j}_{\theta,m})/N_\theta\right] * 100 \end{cases} \tag{3}$$

Where, N_θ and N_I are respectively, the total number of elements in $H_{\theta,m}$ and $H_{I,m}$. The final score of a given keypoint pair is the mean of the obtained matching scores $S_\theta(\%)$ and $S_I(\%)$, such as:

$$S_F(\%) = mean\ (S_\theta, S_I) \tag{4}$$

We consider in the context of our work that a correspondence between two points is correct if the final score is greater than the matching threshold$ThM(\%)$, which we fixed to 70%. Such as:

$$\begin{cases} if \quad S_F(\%) \geq ThM(\%), & Correct\ Match \\ if \quad S_F(\%) < ThM(\%), & Incorrect\ Match \end{cases} \tag{5}$$

We summarized the proposed method above in the algorithm 1, such as.

A perfect match is equivalent to a final score (or percentage of similarity) of 100%. In order to illustrate the matching process in a better way, we illustrated it on the example of Fig.4,

We can clearly see in Fig.6 that all the elements $h^j_{\theta,m} \in H_{\theta,m}$ and $h^j_{I,m} \in H_{I,m}$ in the resulting histogram pair are near to zero. Or at least all the resulting elements are less than their corresponding thresholded elements in the histograms of test image, which corresponds to a perfect match. If two or more feature points in the test image, match a single point in the reference image, we keep the pair with the best match and discard the other(s).

Fig.7 shows some visual results obtained by our descriptor under different angle changes and without any orientation compensation step.

For a first attempts, we used a simple difference between histograms in the matching part. That said, there are other much more sophisticated methods for doing histograms comparison such as Kullback Leibler divergence or Earth-movers-distance, which we will explore for further searches.

3.3. Object detection

3.3.1 Scaling Process

As in the case of SIFT, we used pyramids to get the scale with one difference. In our case, we directly applied them on the patch level where the SIFT descriptor use image pyramids. We applied different scales on the patch of object image. As shown in Fig.8, we increased the patch size surrounding the key point to cover a larger space in the image. We then resized it to the initial patch size by down sampling for the matching operation.

This method allows us to get better results in the matching part as shown by the example in Fig.8, where we can see the similarity between the histograms of the selected keypoints pair. For each pair, we applied three different scales [$Scale_1$, $Scale_2$, $Scale_3$] on the reference image patch and we keep the best matching score from the obtained results using these scales.

3.3.2 Outliers Pre-elimination Process

In order to achieve better object detection results, while preserving the simplicity of our system. We added a pre-elimination phase based on the unsupervised learning method of K-means, in order to eliminate false matches. The clustering method K-means clustering (MacQueen, 1967-kmeans) is a commonly used method to automatically partition a dataset in k groups. It proceeds by selecting k initial groups, then these are refined from iteratively as follows:

1. Each group consists of several instances d_i.

2. The center of each group C_j is updated to become the average of its instances constituent.

Then, the algorithm converges when there is no more change in the assignment of instances to clusters.

The main objective of this operation in our case is to segment the obtained matches in test image to k regions (k groups of correspondences) with a probability for each region to contain the object. This step is performed after the matching process and before the RANSAC test between the two images. Fig.9 shows an example of this process, for $k = 3$.

The choice of k = 3 was made empirically after an important number of tests, where we found that the best results were obtained with it. That said, this parameter will be the subject of further investigation in future work.

We estimate that the group with the highest number of good matching scores defines the most likely region containing the object in the test image. The rest of groups are then automatically eliminated. This helps to squeeze out a considerable amount of false matches.

Algorithm 1: Description and Correspondence prcess

Input:

The intensity image I
The patch size s
Total number of elements in the histograms N_I and N_θ.

Description :

1: Compute the gradient orientation and magnitude I_θ, I_m of the input image I.
2: Find the feature points positions (x_i, y_i) by the MSER detector.
For each feature point do
Extract the patches P_θ, P_m and P_I of size (s x s), surrounding the feature point at position (x_i, y_i), from I_θ, I_m and I.
Quantify the selected patches P_θ, P_m andP_I.
Construct the cumulated histograms $H_{I,m}$ and $H_{\theta,m}$ such as:
For i : 1 to *s* do
For k : 1 to *s* do
$H_{I,m}$ (P_I (i,k), P_m (i,k)) = $H_{I,m}$ (P_I (i,k), P_m (i,k)) +1;
$H_{\theta,m}$ (P_θ (i,k), P_m (i,k)) = $H_{\theta,m}$ (P_θ (i,k), P_m (i,k)) +1;
endfor
endfor
endfor

Correspondence :

1: Subtract the histograms of the reference image ($H^2_{\theta,m}$, $H^2_{I,m}$) from the test image histograms ($H^1_{\theta,m}$, $H^1_{I,m}$) giving the resulting histograms ($H_{\theta,m}$ and $H_{I,m}$).
2: Compute the correspondence scores S_θ and S_p such as :
For i : 1 to size ($H_{I,m}$,1) do
For j : 1 to size ($H_{I,m}$,2) do
If $H_{I,m}$ **(i , j)** $<=$ ***Th*** * $H^1_{I,m}$**(i , j) do**
P_I = P_I + 1;
endIf
endfor
endfor
S_I = (P_I / N_I) * 100;
For i : 1 to size ($H_{\theta,m}$,1) do
For j : 1 to size ($H_{\theta,m}$,2) do
If $H_{\theta,m}$ **(i , j)** $<=$ ***Th*** * $H^1_{\theta,m}$ **(i , j) do**
P_θ = P_θ + 1;
endIf
endfor
endfor
S_θ = (P_θ / N_θ) * 100;
3: Compute the final score S_F, which is the mean of the correspondence scores S_θ and S_p.
4: Classify the correspondence as a correct or false match by comparing the final score to the correspondence threshold *ThM*.

