A Blur-Invariant Interest Point Detector Based on Moment Symmetry for Motion and Gaussian Blurred Image Matching

QIANG TONG[a]
[a]Graduate School of Information Sciences
Tohoku University
Sendai, JAPAN
tong@riec.tohoku.ac.jp

TERUMASA AOKI[a,b]
[b]New Industry Creation Hatchery Center
Tohoku University
Sendai, JAPAN
aoki@riec.tohoku.ac.jp

Abstract: Interest point detector is crucial to local feature-based image matching. However, lacking of robustness to strong blur is the fatal flaw of existing interest point detectors. As far as the authors know, all of the existing image matching methods fail to match a blurred image (caused by camera motion and out of focus, etc.) and a non-blurred image, even though blurred image matching is a critical task for many image/video applications. This article presents a blur-invariant interest point detector for blurred image matching. The proposed detector applies some blur-invariant image moments to detect a kind of special interest points from images. The special interest points are based on a new concept called Moment Symmetry (MS). These interest points are very robust to blur unlike traditional interest points based on corners or blobs. Experimental results show the proposed detector outperforms the state of the art interest point detectors for blurred image matching.

Key–Words: blur-invariant, local feature, moments, symmetry, interest point detector, image matching

1 Introduction

In the past decades, image matching methods based on local features have been successfully applied for many applications such as object recognition [1], wide baseline matching [2], image retrieval [3, 4], etc. Local feature-based image matching methods usually show much better performances than global feature-based methods for partial changes of images such as rotation, resizing and occlusion. However, as far as the authors know, there is no method which can be applied for blurred image matching in spite of the fact that blurred image matching is a critical technique for lots of applications. For example, blurred image matching is a very important and fundamental function to achieve traffic signs detection and object recognition for self-driving cars. When a self-driving car runs fast, the camera on the car will take motion blurred and Gaussian blurred videos or pictures. Hence, blurred image matching is an essential work which needs to be done for developing self-driving cars. Other typical examples are moving object tracking, image retrieval for moving cameras, face recognition, etc.

local feature-based image matching method usually includes three main steps. First, interest point detector finds hundreds or thousands of interest points from an input image and a reference image respectively. Then, local feature descriptor describes the surrounding information of each interest point and generates a feature vector. The final step is matching. By comparing these feature vectors from the input image and the reference image, it can be judged whether these two images are wholly or partially the same or not.

Unfortunately, lacking of robustness to strong blur is the fatal flaw of all existing interest point detectors. That is the biggest reason why the existing image matching methods cannot work for blurred image matching. For example, SIFT [5], which is well known as one of the best image matching methods is not good at blurred (especially motion blurred) image matching because its detector computes interest points from intensities of pixels which are very sensitive to blur. Other existing methods such as SURF [6], ORB [7], BRISK [8] are also sensitive (= not robust) to blur for the same reason. As a result, none of them are suited to achieving blurred image matching.

To solve this problem, a straightforward way is to introduce some deblurring algorithms [9–11] in a preprocessing step. Namely, an input image (blurred image) is deblurred by these algorithms first. Then, the deblurred input image and reference (non-blurred) images are matched by using existing image matching methods. However, this approach cannot work well in practice because all deblurring algorithms have at least one of the following two drawbacks. The first drawback is that all of the deblurring algorithms require huge computation cost since these algorithms always need an iterative process to correctly estimate

