IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2020, accepted May 20, 2020, date of publication May 25, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996944

Non-iterative Covariant Feature Extraction Based on the Shapes of Local Support Regions

LUPING LU[1], **YONG ZHANG**[2], **(Associate Member, IEEE), AND KAI LIU**[3]

[1]Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China
[2]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[3]Farsee2 Technology Company Ltd., Wuhan 430079, China

Corresponding author: Yong Zhang (6553961@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 41701526, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042017kf0036.

ABSTRACT Feature extraction is important in image matching. However, the perspective deformations, especially the anisotropic scaling deformations will affect the performances of feature extraction algorithms. To improve the image matching results when notable perspective deformations exist, an algorithm for extracting feature points and covariant regions is introduced in this paper. We propose using a new type of feature points, the ''inside corner points'' as seed points. And we propose using a multi-scale seeded region growing method to find the local support regions for feature points. Based on the shapes of local support regions, an image patch around a feature point can be rectified by doing shape normalization, and the anisotropic scaling deformations can be reduced by the rectification. By doing image matching with these rectified image patches, the matching results are notably improved.

INDEX TERMS Feature extraction, covariant region, local support region, shape normalization, image matching.

I. INTRODUCTION

We know that image feature matching becomes harder when the perspective deformations of images increase. On the one hand, features are less likely to be extracted from the same position in different images because of the image deformations, as what is we call the repeatability of features. On the other hand, even if features can be extracted from the same position, the similarities between feature descriptors will be low if we don't take care of the image deformations. From simple to complex, the image perspective deformations can be classified into three groups: the rotation in the plane ('stage 1' deformation), the isotropic scaling ('stage 2' deformation) and the anisotropic scaling ('stage 3' deformation). The 'stage 0' indicates no deformation, as showed in Figure 1. Among them, the rotation in the plane has the smallest influence on feature extraction [1], and it's easy to solve by estimating a main direction in feature description. Isotropic scaling has a larger influence on feature extraction because the details of objects appear different at different scales. Currently, the isotropic scaling is handled by building image scale space. The anisotropic scaling has the largest influence on feature extraction compared to the former two, and circular feature description window is not suitable anymore for images with severe anisotropic scale deformations. A good feature extraction algorithm is expected that the corresponding features are detected at the same location, and each feature is accompanied by a local support region whose shape is covariant to whatever the deformations on the images.

Removing the anisotropic scaling deformation is the key to solve the image matching problem under large perspective deformations. And shape normalization methods play an important role in removing the anisotropic scaling deformation. Shape normalization algorithms have been proposed many years ago, and here we introduce two of them. The first is called Shape Compacting [2], [3] which uses Eigenvalue Decomposition of the shape covariance matrix to rectify the shapes. And the second is referred to as Cholesky Shape Normalization [4] which uses Cholesky Decomposition [5] of the shape covariance matrix to rectify the shapes. It's interesting to notice that both of them use the shape of the object to remove the anisotropic scaling deformation. We think the underlying principle is: the perspective deformations change the shape of an object, so according to the shape of the object, the perspective deformations can be discovered and removed. By employing shape normalization, two shapes of an object

The associate editor coordinating the review of this manuscript and approving it for publication was Hugo Proenca.

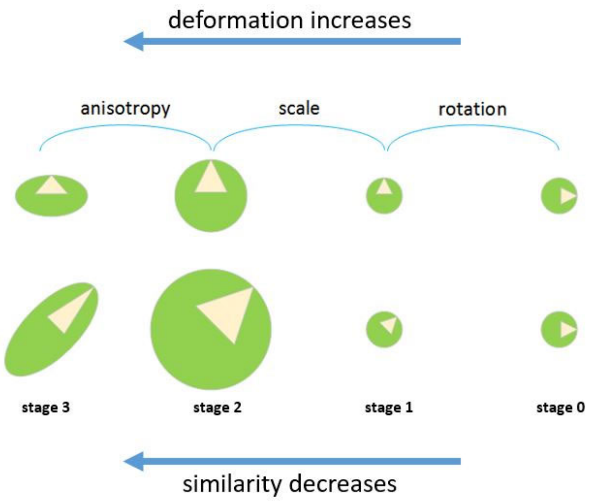


FIGURE 1. **The similarities between shapes decrease as the image deformations increase.**