Fig. 5: The proposed method algorithm.

This step is ignored in the case of image alignment. Experimental results showed a clear increase in accuracy using this module.

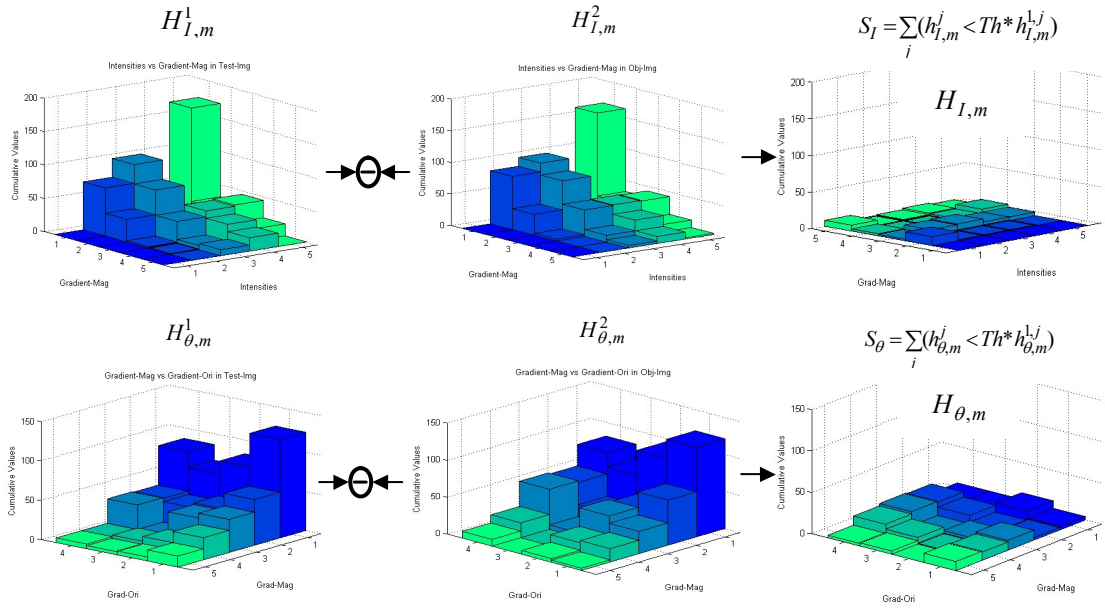

Fig. 6: Example showing the matching process of our descriptor.

Fig. 7: Visual results obtained by our descriptor under rotation change, without any orientation estimation step.

4. Experimental results

4.1. Image matching

We tested our descriptor in comparison to the widely used SIFT and SURF descriptors. We also compared it to BRIEF descriptor, which is free from orientation assignment. As ours, DAISY is a histogram-based descriptor, we then added it to the comparison process. Finally, we compared our descriptor to a learning based descriptor [36]. For simplicity, we named our descriptor ADOCH, for absolute difference of cumulated histograms. Our tests were performed under Matlab R2015a.

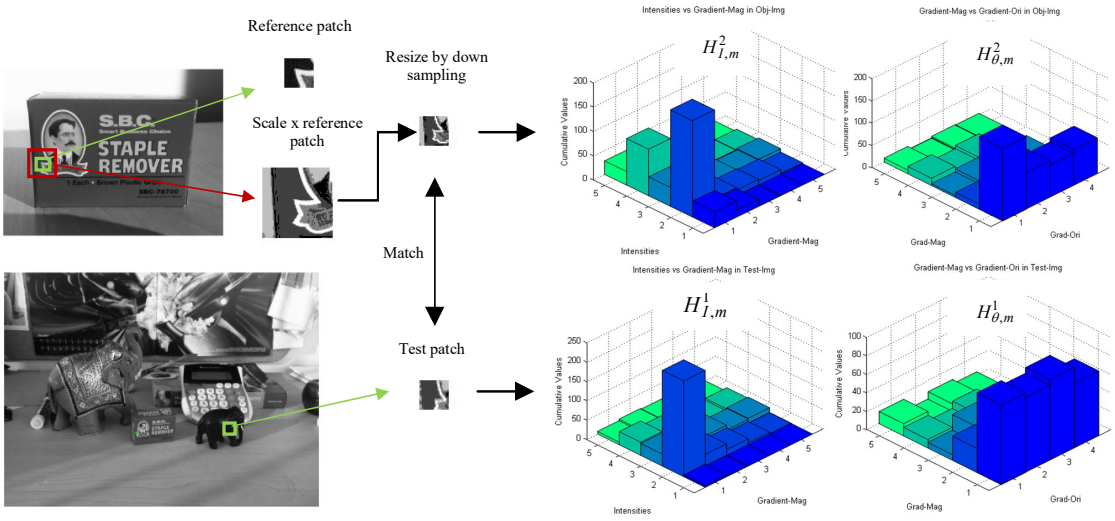

Fig. 8: Illustration of scaling process under the context of object detection.