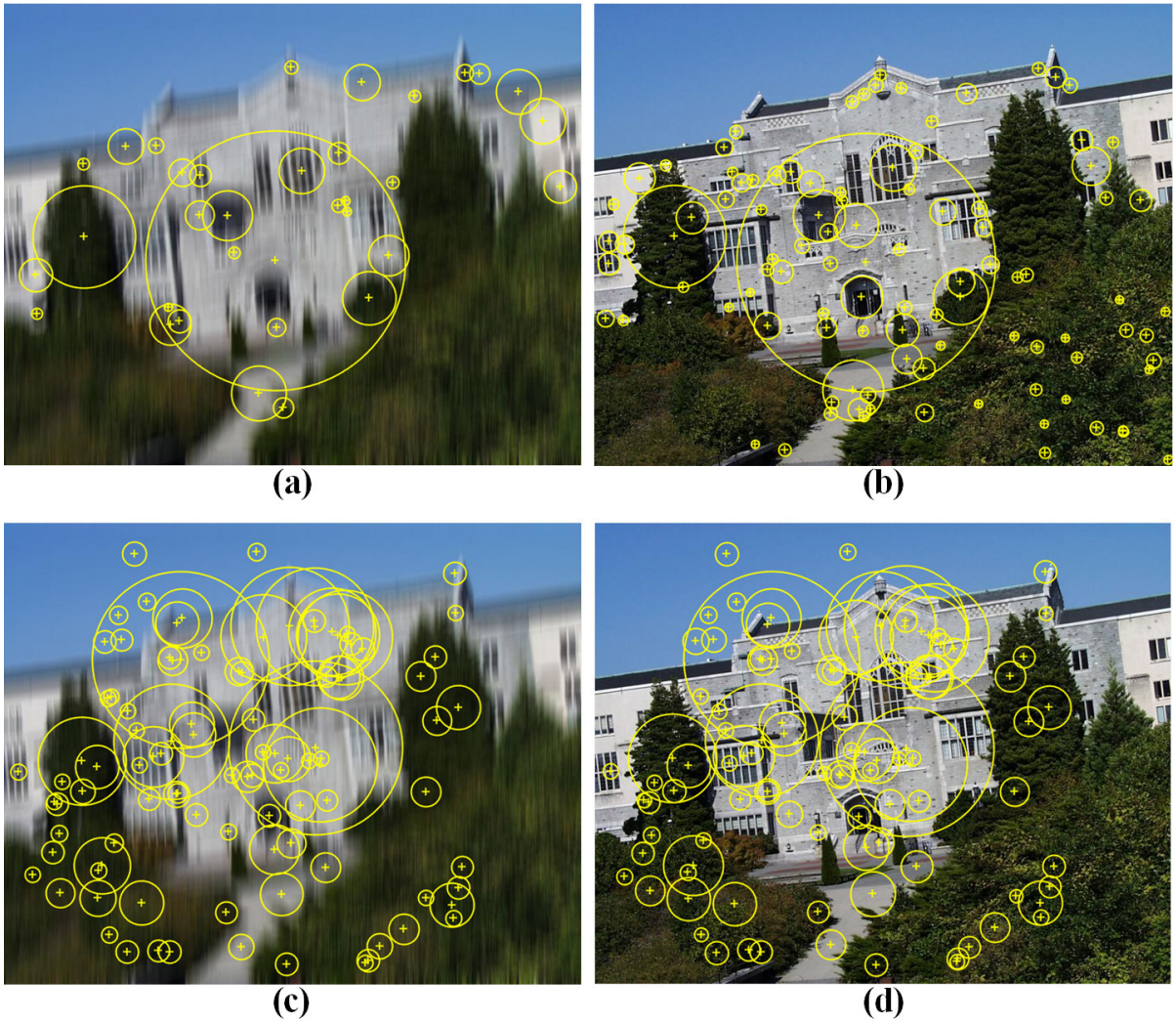

Figure 1: Result of SIFT detector for a motion blurred image and a non-blurred image are shown in (a) and (b) respectively, we can see that SIFT detector failed to detect corresponding interest points (yellow +) and their scales (sizes of the yellow circles) from these two images (less than 30 corresponding points are detected), result of our detector for these two images are shown in (c) and (d) respectively, we can see that our detector can correctly detect more than 300 corresponding interest points (only 100 are shown) from these two images.

some parameters of blur. The second drawback is that these algorithms may generate new artifacts such as blocks or ring bells. That usually may degrade the performance of image matching. Hence, we believe the best way to accomplish blurred image matching is to design a blur-invariant interest point detector and a blur-invariant descriptor.

In this paper, we present a blur-invariant interest point detector for blurred image matching. As shown in Fig. 1 (c) and (d), the proposed detector can detect lots of the same interest points from a blurred image and its non-blurred image unlike the state of the art detector ((a) and (b) in Fig. 1). The proposed detector is based on a new concept called "Moment Symmetry (MS)". MS is similar to reflection symmetry but MS is defined by looser conditions than reflection symmetry. We found MS regions (the regions extracted by the definition of MS) in images are very robust to strong blur unlike the traditional interest points such as corners or blobs. Then, we propose our detector based on some blur-invariant image moments to find these MS regions from images. In conclusion, our detector can correctly detect the same interest points from blurred or non-blurred images.

In this paper, we first describe the related work in section 2. Then, we present MS and the proposed detector in section 3, 4 respectively and give the experimental results in section 5. Finally, we conclude this paper in section 6.

2 Related Works

Until now, a lot of interest point detectors have been developed in practice. One of the most famous detectors is Harris corner detector [12] which detects corner points from an image by using the second moment matrix. Then, a scale-invariant corner detector called Harris-Laplace detector has been proposed in [13]. It combines Harris corner detector and a scale selecting approach - Laplacian of Gaussian (LoG) [14]. After that, many other scale-invariant detectors have been proposed. One of the most popular scale-invariant detectors is SIFT [5]. It approximates the Laplacian of Gaussian by using Difference of Gaussian (DoG) filters to detect corners and blobs from images. SIFT detector has been widely accepted as one of the best detectors. Fast Hessian detector of SURF [6] is another well-known detector. It maintains good performances while keeping a low computation cost. Other famous detectors are MSER [15], salient regions detector [16], etc.

However, most of the existing detectors extract corners or blobs from images by computing intensities or gradients of pixels. When an image is blurred, image structures like corners, blobs, and edges are likely to disappear, so that information like gradients, intensities drastically change. That is why all of the existing detectors mentioned above are sensitive to blur.

3 Moment Symmetry

In this section, we will give the definition of a new symmetry called "Moment Symmetry (MS)" for the following discussion. Because we found that the regions extracted by this definition (we call these regions "MS regions" from now on) are very robust to blur unlike the traditional interest points.

For each pixel of an image, we first choose a local region whose size is $n \times n$ (n is odd) surrounding the pixel. Then, for each local region, we define its "Moment Symmetry" as follows and check whether this region can satisfy the conditions of MS or not.

Definition 1 *We separate a local region into two portions by a symmetrical axis with the same number of pixels. Then we check all of the following three conditions.*

(I) The sum of intensities of each portion is the same for each other.

(II) The distance from the centroid of each portion to the symmetrical axis is the same for each other.

(III) The sum of skewnesses of two portions equals zero.

Fig.2 shows the simple examples of MS regions. Fig.2 (a) represents a reflection symmetrical region in an image. (b) is a motion blurred image of (a). (c) is a Gaussian blurred image of (a). Although (b) is not reflection symmetrical, the regions (a), (b), and (c) satisfy the above 3 conditions of MS, so all of them can be considered as MS regions. In other words, the definition of MS loosen that of reflection symmetry. In fact, MS regions includes reflection symmetrical regions and motion-blurred and Gaussian-blurred reflection symmetrical regions. As a conclusion, when we extract MS regions from images, we can find the same regions from non-blurred images and (Gaussian and motion) blurred images. And if we define the center of each MS region as an interest point, we can detect blur-invariant interest points from images.

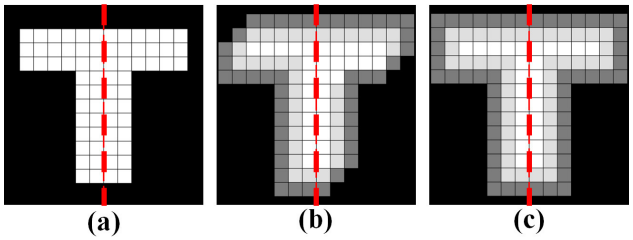

Figure 2: Some examples of MS regions, red lines in (a)∼(c) are the symmetrical axes, grids mean pixels, colors represent values of intensities.