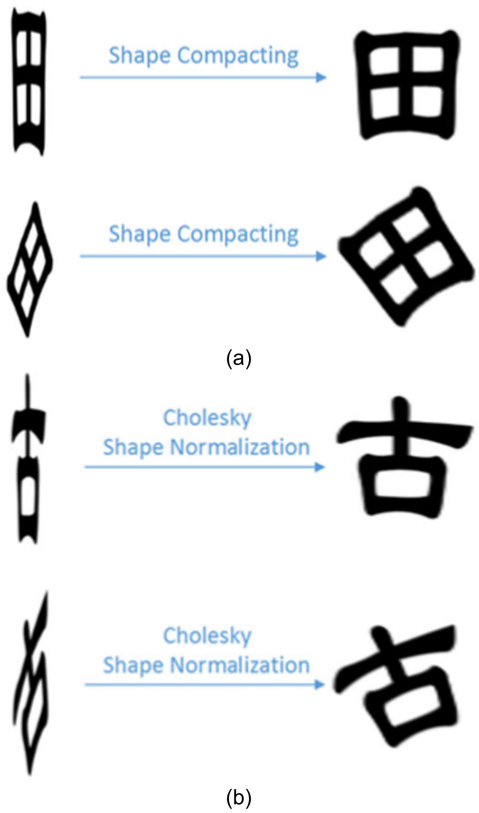


FIGURE 2. **Shape normalization results of the Shape Compacting method and the Cholesky Shape Normalization method.**

under 'stage 3' deformation can be transformed into shapes under 'stage 2' or even 'stage 1' deformation. Examples of applying Shape Compacting and Cholesky Shape Normalization are shown in Figure 2. In the examples, the black area forms the shapes of a feature. After applying shape normalization, the rectified image patches can be matched by existing algorithms (such as SIFT [6]) very easily.

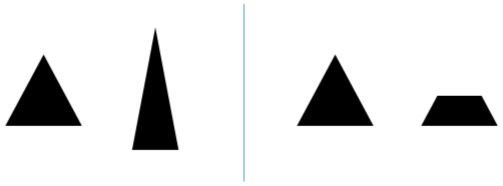


FIGURE 3. **Covariant regions and non-covariant regions.**

From above, we can see the shape normalization methods are very effective in removing the anisotropic scaling deformation. But we should also notice an important fact that the algorithms work well only if the corresponding shapes of a feature are perfectly detected from images. 'Perfectly detected' means the shapes of a feature (we call it the local support region of a feature in this paper) detected from different images contain exactly the same contents of the feature. For example, the two shapes on the left of Figure 3 are perfectly detected and they contain the same contents of the triangle despite under different deformations. In other words, they are covariant regions. But the two shapes on the right are not. If the local support regions of a feature detected from different images contain different contents of the feature, the results generated by shape normalization will still exist anisotropic scaling deformations.

So the problem of removing the anisotropic scaling deformation transforms into the problem of detecting features from images as well as detecting perfectly corresponding local support regions of the features. And the method we propose in this paper is aiming to solve this problem.

The contributions of this paper can be summarized as follows: 1) we proposed using a new type of points, the "inside corner points" as feature points. They are not corner points nor blob points, instead, they are points that locate a little inside of the corners. 2) We proposed using a multi-scale seeded region growing method to find the local support regions for feature points. The anisotropic scaling deformations can be reduced by doing shape normalization to these local support regions.

The paper is organized as follows: the related works are introduced in section II, the implementation of our algorithm is introduced in section III, the experiments and results are presented in section IV, and the conclusions are given in Section V.

II. RELATED WORKS

Feature extraction [7], feature description [8] and false match removal [9], [10] are three important contents of image matching. The topic of this paper belongs to the feature extraction and the other two topics will not be covered, although they are important.