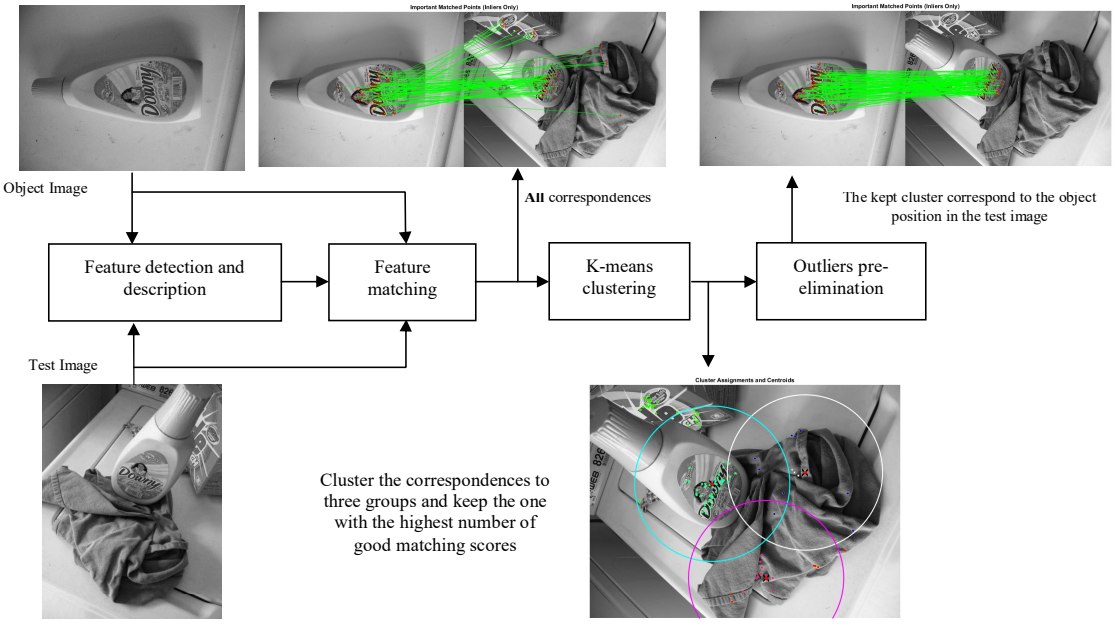

Fig. 9: Example illustrating the outliers pre-elimination process.

We selected four popular datasets to test our descriptor. In each sequence, a known increasing amount of transformations are performed between the first and the rest of images. The first one is the well-known (*Oxford*) dataset introduced by Mikolajczyk and Schmid [54]. It contains image sequences with six to nine images of rotation, illumination, scale and blur change. Salzmann's dataset [55] has been used to evaluate our descriptor performance for 3D deformable objects. The Strecha's dataset [56] were applied for the multiview stereo case and Heinly's dataset [57] for illumination, pure camera rotation, and pure scale change. We also

computed the generation time of our descriptor comparing to the rest of descriptors.

The descriptors performances are highly related to the combination detector/descriptor, since some descriptors are more discriminant for blobs than corners. Nevertheless, the global ranking of their performances remain the same regardless to the selected detector.

In [58], [59], authors shows that MSER is the best affine-invariant region detector in terms of accuracy and repeatability. We then used it in combination of our descriptor, about 500 to 1000 keypoints were detected for all tests.

We empirically chosen the optimum patch size after multiple tests, which we fixed to $s = 34$ pixels. That said, an optimization of this parameter could be subject to further investigations in future works.

In the matching part, we fixed the threshold Th to, $Th = 0.25$ since we estimate that a resulting elements $h^j_{\theta,m} \in H_{\theta,m}$ and $h^j_{I,m} \in H_{I,m}$ from the subtraction operation that are 75% inferior to their corresponding elements $h^1_{\theta,m}$, $h^1_{I,m}$ in the test histograms are considered as near to zero. Therefore, we consider that if $h_{\theta,m}$<0.25*$h^1_{\theta,m}$ or $h_{I,m}$<0.25*$h^1_{I,m}$, this is equivalent to a very small difference between test and reference histograms, we then add one to the matching scores S_θ and S_p.

We used the recall vs 1-precision curves to evaluate our descriptor performances under different constraints like blur, brightness, rotation and scale change. Such that

$$\text{Re}\,call = \frac{Number\ of\ trueMatch}{Number\ of\ Correspondences} \tag{6}$$

$$1 - precision = \frac{Number\ of\ falseMatch}{Number\ of\ trueMatch\ +\ Number\ of\ falseMatch} \tag{7}$$

We estimate that a match is correct if the final correspondence score $S_F(\%)$ is superior to the similarity threshold, which we fixed to $ThM(\%) = 70\%$. Only few number of incorrect matches have a similarity score which is superior to the fixed threshold. Nevertheless, these ones are not quantized as true Matches. Fig.10 shows our descriptor performances on the (*Oxford*) dataset, the obtained results illustrate the resistance of our descriptor for different kind of changes.

Even if the sensitivity of the gradient magnitude to blur change affect negatively the performances of our descriptor. For the viewpoint, rotation and JPEG compression changes, our descriptor performance is high. Our descriptor also performs well under illumination and rotation-&-scale change.

Fig.11 shows that our descriptor performs well in the case of 3D deformable objects. In the case of deformable objects, the edges distribution over the patch isn't highly affected and the strong aspect of our descriptor is precisely the fact that it's based on histograms which contains all important changes around the patch edges. This property makes it very resistant to this kind of changes. The same goes for Fig.12, where it shows our descriptor results under multiview for the stereo case.

The obtained results are extremely satisfactory and even better than some state-of-the-art descriptors. Moreover, we should note in this case that our descriptor doesn't need any preprocessing phase such as pattern creation in the case of binary descriptors, training phase for CNN descriptors or the orientation computation and compensation step for the histograms-based descriptors.

Finally, we tested our descriptor for pure angel and scale changes on Heinly's dataset, as shown in Fig.13 our descriptor performs extremely well.

This not only confirms our first intuition for the rotation invariance property of our descriptor, but also shows its resistance to pure scale change.