Since all the above 3 conditions can be represented by some blur-invariant moments [17, 18] (0-order, 1^{st}-order geometric moments and 3^{rd}-order centroid moments), we can directly apply these blur-invariant moments to check whether a local region satisfies the MS conditions or not.

First, we define image moment for each portion as follows:

$$M_{pq} = \sum_{x=1}^{(n-1)/2} \sum_{y=1}^{n} x^p y^q f(x,y) \tag{1}$$

and

$$u_{pq} = \sum_{x=1}^{(n-1)/2} \sum_{y=1}^{n} (x-x_c)^p (y-y_c)^q f(x,y) \tag{2}$$

with

$$(x_c, y_c) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}) \tag{3}$$

where p+q is the order of moment, M_{pq} and u_{pq} are geometric moment and centroid moment respectively,

and x_c and y_c are the coordinates of the centroid, y-axis is always parallel to the symmetrical axis.

Then, we can use M_{00} to check the sum of intensities in condition (I) and use x_c, y_c to check the position of each centroid in (II). Also, we use u_{30} to check the skewness of each portion in condition (III). Finally, we extract MS regions and interest points from an image by checking these conditions.

4 Blur-Invariant Interest Point Detector Based on Moment Symmetry

As described in section 3, we can detect blur-invariant interest points from images by checking the MS conditions. So, in this section, we will present our blur-invariant interest point detector based on MS.

4.1 Moment symmetry in spatial domain

According to Def.1, before we check the MS conditions, we need to find a symmetrical axis for each region and divide the region into two portions. So, we present an efficient method as follows to do this work.

First we apply only a few filters to calculate some blur-invariant moments from the whole region instead of two portions of each local region. Then we find a symmetrical axis for the local region and check whether the region is a MS region or not according to these moments.

At first, we define image moment for the whole region as follows:

$$L_{pq} = \sum_{x=1}^{n}\sum_{y=1}^{n} x^p y^q f(x,y) \tag{4}$$

and

$$v_{st} = \begin{cases} \sum\limits_{x=(1-n)/2}^{(n-1)/2} x^s \sum\limits_{y=1}^{n} y^t f(x,y) \\ \qquad\qquad \text{if} \quad s+t>0 \\ \sum\limits_{x=(1-n)/2}^{(n-1)/2} sign(x) x^s \sum\limits_{y=1}^{n} y^t f(x,y) \\ \qquad\qquad \text{if} \quad s=0, t=0 \end{cases} \tag{5}$$

where L_{pq} and v_{st} are geometric moment and our centroid moment of the whole region respectively.

To easily compute the moments of (4) and (5) for each pixel in scale space described in section 4.2 and spatial domain of an image, we apply 5 re-sizable filters to convolute the image. Fig.3 shows some examples of the filters. Note that all these filters are circular filters in practice.

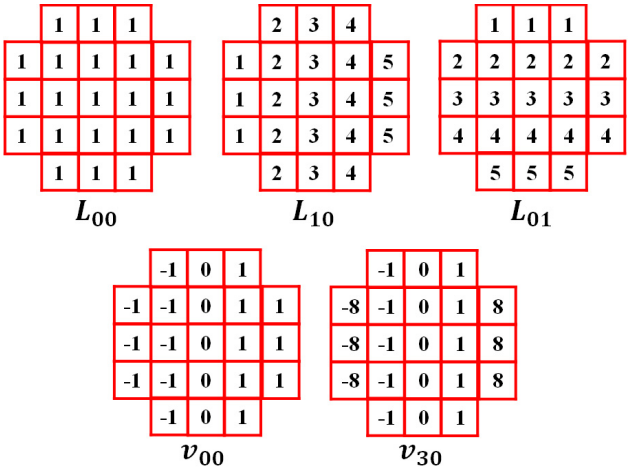

Figure 3: Some examples of the filters of moments, the sizes n of all filters are 5, the value in each grid represents weight $x^p y^q$.

We first use filters L_{00}, L_{10} and L_{01} with a certain size to convolute a given image to get 3 moments of the whole local region (its size equals each filter's size) surrounding each pixel. Then we calculate angle θ of the symmetrical axis for each local region as follows:

$$\theta = \arctan \frac{-(y_c - n/2)}{(x_c - n/2)} \tag{6}$$

with

$$x_c = \frac{L_{10}}{L_{00}},\ y_c = \frac{L_{01}}{L_{00}} \tag{7}$$

As described in [19], angle θ of the symmetrical axis can be easily obtained from (6) and (7). And according to the conclusions in [17,18] and the results of our preliminary experiments, centroids x_c and y_c of a local region are very robust to strong motion blur and Gaussian blur. That means we can correctly find the symmetrical axis even for a strongly blurred region according to (6) and (7).

After extracting the axis, we define new y-direction and x-direction which are parallel and perpendicular to the symmetrical axis respectively, and rotate filters v_{00} and v_{30} following the new direction. We convolute the image by using rotated v_{00}, v_{30} filters and get the values (responses) of each pixel. We call these values ***MS scores***. We then rewrite the MS conditions in ***definition 2*** and check these conditions for each pixel of the image. If the pixel satisfies all the conditions according to its MS scores, we can consider its surrounding region is a MS region and this pixel is an interest point candidate. Note that we also check the relationship of MS scores between the pixel

and its neighbor pixels in these conditions of Def.2 to make sure the detected interest points are well localized.

Definition 2

*(I) MS score of the pixel convoluted by filter v_{00} is less than a threshold **Th 1** and the scores of its 8 neighbor pixels.*

*(II) (x'_c-$\frac{n}{2}$) is less than a threshold **Th 2** and the values of its 8 neighbor pixels, where x'_c is a new x-coordinate of the centroid.*

*(III) MS score of the pixel convoluted by filter v_{30} is less than a threshold **Th 3** and the scores of its 8 neighbor pixels.*

Note that we can set 3 thresholds (***Th 1***, ***Th 2*** and ***Th 3***) for conditions $(I) \sim (III)$ of Def.2 respectively, but we prefer to set 1 threshold ***Th*** for all of these conditions. According to our preliminary experiments, we set the threshold $Th = -5$ (the value is Logarithmic value).