The MSER algorithm [11] is the most related to our topic. We think it's a very inspiring algorithm. This algorithm extracts region features and the pixels belong to a region constitute the support region of the feature. Then it

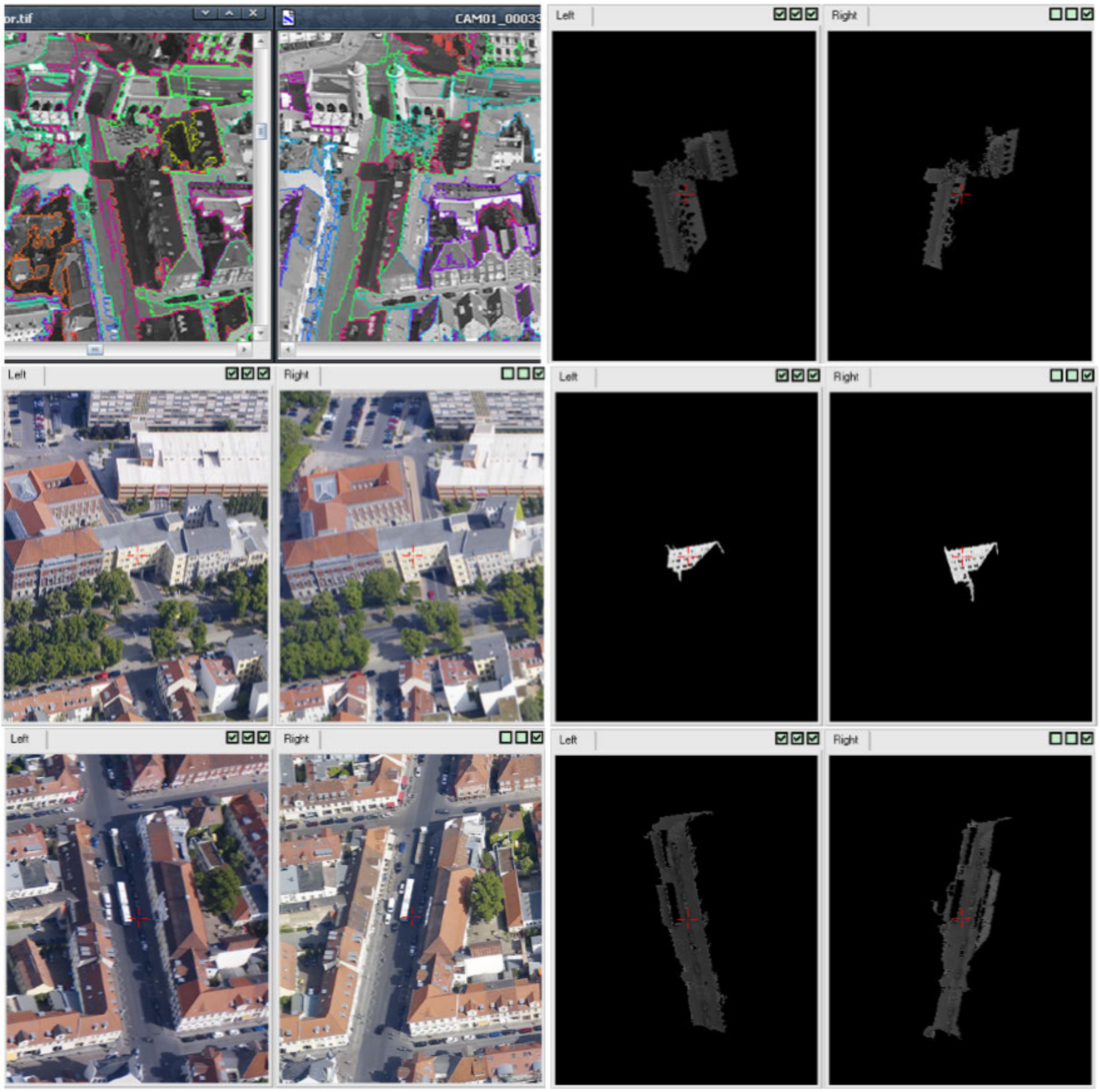


FIGURE 4. **Examples of regions detected by MSER algorithm. (In each row, two original image patches that contain a same position are given on the left, the MSER regions detected from them are presented on the right.)**