Fig.15 shows some visual results of our descriptor on the four datasets proposed before. We can clearly see the performances of our descriptor under different types of changes.

4.2. Object recognition

For the last investigation, we tested our descriptor in the context of object detection with the

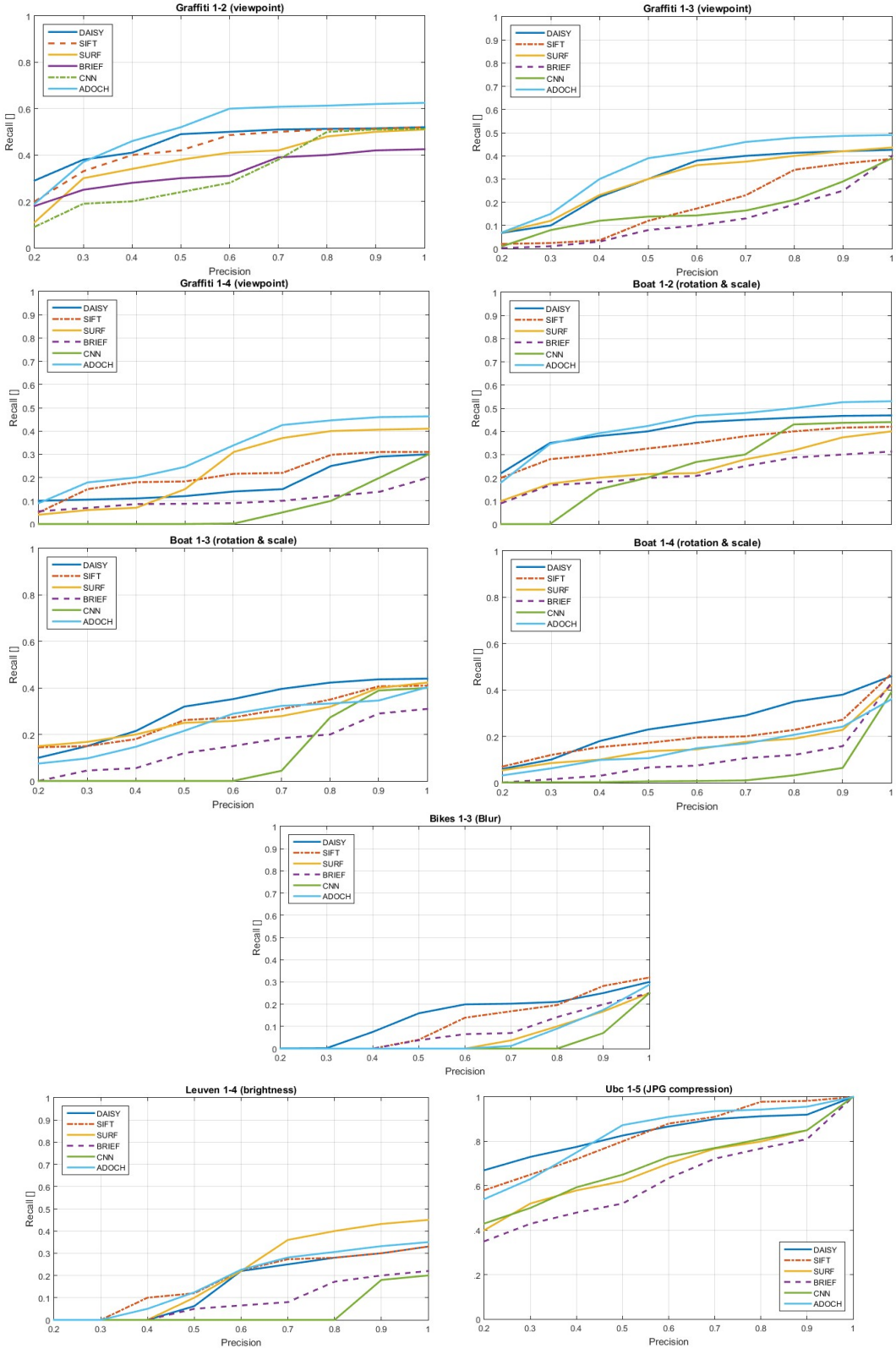

Fig. 10: Our descriptor performances on the (*Oxford*) dataset under view point (Graffiti), rotation and scale (Boat), blur (Bikes), luminance (Leuven) and JPEG compression change (Ubc).