In this way, we don't need to divide each region into two portions, we can only calculate their moments by applying some filters and directly detect interest points based on the convolutional results.

4.2 Optimal scale selection

In this section, we provide a strategy of the optimal scale selection for each interest point. First, we use the filters with different sizes described in Fig.3 to convolute a given image and find the candidates of interest points. Then, for each candidate, we select the largest size which can retain its moment symmetry as its optimal scale.

We found the total difference between MS score of the interest point and the scores of its neighbor pixels attains the maximum in spatial domain and scale space, when the size of each filter equals the largest size of the MS region. So, we can borrow some ideas from image pyramid introduced in SIFT detector [5] and scale selecting method introduced in [14] to efficiently find the optimal scale for each interest point.

First, we down-sampled the given image by scaling factor of $1/2$ to generate some octaves of the pyramid of the image, until the size (either width or length) of the image is lower than a predefined value (for example 2^5 pixels). Namely in each octave, both length and width of the image is 2 times bigger than the image in the following octave. We also subdivide each octave into N_l layers (where N_l is a constant number). In each layer, we use the filters (described in 4.1) with the sizes S_f to convolute the image. We define the

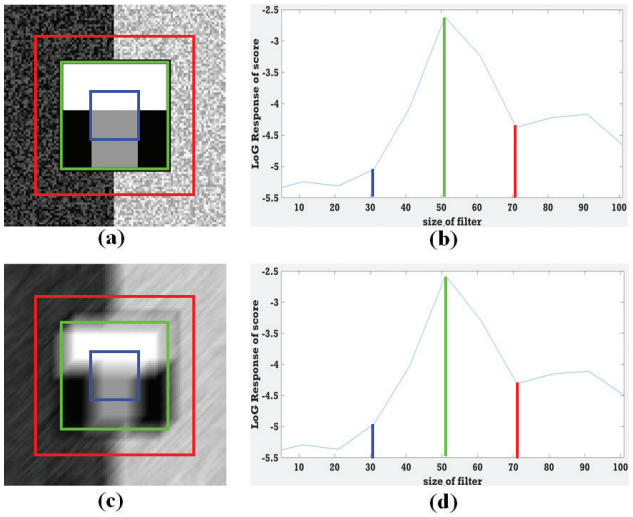

Figure 4: An example of response D of interest points (the center of the (a), (c)), the response D of the scores of the interest points in a scale space are shown in (b), (d), the scores are obtained by the filters with sizes from $5 \times 5 \sim 101 \times 101$ (horizontal axes in (b), (d)), the optimal sizes of moment symmetrical regions are 51×51, when the sizes of (green) filters in (a), (c) equal this size, their (green) responses (vertical axes) in (b) and (d) attain maximum.

size $S_f = 2^{\frac{2}{N_l}}\ l_0\ k$, where l_0 is the initial length of the filter in each octave, k is the order of this layer in its octave. In our prototype system, we set $N_l = 4$, $l_0 = 11$, that means in each octave the size of the initial filter is 11×11, and the size of each filter used in the following layer is 15×15, then 21×21, 29×29.

For each layer of each octave, we then use 5 filters to obtain MS scores of each pixel and detect the candidates of the interest point in spatial domain as described in section 4.1. Note that, we define average MS score $R = \frac{R_{00}+R_c+R_{30}}{3}$ instead of 3 raw scores R_{00}, R_c and R_{30} which are MS scores in 3 conditions of Def.2 respectively.

We use Laplacian-of-Gaussian (LoG) kernel to compute the LoG response D for the MS score R of each candidate and its $3 \times 3 \times 3$ neighbor pixels in the spatial domain and scale space as follows.

$$D(x, y, S) = L(x, y, \sigma) * R(x, y, S) \tag{8}$$

where $D(x, y, S)$ is the LoG response of MS score $R(x, y, S)$, $L(x, y, \sigma)$ represents LoG function, $\sigma = 0.5$, x, y and S represent the location and the scale of the candidate respectively.

The LoG kernel represents the difference between the center part and its surrounding neighbors of the filter, which helps to easily find the local maximum