employs the shape normalization methods to rectify the image patches and these rectified image patches are used to do image matching. The idea is very good, but the algorithm has some disadvantages in practical use: (1) it has a very small number of features since the method detects maximally stable extremal regions only, and this method performs well only when the images contain many homogeneous regions with distinctive boundaries; (2) many MSER features are not local and some features cover a large area of the image. The non-local features are more likely to contain non-coplanar objects and detecting perfectly corresponding shapes for such features is harder because of image occlusions.

Due to the complexities of the scene, the regions detected from images by MSER algorithm do not very often cover the same image contents. Three examples of MSER regions are given in Figure 4. In each row of this figure, two original image patches that contain the same position are given in the first two columns, and the MSER regions detected from them are given on the last two columns respectively. The corresponding MSER regions cover different image contents more or less. From these examples, we can see it's not easy to detect perfectly corresponding shapes.

Tuytelaars and Van Gool also proposed two methods, the EBR and IBR [12] to extract covariant regions. The EBR algorithm starts from corners and uses the nearby edges to form a region. The IBR algorithm starts from intensity extrema to detect regions. The performance of EBR relies on the edge detection results, and the IBR has some similarities with MSER. We categorize MSER, EBR and IBR as covariant region detection methods based on the geometry (or shape) information of an image.

There are also some methods, such as Harris-affine and Hessian-affine [13], have been proposed to detect covariant regions. But they use gradients of the image intensities, not

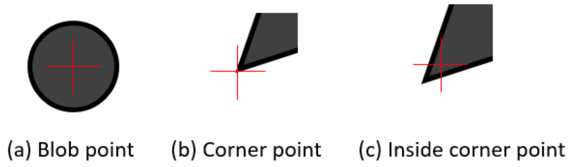

(a) Blob point (b) Corner point (c) Inside corner point

FIGURE 5. Illustration of three different types of feature points.

the geometry information of the images. So we categorize them as covariant region detection methods based on intensity gradients information. Unlike the methods based on shape information, these type of methods have to work in an iterative way. The readers can know more about those methods from paper [14]–[16]. We prefer methods based on shape information because we think the geometrical deformations change the shapes of an object, so according to the shapes of the object, the geometrical deformations can be discovered and removed, as we have mentioned above.

The methods listed above are very representative, although they are relatively old now. In recent years, a few methods were proposed. Paper [17] extracts a set of scale invariant blob points by means of the idea of blob evolution along with different scales by checking the value of the Gaussian curvature. And they achieved the affine shape adaptation by iteratively fitting an anisotropic Gaussian function to the blob features by means of a nonlinear least squares approach. Paper [18], [19] proposed an affine invariant similarity comparison between image patches from the point of view of Riemannian Manifolds. According to their derivation, the tensor product of the gradient vector can be used as the affine covariant structure tensors, which is the second moment matrix essentially.

In this paper, we propose an algorithm to detect covariant regions which we named AIFE (Affine Invariant Feature Extraction). Our purpose is to extract regions, but our algorithm begins with feature point detection. We propose a method to detect the ‘‘inside corner points’’. These type of points are not corner points nor blob points, they are points that locate a little inside of the corners, as shown in Figure 5. Then we use such a point as the seed point to do a local seeded region growing. The result of the region growing operation becomes the local support region of the seed point. Then we use the shape of the local support region to do a shape normalization, the parameters calculated from shape normalization are used to rectify the local image patches. And like what MSER algorithm did, these rectified image patches are used to do image matching. Further details about AIFE algorithm are introduced in the subsequent sections.