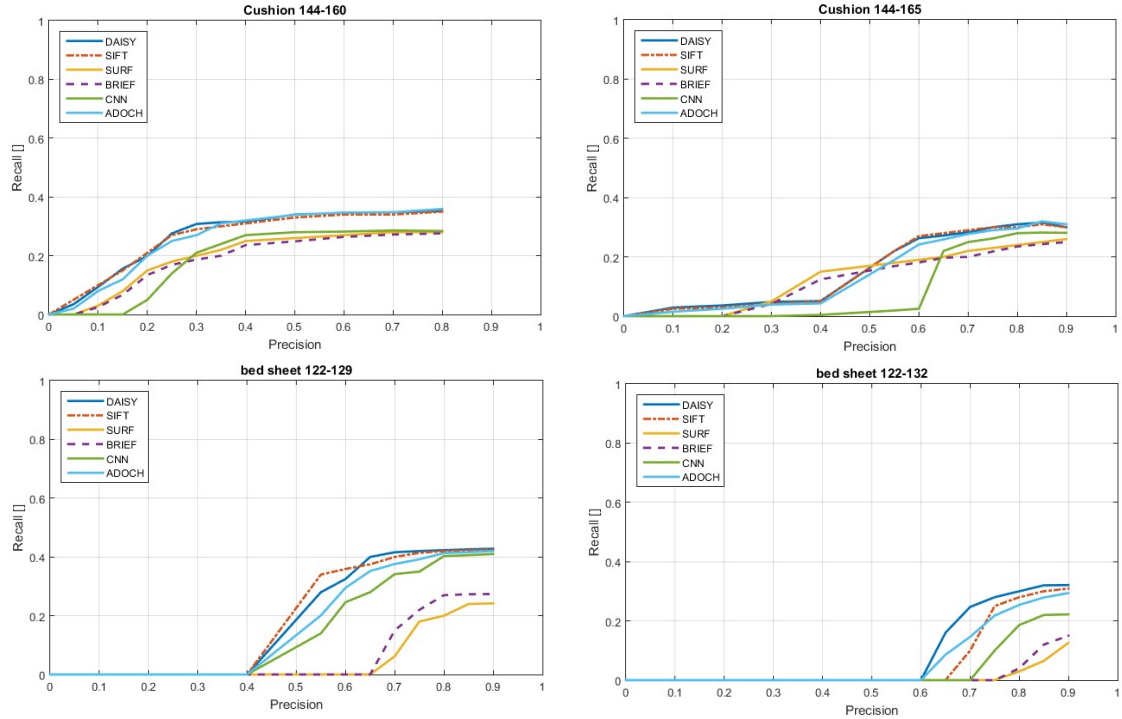

Fig. 11: Performance of our descriptor on the Salzmann's dataset for deformable objects.

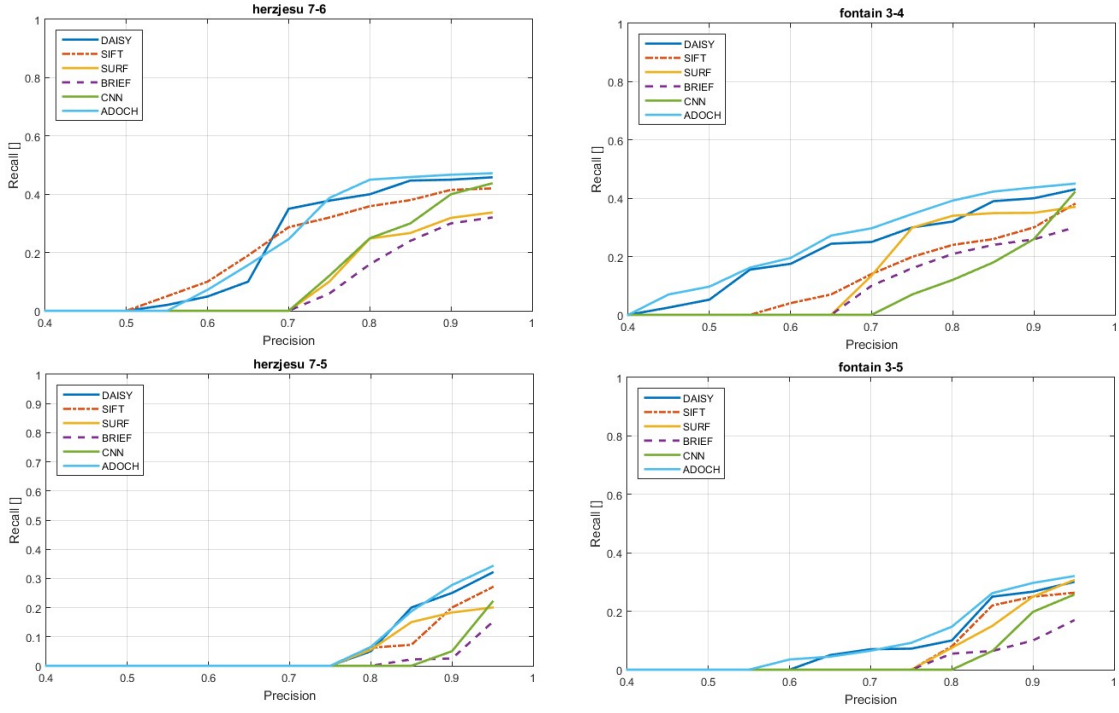

Fig. 12: Shows the obtained results from our descriptor on the multiview case.