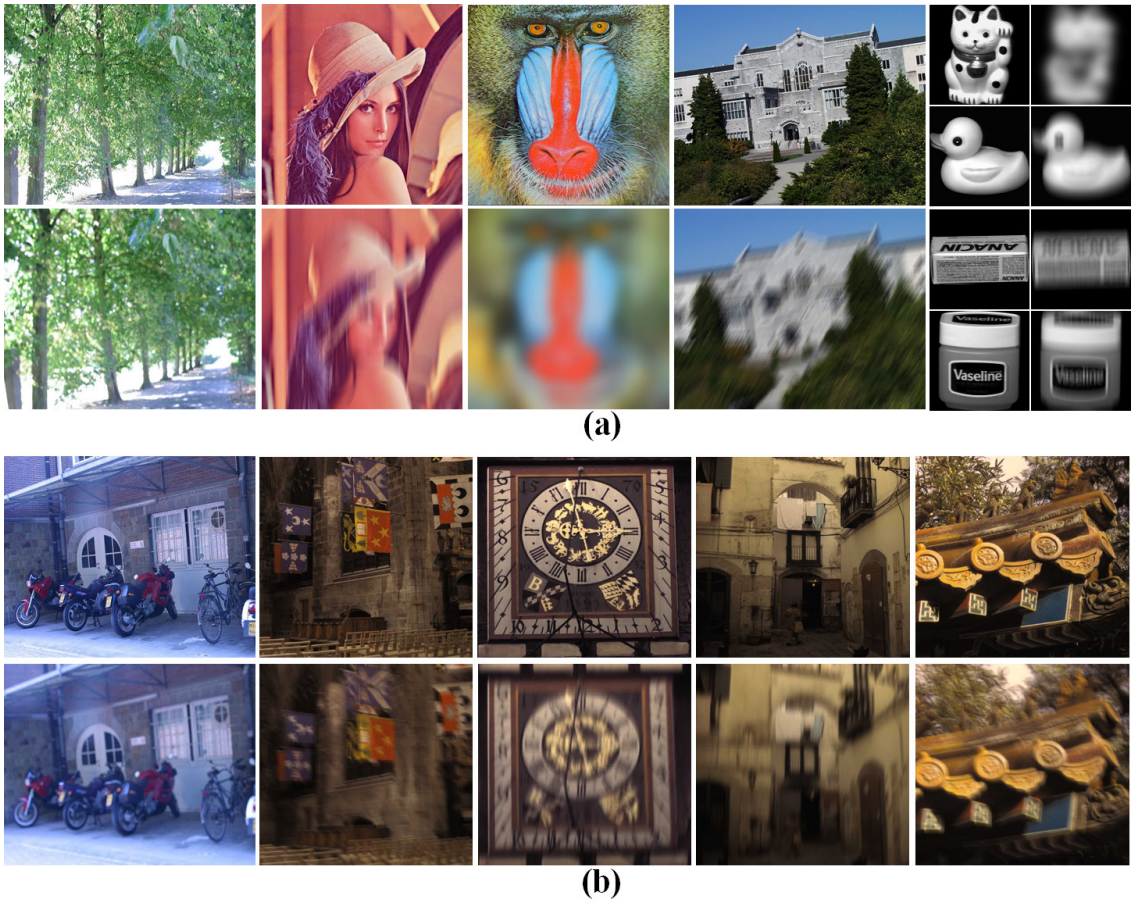

Figure 5: Some examples of our datasets: some images of Data-S and "Trees" are shown in (a), some images of "Bikes" and "Benchmark" are shown in (b).

of MS score of the point as shown in Fig.4. We can see that the difference D between the MS score R of the interest point and the scores of its neighbor pixels attains the maximum when the size of the filter equals the largest size of MS region.

Finally, we use a non-maximum suppression in the $3 \times 3 \times 3$ neighborhood of each candidate in scale space as described in [5] to easily find the location where the response D is a local maximum thereby find the optimal scale for this candidate.

4.3 Interest points localization and noises elimination

After all candidates of interest points with their scales are extracted in section 4.2, we can use a 3D quadratic fitting described in [20] to determine the interpolated location of the candidate and reject the candidates with poor score and poor location. We also check the redundancies of interest points. Namely, the point with the smaller scale will be discarded if there are more than one interest point detected at the same location. Since we assume the small MS region may be an image noise whereas the large MS region must be meaningful, stable and robust.

5 Experimental Results

5.1 Datasets

In our experiments, we used two different datasets: Dataset-S and Dataset-N. Dataset-S contains 100 non-blurred images including Lena, Baboon, and images from Columbia database [21], and their 3200 synthetically blurred images. All synthetic images are generated by motion blur kernels (8 directions: $0°$, $30°$, $45°$, $60°$, $90°$, $120°$, $135°$, $150°$) and Gaussian blur kernels, the sizes of which are from 1/12 to 1/6 of the image sizes. Also 200 blurred images are resized by 1/2 to 2 of sizes of non-blurred images. On the other hand, Dataset-N includes "Bikes" and "Trees" sets (each set includes 1 non-blurred image and 5 blurred images) from [22] and "Benchmark" set (4 non-blurred images and 48 blurred images) from [23]. All images in Dataset-N are real-world images extracted from natural videos. Some examples of Dataset-S and Dataset-

N are shown in Fig.5.

5.2 Evaluation for interest point detector

As the evaluation metric for interest point detectors, we used the number of corresponding points N_c and repeatability score R_s [13] defined as follows.

$$R_s = \frac{N_c}{min(N_b, N_{nb})} \tag{9}$$

where N_b and N_{nb} are the total number of interest points detected from a blurred image and a non-blurred image respectively. N_c is the number of the corresponding interest points between the blurred image and the non-blurred image. When two interest points from a blurred image and a non-blurred image are located at the same physical position, we call these two points "corresponding points".

Table 1: Examples of comparison of Dataset-S.

Detector	Blur size/ Image size	$Avg.N_c$	$Avg.R_s(\%)$
Our Detector	1/12	417	77
	1/9	352	71
	1/6	266	61
DoG	1/12	57	55
	1/9	41	46
	1/6	28	38
Fast-Hessian	1/12	55	61
	1/9	39	49
	1/6	32	46
Harris-Lap	1/12	27	44
	1/9	14	42
	1/6	2	50
MSER	1/12	17	70
	1/9	9	70
	1/6	2	49
Salient	1/12	31	64
	1/9	22	62
	1/6	6	67
FAST	1/12	0	0
	1/9	0	0
	1/6	0	0

We compare our detector to some existing methods such as DoG detector from SIFT, Fast-Hessian detector from SURF, Harris-Laplace, MSER, Salient regions detector [16], Fast corners detector from BRISK. Since both the number of detected interest points and the number of matched pairs are very important to image matching, we consider that a detector fails to find interest points from an image pair if N_c detected from this image pair is less than 50. Note that we know the homographic relationships between all non-blurred images and their blurred images in our datasets, so we can directly apply the function in ***OpenCV*** to calculate N_c and R_s for all detectors.

Table.1 shows the results of Dataset-S between our detector and some existing detectors. As shown in this table, we divide all blurred images into 3 classes according to the sizes of blur kernels regardless of Gaussian blur or motion blur in each class. And we just show the average results $Avg.$ in the third and fourth columns due to space limitations. We can see that detectors of SIFT (DoG) and SURF (Fast-Hessian) are better than other existing detectors but still cannot work well under the blur situation (failed to detect points from more than 80% images in this dataset, and the other existing detectors failed to detect points from more than 92% images). In contrast, our detector successfully detected the points from all images in this dataset. And average N_c and R_s of our detector are about 7∼9 times larger and 20∼25% higher than that of the best existing ones respectively.

Table 2: Examples of comparison of Dataset-N.