III. IMPLEMENTATION OF AIFE ALGORITHM

A. DETECT ‘‘INSIDE CORNER POINTS’’

First, we will introduce the method for detecting the ‘‘inside corner points’’. We want ‘‘inside corner points’’ because we will use them as seed points for region growing, and the image intensity of an inside corner point is more stable than a point located on the corner. Our method is modified on the basis of the Harris algorithm [20] and the steps of the Harris algorithm are listed below:

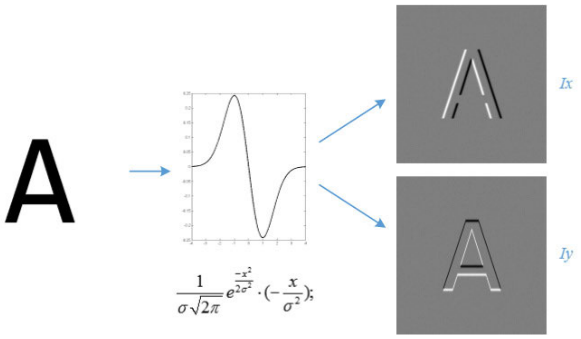

A
$\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-x^2}{2\sigma^2}}\cdot(-\frac{x}{\sigma^2});$
Ix
Iy

FIGURE 6. Illustration of calculating the gradients of image intensities with the first order Gaussian derivative function.

(1) Calculate the gradients of image intensities along the x direction and y direction by doing image convolutions with the first order Gaussian derivative function $D(\sigma)$. The convolution results are notated by Ix and Iy respectively, as showed in Figure 6. In the figure, the original image is given on the left, the first order Gaussian derivative function is given in the middle and the convolution results are given on the right.

(2) Calculate Ix^2, $IxIy$, Iy^2 by doing pixel-wise multiplication within Ix and Iy.

(3) Implement image convolutions to the three images $Ix^2, IxIy, Iy^2$ with a Gaussian function $G(\sigma)$. The convolution results constitute the three components of the Second Moment Matrix $\mathbf{M}$, as showed in Formula (1).

(4) Calculate the Harris response R for every pixel of the image according to Formula (1).

(5) Find local extrema of the Harris response. The amount of the extracted feature points can be controlled by setting a grid size in searching extrema.

$$\mathbf{M} = \begin{bmatrix} G(\sigma) * Ix^2 & G(\sigma) * IxIy \\ G(\sigma) * IxIy & G(\sigma) * Iy^2 \end{bmatrix}$$
$$R = \det(\mathbf{M}) - \alpha trace(\mathbf{M})^2; \tag{1}$$

In the Harris algorithm, two parameters are very important to the extraction results, one is the standard deviation $\sigma 1$ in the Gaussian derivative function $D(\sigma)$ and the other is the standard deviation $\sigma 2$ in the Gaussian function $G(\sigma)$. We find that by increasing the values of the two parameters, the positions of local extrema of the Harris response will move towards the inside of the corners, as shown in Figure 7. According to this observation, we can detect the ‘‘inside corner points’’.

In Figure 7, the original image is given on the left, which is one black square on white background. And images (a), (b), (c), (d) present the Harris responses when the values of the two parameters are both 1.0, 4.0, 8.0, 22.0, respectively. And take the bottom left corner of the square as an example, the positions of local extrema (indicated by a red cross) of the Harris response are shown in images (1), (2), (3), (4).