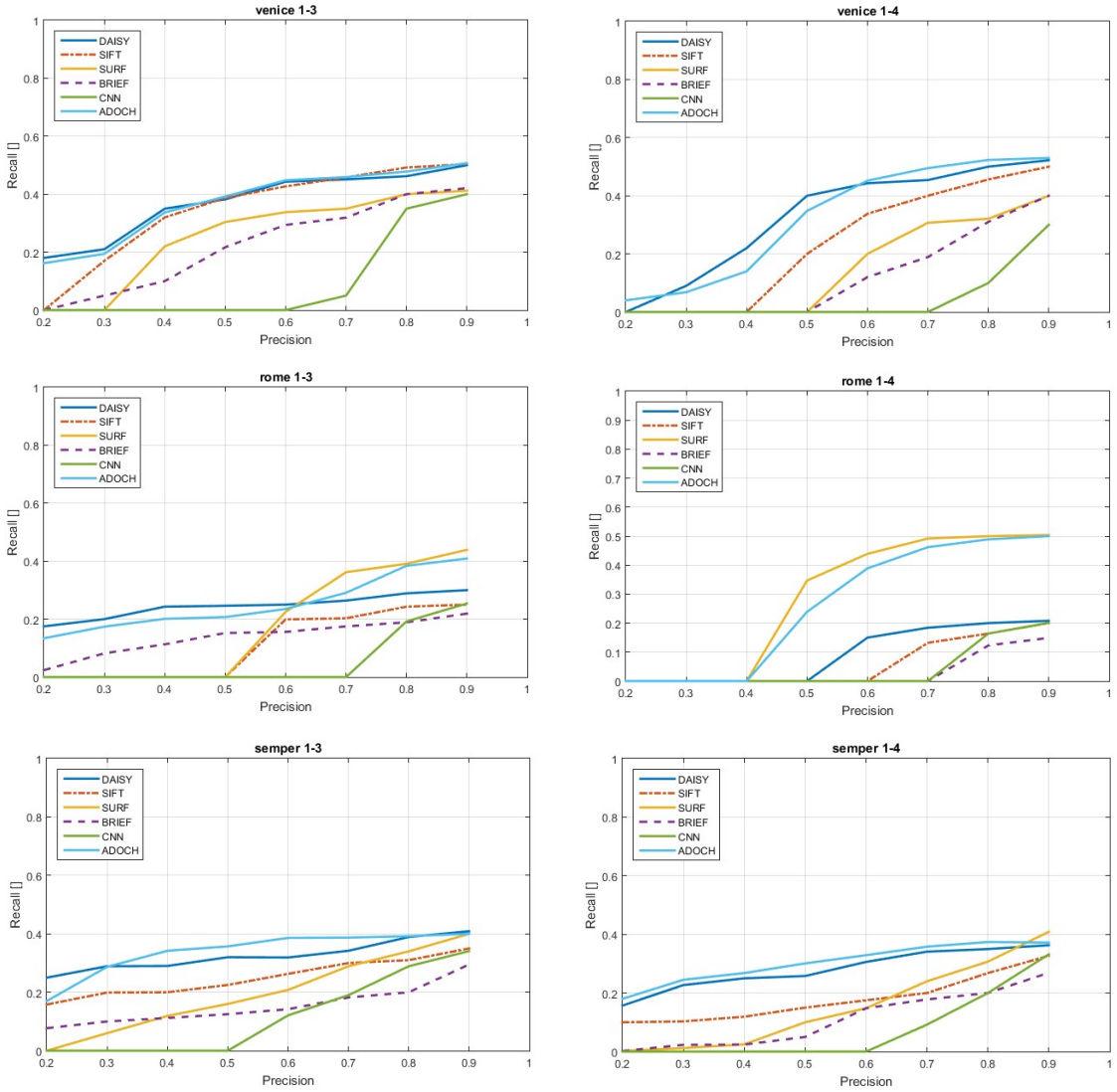

Fig. 13: Performances of our descriptor on the Heinly's dataset.

pre-elimination step. We tested our descriptor on two publicly available datasets, which are the 53 Objects [62] and Home Objects 06 [63] datasets composed of multiple objects with different declinations of each one of them. We also tested it on our own dataset that consist on real word images of multiple home objects under different f luminance, scale and orientation change, some samples images of it are shown in Fig.16.

In order to test the efficiency of the pre-elimination phase, we computed the accuracy rate of our descriptor, with and without this step. Such as,

$$\begin{cases} Object\ Accuracy\ Rate \\ = \dfrac{Number\ of\ objects\ correctly\ recognized}{Total\ number\ of\ declinations} \\ Total\ Accuracy\ Rate(\%) \\ = Mean\ (Object\ Accuracy\ Rate) \times 100 \end{cases} \tag{8}$$

Object accuracy rate (AR) reflect all the cases where the object is correctly detected under its different declinations. Total accuracy rate

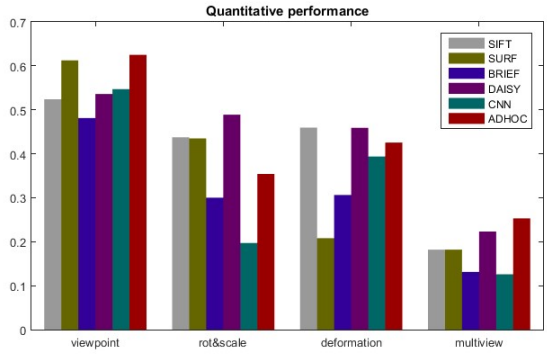

Fig. 14: Quantitative performance of the descriptors using the $F_{0.5}$.

(TAR) is the mean of all the obtained rates for all objects in the dataset.

These rates are computed with and without the pre-elimination phase. The obtained results are summarized in Tab.1, we took in this case $k = 3$.

Data set	TAR without clustering	TAR with clustering
Home Objects	46.425 %	**52.361 %**
53 Objects	32.913 %	**43.785 %**
Our dataset	45.325%	**53.784%**

Tab. 1: Total accuracy rates with and without clustering.

Tab.1 show an enhancement of about 10% in the total accuracy rate, which is considerable in terms of precision. The modularity of this step is its main advantage, since it can be added to any descriptor and can be activated for object detection and deactivated otherwise.

Fig.17 illustrate some visual results of our descriptor before and after the pre-elimination phase,

We can clearly see in Fig.17 that a considerable reduction of outliers have been performed after the clustering operation.

Table 2 contains comparison of descriptors computation time. As reference, we took the computation time of descriptors which were processed on similar or faster machines. The table contains timings per keypoint. All presented experiments were run on Intel Core i7-2720QM 2.2GHz, 16GB of RAM.

Results with * were obtained in experiments reported in this paper.

Name	Descriptor generation time (ms)
ADOCH	0.86*
SIFT	6.156 [11], 2.5 [12],
SURF	2.071 [46]
BRIEF	1.4 [12], 0.67 [27], 0.81 [46] 0.046 [46]
DAISY	0.012 [46]
CNN	-

Tab. 2: Descriptor computation time (per keypoint).