Detector	Sub-set	$Avg.N_c$	$Avg.R_s(\%)$
Our Detector	'B&T'	2205	73
	'Bench'	533	63
DoG	'B&T'	1054	51
	'Bench'	61	43
Fast-Hessian	'B&T'	704	53
	'Bench'	67	41
Harris-Lap	'B&T'	317	49
	'Bench'	21	39
MSER	'B&T'	109	48
	'Bench'	17	54
Salient	'B&T'	181	49
	'Bench'	22	46
FAST	'B&T'	96	40
	'Bench'	2	50

Table.2 shows the comparison results of Dataset-N. We used 3 subsets - "Bikes", "Trees" ($'B\&T'$) and "Benchmark" ($'Bench'$). And we also only show average results due to space limitations. Even though images of "Bikes", "Trees" are famous images and are

widely used in computer vision, they contain small blurs for our research. So, we just used 2 strongest blurred (the fifth and sixth) images in "Bikes" and "Trees". As shown in Table.2, we can see that our detector is better than the existing detectors. Average N_c and R_s of our detector are about 2∼8 times larger and 20∼30% higher than that of the best existing detectors respectively and without failed cases.

Table 3: Parameter Tuning of Our Detector.

N_l	l_0	$Avg.N_c$	$Avg.R_s(\%)$
	3	112	33
	7	136	39
2	11	176	44
	15	185	45
	3	240	60
	7	276	64
4	11	345	70
	15	367	72
	3	257	61
	7	290	62
6	11	371	71
	15	394	71
	3	354	52
	7	388	54
8	11	447	58
	15	464	59

Table.3 shows the parameter tuning of our detector. We adjusted the number of layer N_l and the initial length of the filter l_0 in the scale space described in section 4.2. Note that we only used Dataset-S to check the average N_c and R_s of our detector. As shown in Table.3, we can see that $N_l = 4$, $l_0 = 11$ as described in section 4.2 seem to be appropriate parameters. Although the results become a little bit better when we set larger N_l or l_0, the computational cost becomes much higher.

We then show some results of the comparison between the existing detector and the proposed detector in here. According to our preliminary experiments, SIFT's detector seems better than other existing detectors for blurred images. Hence, we just compared our detector to SIFT's detector due to space limitations.

As shown in Fig.6, each white + and each white circle represent an interest point and its scale respectively. We can see that less than 10 correct interest points are detected from two (motion and Gaussian) blurred image pairs by SIFT's detector as shown in Fig.6(a), (c). On the other hand, our detector can detect more than 200 correct interest points (only 100 points are shown) from these image pairs as shown in Fig.6 (b), (d). Also, our detector is scale-invariant as shown in Fig.7 (b).

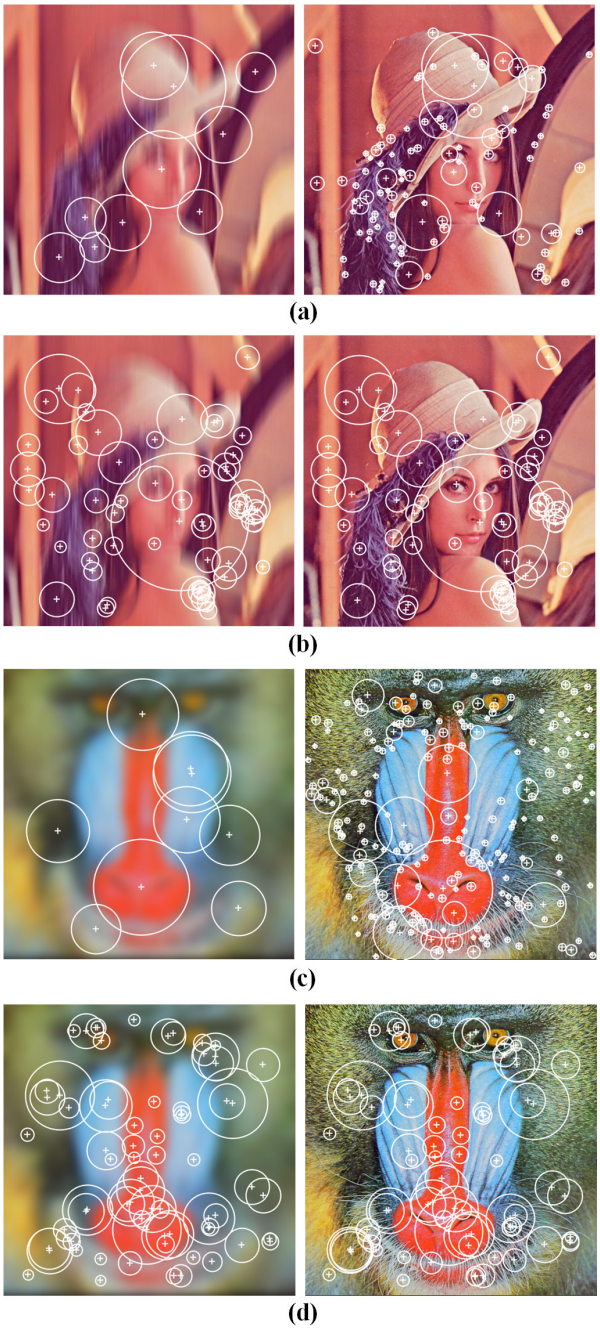

Figure 6: Some examples of interest point detection: the results of SIFT's detector are shown in (a), (c), the results of our detector are shown in (b), (d).