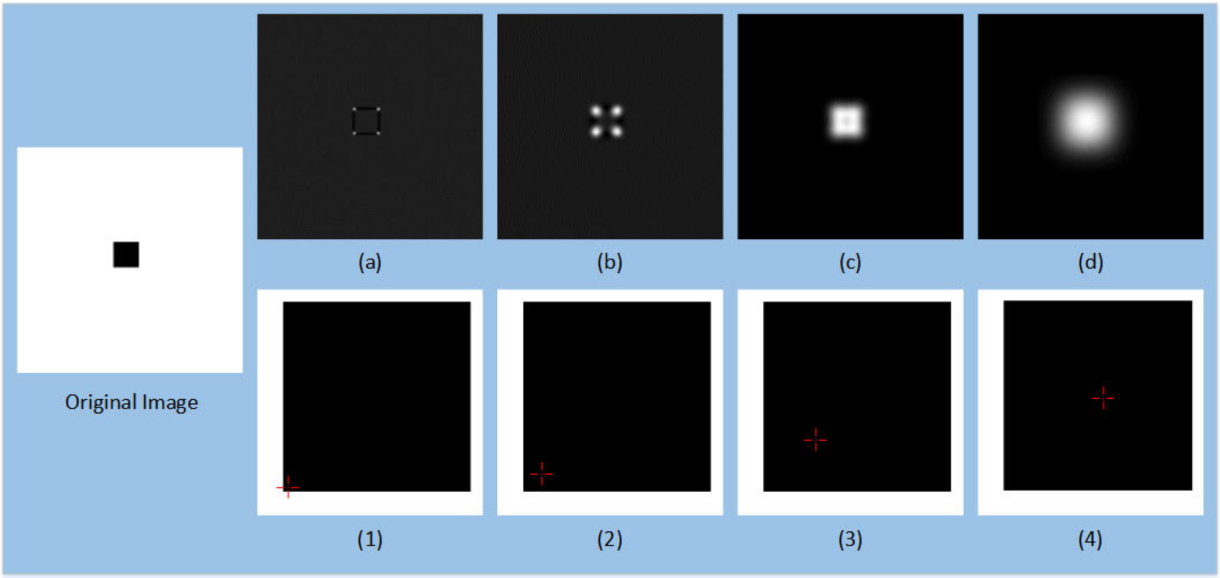


FIGURE 7. Impacts to the Harris responses and the positions of local extrema when increasing the values of the two standard deviations.

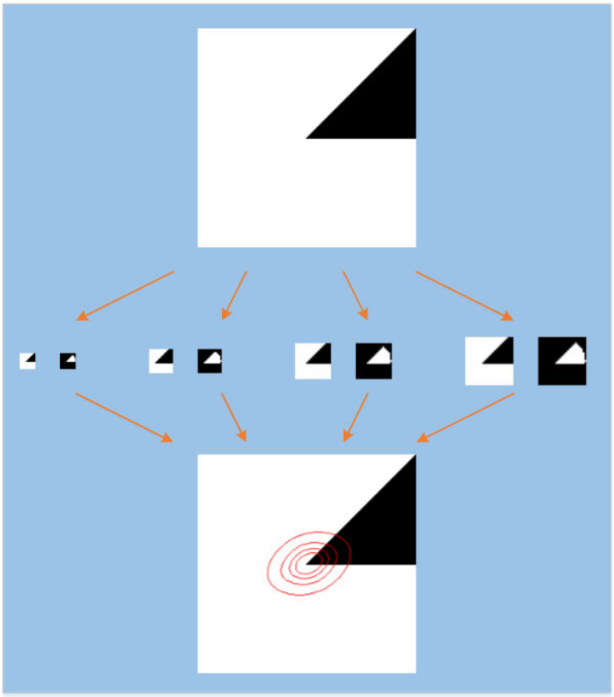


FIGURE 8. Using multi-scale seeded region growing to obtain local support regions for a feature point.

In order to see more clearly, the images (1), (2), (3), (4) are enlarged versions of the original bottom left corner. It can be seen that the positions of local extrema move towards the inside of the corner when increasing the parameters' values. So, if one wants the extracted feature points to stay exactly on the corners, smaller parameters are better; if one wants the extracted feature points to stay inside of the corners, larger parameters should be chosen. To the requirement of our algorithm, we want the feature points to be a little inside of the corners, so the value 2.0 is used in our algorithm.