The obtained results show that our descriptor generation time is less than SIFT and near to the SURF descriptors. The BRIEF descriptor is faster, since that the sampling pattern creation time is not included here. The DAISY descriptor is also faster than ours. It is important to note that the orientation computation and compensation time is not included in the case for the SIFT, SURF and DAISY descriptors, whereas this step is skipped for our descriptor.

If we look to the ADOCH computation steps, we can clearly conclude that they are mainly independent, e.g., histograms can be processed independently, which makes the descriptor easy to parallel. Therefore, some further improvements in shortening the computation time are expected.

Experiments show that our descriptor is very efficient for high rotation, viewpoint, 3D deformable objects and JPEG compression changes. It also performs well in the case of blur and scale change. These properties makes it very attractive for large use applications such as, Multiview stereo vision or surveillance cameras, considering its ease of implementation.

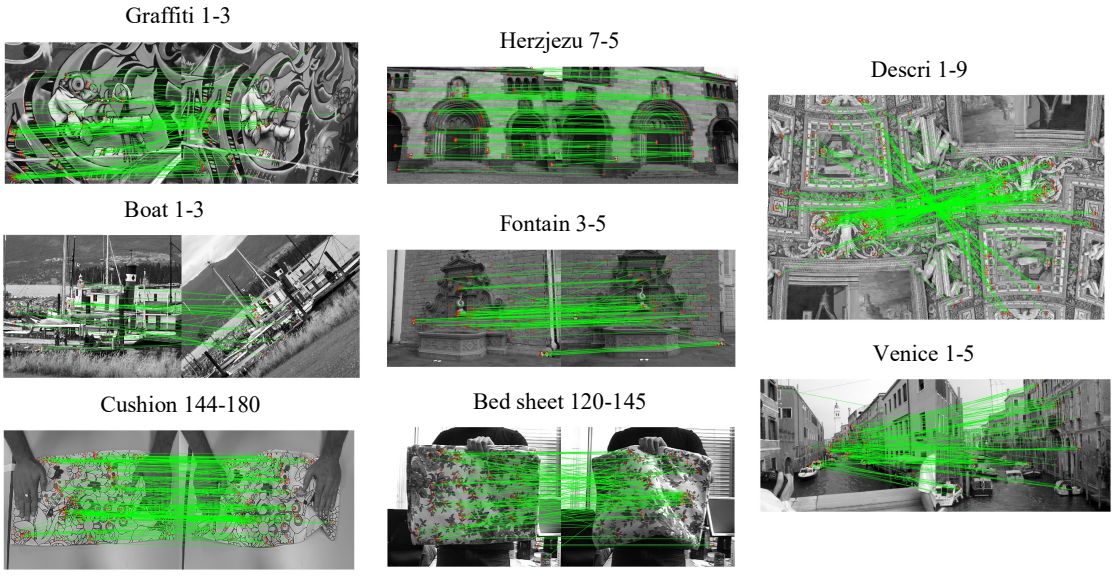

Fig. 15: Some visual performances of our descriptor on the four proposed datasets.

Fig. 16: Sample omages of our home object dataset composed of four object with two test images for each one.

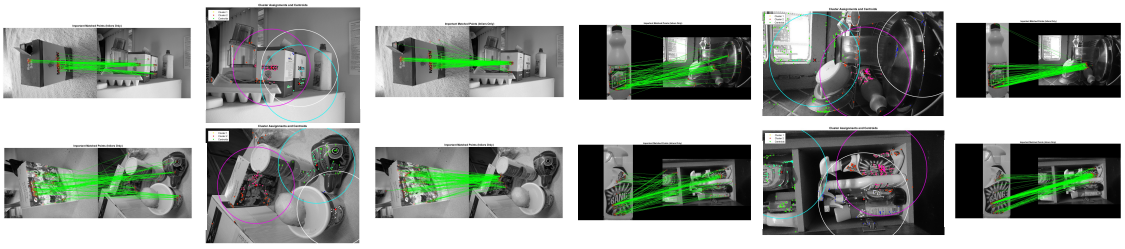

Fig. 17: Visual illustration of the pre-elimination phase showing sime examples of object detection before and after the pre-elimination phase.

5. Conclusions

In this paper, we proposed two contributions in the field of feature description and matching. The main property of our descriptor is its rotation invariance without the need to an orientation compensation step. Moreover, we added an outliers pre-elimination step, based on k-means clustering under the context of object detection, in order to enhance our descriptor detection accuracy. The ease of implementation of our descriptor and its resistance to different image changes makes it very attractive for wide use applications such as stereo Multiview or surveillance cameras. Experiments show a clear in-

crease of precision with the pre-elimination step in the context of object detection.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in Proceedings of the IEEE International Conference on Computer Vision, 2009.

[2] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in Proceedings - IEEE International Conference on Robotics and Automation, 2005.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., 2004.

[4] M. Güzel, "A Hybrid Feature Extractor using Fast Hessian Detector and SIFT," Technologies, 2015.

[5] Y. K. Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004., 2004.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in Proceedings of the IEEE International Conference on Computer Vision, 2011.

[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in Proceedings of the IEEE International Conference on Computer Vision, 2011.

[10] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012.

[11] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," IEEE Trans. Pattern Anal. Mach. Intell., 2010.

[12] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in Proceedings of the IEEE International Conference on Computer Vision, 2011.

[13] P. F. Alcantarilla, L. M. Bergasa, and A. J. Davison, "Gauge-SURF descriptors," Image Vis. Comput., 2013.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., 2012.

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in Proceedings of the British Machine Vision Conference 2015, 2015.

[16] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[18] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[19] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary Online Learned Descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 8, pp. 1–1, 2017.