Finally, we show some results of blurred image matching in Fig.8. All these images are from Dataset-N. For each image pair including a blurred image and its non-blurred image, we use SIFT's detector and the proposed detector respectively to detect interest points

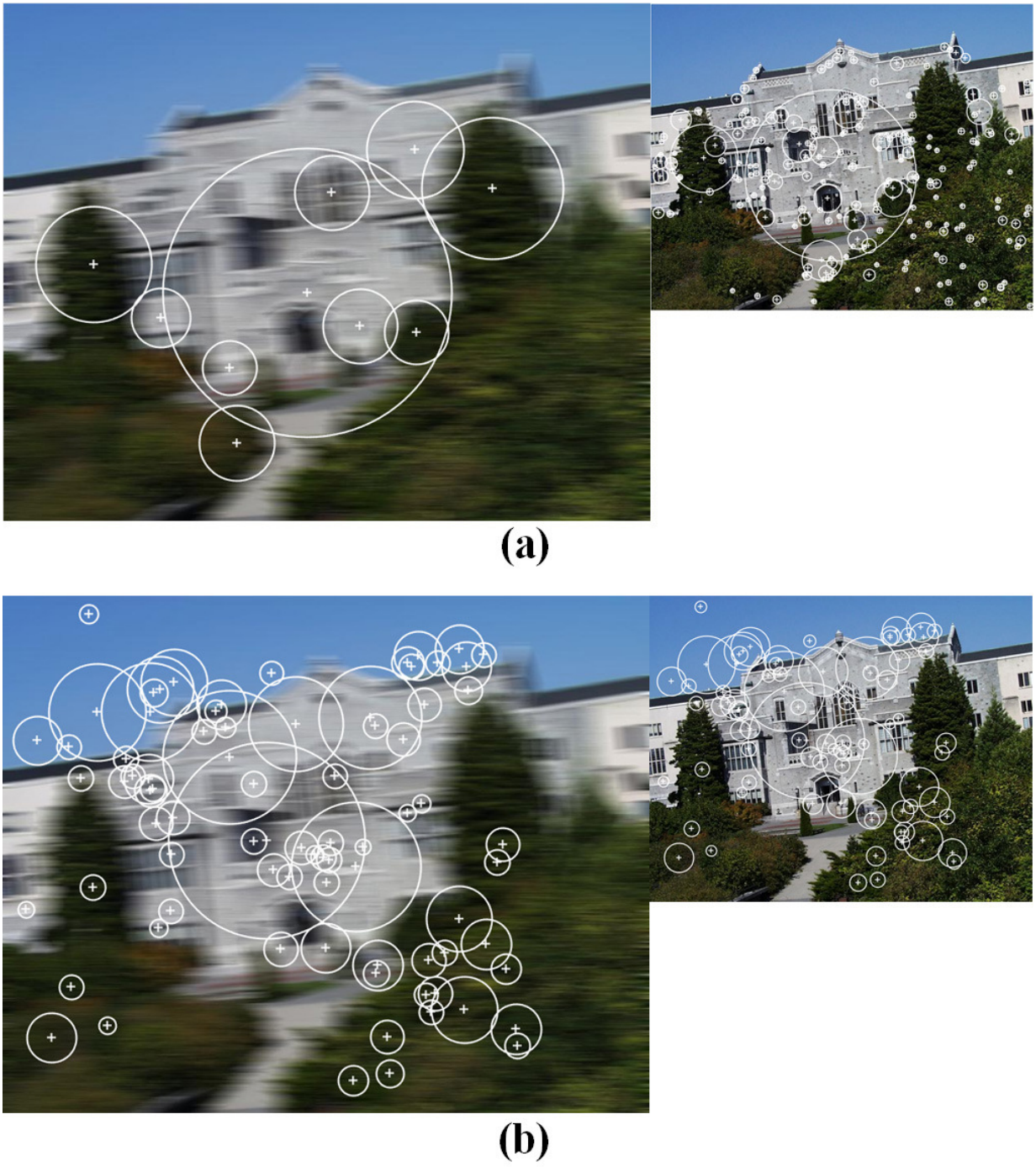

Figure 7: Some other examples of interest point detection for a zoom changing and blurred image pair: the results of SIFT's detector are shown in (a), the results of our detector are shown in (b).

from these images. Then, for each interest point from an image, the distances of all point pairs between this point and all interest points from another image are computed by using the blur-invariant descriptor [24]. If two points of a point pair are located at the nearest position in multidimensional vector space, this point pair is considered as a matched pair. After all matched pairs from an image pair are found out, we apply $RANSAC$ algorithm to eliminate outlier pairs, and we consider the retained inlier pairs as correct matches. Since the number of correct matches is very important to image matching, we consider that the two images in this pair cannot be matched, if there are less than 50 correct matches in this image pair.