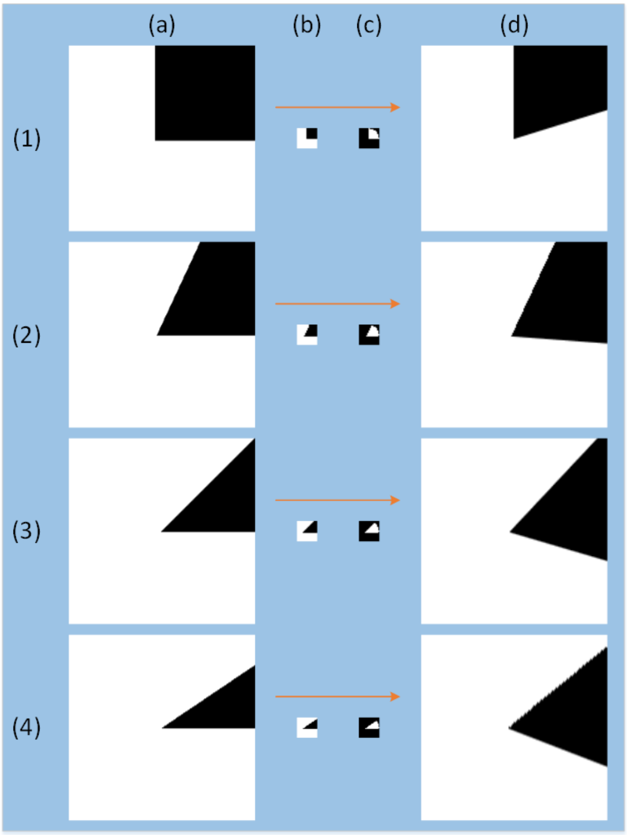
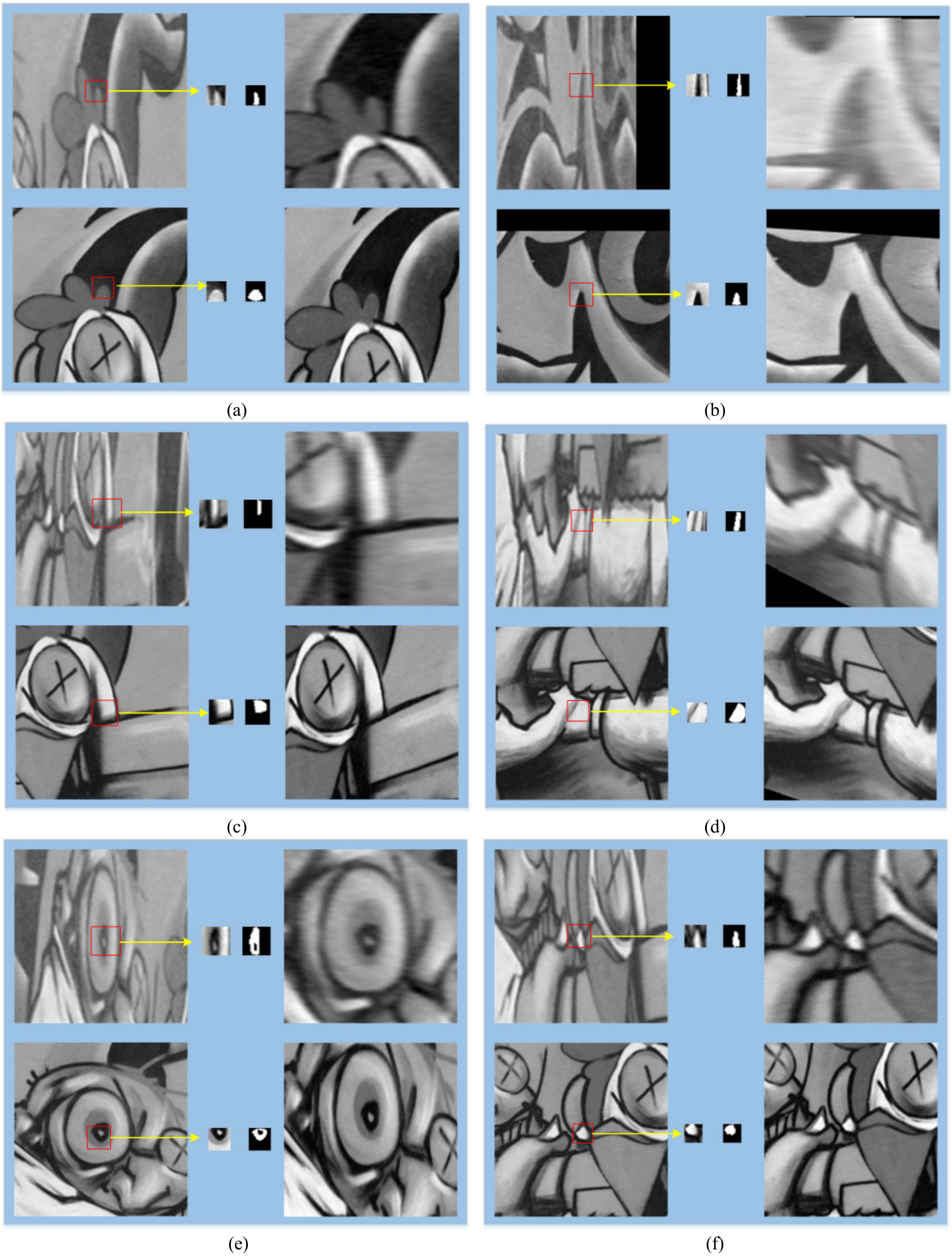