[20] K. He, Y. Lu, and S. Sclaroff, "Local Descriptors Optimized for Average Precision," 2018.

[21] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-Aware Local Binary Feature Learning for Face Recognition," IEEE Trans. Pattern Anal. Mach. Intell., 2017.

[22] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.

[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," 2017.

[24] L. M. J. Florack, B. M. Ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "General intensity transformations and differential invariants," J. Math. Imaging Vis., 1994.

[25] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, "Moment invariants for recognition under changing viewpoint and illumination," Computer Vision and Image Understanding. 2004.

[26] A. Baumberg and S. Gu, "Reliable Feature Matching Across Widely Separated Views 1 . 1 . Current approaches to stereo correspondence 1 . 2 . Current approaches to wide baseline stereo," Current, 2000.

[27] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?,"' Comput. Vis. - ECCV 2002, 2002.

[28] G. Carneiro and A. D. Jepson, "Multi-scale phase-based local features," in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., 2003.

[29] T. Yamasaki, "Histogram of Oriented Gradients (HOG)," J. Inst. Image Inf. Telev. Eng., 2010.

[30] J. Donahue et al., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," 2013.

[31] M. S. Extremal, J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from," Br. Mach. Vis. Conf., 2002.

[32] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," ... Sci. Dep. Univ. Toronto, Tech. ..., 2009.

[33] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.

[34] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, 1998.

[36] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT," pp. 1–10, 2014.

[37] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in Proceedings of the IEEE International Conference on Computer Vision, 2015.

[38] C. B. Choy, S. Savarese, and M. Chandraker, "Universal Correspondence Network," pp. 1–17.

[39] V. Balntas, "Learning local feature descriptors with triplets and shallow convolutional neural networks," Bmvc, 2016.

[40] V. K. B. G, G. Carneiro, and I. Reid, "Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimizing Global Loss Functions," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[41] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[42] X. Zhang, F. X. Yu, S. Kumar, and S. F. Chang, "Learning Spread-Out Local Feature Descriptors," in Proceedings of the IEEE International Conference on Computer Vision, 2017.

[43] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in Proceedings of the IEEE International Conference on Computer Vision, 2005.

[44] W. T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," IEEE Trans. Pattern Anal. Mach. Intell., 1991.

[45] K. Mikolajczyk, C. Schmid, K. Mikolajczyk, C. Schmid, I. Computer, and C. Schmid, "Indexing based on scale invariant interest points To cite this version?:," 2010.

[46] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptor with multi-quantization," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[47] J.-M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," SIAM J. Imaging Sci., 2009.

[48] G. Yu and J. Morel, "A fully affine invariant image comparison method," Int. Conf. Acoust. Speech Signal Process. 2009. ICASSP 2009. . . . , 2009.

[49] C. C. Aggarwal, Data Mining: The Textbook. 2015.

[50] Q. Yu, Y. Luo, C. Chen, and X. Ding, "Outlier-eliminated k-means clustering algorithm based on differential privacy preservation," Appl. Intell., 2016.

[51] F. Jiang, G. Liu, J. Du, and Y. Sui, "Initialization of K-modes clustering using outlier detection techniques," Inf. Sci. (Ny)., 2016.

[52] G. Gan and M. K. P. Ng, "k-means clustering with outlier removal," Pattern Recognit. Lett., 2017.

[53] P. E. Forssén and D. G. Lowe, "Shape descriptors for maximally stable extremal regions," in Proceedings of the IEEE International Conference on Computer Vision, 2007.

[54] K. Mikolajczyk, K. Mikolajczyk, C. Schmid, and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Anal. Mach. Intell., 2005.

[55] M. Salzmann and F. Moreno-Noguer, "Closed-Form Solution to Non-Rigid 3D Surface," Eccv, 2008.

[56] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," 2008 IEEE Conf. Comput. Vis. Pattern Recognit., 2008.

[57] J. Heinly, E. Dunn, and J. M. Frahm, "Comparative evaluation of binary features," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012.

[58] K. Mikolajczyk et al., "A comparison of affine region detectors," Int. J. Comput. Vis., 2005.

[59] A. Haja, B. Jähne, and S. Abraham, "Localization accuracy of region detectors," in 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2008.

[60] M. H. Lee and I. K. Park, "Performance evaluation of local descriptors for maximally stable extremal regions," J. Vis. Commun. Image Represent., vol. 47, pp. 62–72, 2017.

[61] Aparna, K., Nair, M.K., 2016. Computational Intelligence in Data Mining. Springer. volume 2. chapter Effect of Outlier Detection on Clustering Accuracy and Computation Time of CHB K-Means Algorithm. pp. 25–35.

[62] 53 Objects dataset: Zurich Buildings Database. http://www.vision.ee.ethz.ch/en/datasets., 2003.

[63] Pierre Moreels. Home Object dataset. http://www.vision.caltech.edu/pmoreels/Datasets/Home_Objects_06/., 2006.

About Authors

Manel BENAISSA received her engineer degree in 2010 in signal processing from the University of Constantine 1, Algeria. In 2012, she received her master degree in communication systems, from the University of Skikda. She obtained the Magister degree in 2014 in image and video processing, from university of Constantine 1. She is now a Ph.D. student in the field of computer vision, in the signal processing lab of the electronics department in the University of Constantine 1.

Abdelhak BENNIA received his D.E.S. degree in 1983 in physics from the University of Constantine, Algeria. From 1984 to 1990, he attended the graduate school in electrical engineering at Virginia Tech. He received the M.Sc. degree in 1986 and the Ph.D. degree in 1990. Since 1990, he has been with the electronics department of the University of Constantine. His current research interests are neural network, character recognition, control systems, signal and image processing.