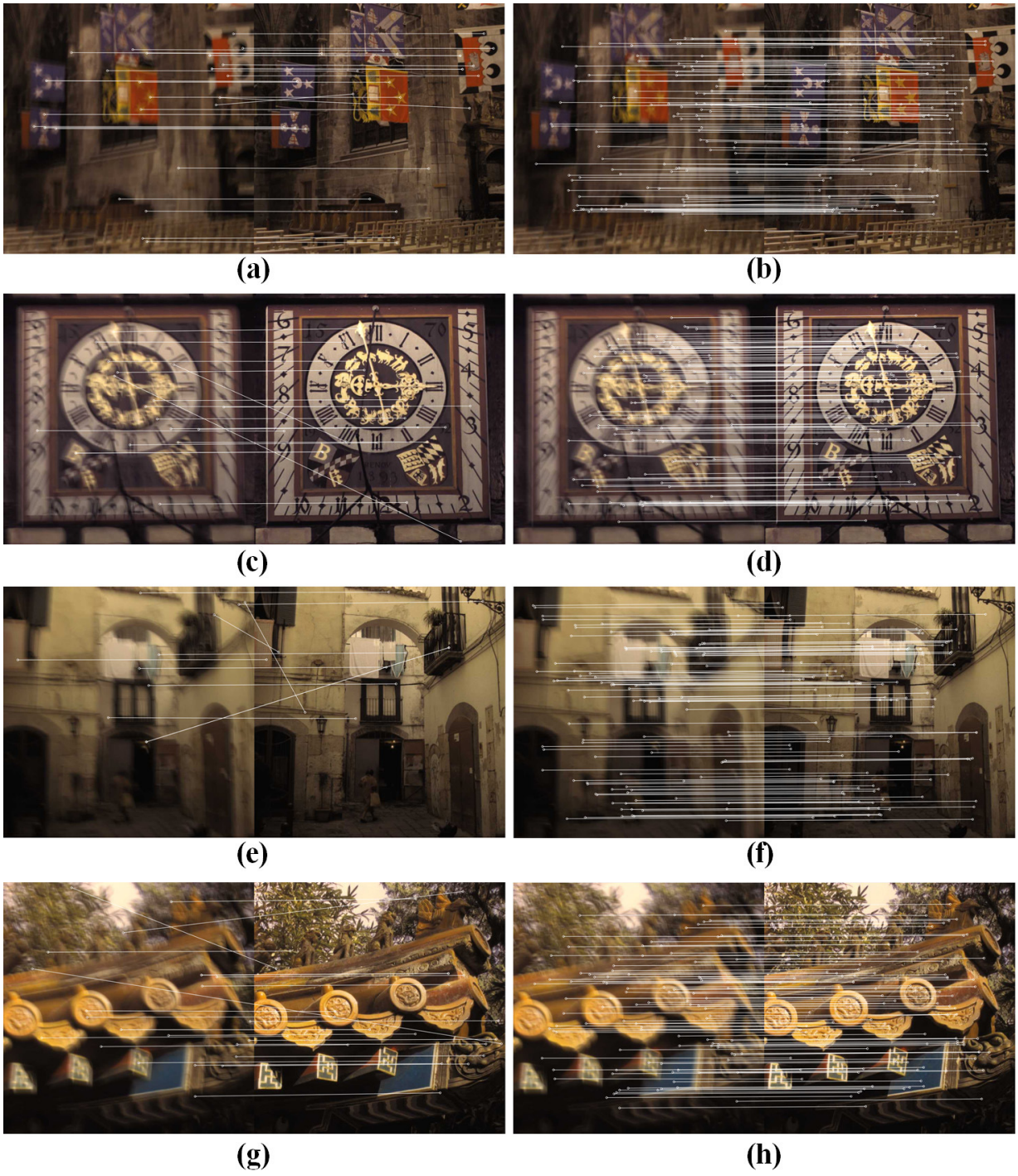

Figure 8: Some examples of image matching for Dataset-N: the results of SIFT's detector and the descriptor in [24] are shown in (a), (c), (e), (g), the results of our detector and the descriptor in [24] are shown in (b), (d), (f), (h).

As shown in Fig.8, each white line and each white circle represent valid matched point pair and interest point respectively. As shown in Fig.8(a), (c), (e),(g), we can see that 4 image pairs of Dataset-N cannot be matched by using SIFT's detector and the descriptor in [24], since there are less than 20 correct matches in each image pair. On the other hand, these 4 image pairs are successfully matched by using the proposed detector and the descriptor in [24] as shown in Fig.8 (b), (d), (f), (h). Note that, there are 200 correct matches in each image pair (b), (d), (f), (h), but we only show 100 correct matches. In conclusion, the proposed interest point detector can work well for blurred image matching.

6 Conclusion and Future Work

In this paper, we proposed a blur-invariant interest point detector based on "Moment Symmetry" and blur-invariant moments. By using our detector, we can easily detect the same interest points from a blurred image and a non-blurred image. The proposed detector outperforms the state of the art detectors and are good at Motion blurred and Gaussian blurred image matching. The proposed detector is also robust to scale changes.

However, there are still some limitations and problems which need to be fixed and solved. For example, the accuracy of localization of interest points may become worse when blur in an image becomes extremely strong. We plan to solve this problem in the near future. Also, we just focused on motion blurred and Gaussian blurred image matching in this paper. However, it is also one of our future work to make our detector suitable for other type blurred images .

References:

[1] V. Ferrari, T. Tuytelaars, L. VanGool, Simultaneous object recognition and segmentation by image, in: ECCV (2004) 40–54.

[2] T. Tuytelaars, L. VanGool, Matching widely separated views based on affine invariant regions, IJCV 59 (1) (2004) 61–85.

[3] K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest points, in: ICCV (2001) 525–531.

[4] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, PAMI 19 (5) (1997) 530–534.

[5] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.

[6] H. Bay, T. Tuytelaars, L. VanGool, Surf: Speeded up robust features, in: ECCV (2006) 404–417.

[7] E. Rublee, et al, Orb: An efficient alternative to sift or surf, in: ICCV (2011) 2564–2571.

[8] S. Leutenegger, Y. Siegwart, Brisk: Binary robust invariant scalable keypoints, in: ICCV (2011) 2548–2555.

[9] J. Cai, H. Ji, C. Liu, Z. Shen, Blind motion deblurring using multiple images, J. Comput. Physics 228 (14) (2009) 5057–5071.

[10] C. Schuler, M. Hirsch, S. Harmeling, Learning to deblur, PAMI 38 (7) (2016) 1439–1451.

[11] Z. Hu, M. Yang, Good regions to deblur, in: ECCV (2012) 59–72.

[12] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey vision conference (1988) 10–14.

[13] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, IJCV 60 (1) (2004) 63–68.

[14] T. Lindeberg, Feature detection with automatic scale selection, IJCV 30 (2) (1998) 79–116.

[15] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: BMVC (2002) 384–396.

[16] T. Kadir, M. Brady, Saliency, scale and image description, IJCV 45 (2) (2001) 83–105.

[17] J. Flusser, T. Suk, Degraded image analysis: an invariant approach, PAMI 20 (6) (1998) 590–603.

[18] J. Flusser, et al, Recognition of images degraded by linear motion blur without restoration, in: Theoretical Foundations of Computer Vision (1996) 37–51.

[19] D. Shen, et al, Symmetry detection by generalized complex (gc) moments: a close-form solution, PAMI 21 (5) (1999) 466–476.

[20] M. Brown, D. Lowe, Invariant features from interest point groups, in: BMVC (2002) 656–665.

[21] http://www1.cs.columbia.edu/CAVE/software/softlib/coil 20.php.

[22] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, PAMI 27 (10) (2004) 1615–1630.

[23] R. Khler, S. Harmeling, Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database, in: ECCV (2012) 27–40.

[24] Q. Tong, T. Aoki, A blur-invariant local feature for motion blurred image matching, in: International Conference on Digital Image Processing (2017) 181–187.