FIGURE 9. Examples of image rectification based on the shape of local support region.



(a) (b) (c) (d) (e) (f)

FIGURE 10. Examples of image rectification results on real images using AIFE algorithm.

B. OBTAIN LOCAL SUPPORT REGIONS

As we mentioned above, detecting perfectly corresponding shapes for features is the key for shape normalization. We archive this goal by doing multi-scale seeded region growing. For each ‘‘inside corner point’’, we use it as the seed point for region growing. And the region growing is limited within the circular area that with the seed point as the center and the radius of k pixels. We use 6 different region growing radiuses k (ranging from 15 pixels to 25 pixels) for each seed point to realize the multi-scale. So for each seed point, we will obtain 6 region growing results and each result forms a local support region for this feature point. The process is illustrated in Figure 8. In this figure, 4 different region growing radiuses are presented. The image in the top row presents the original

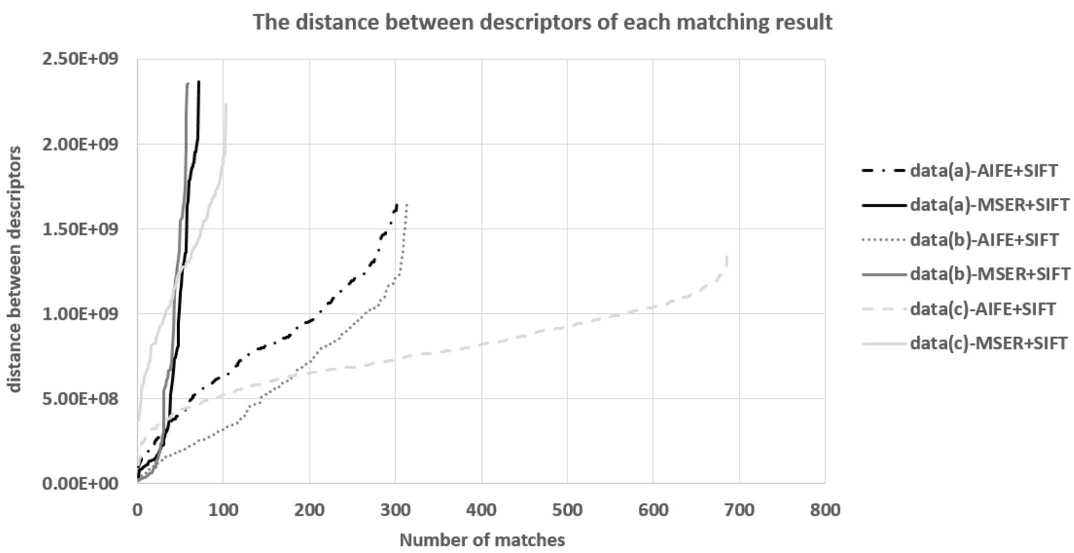


FIGURE 11. Distances between feature descriptors of corresponding points for the six matching results.

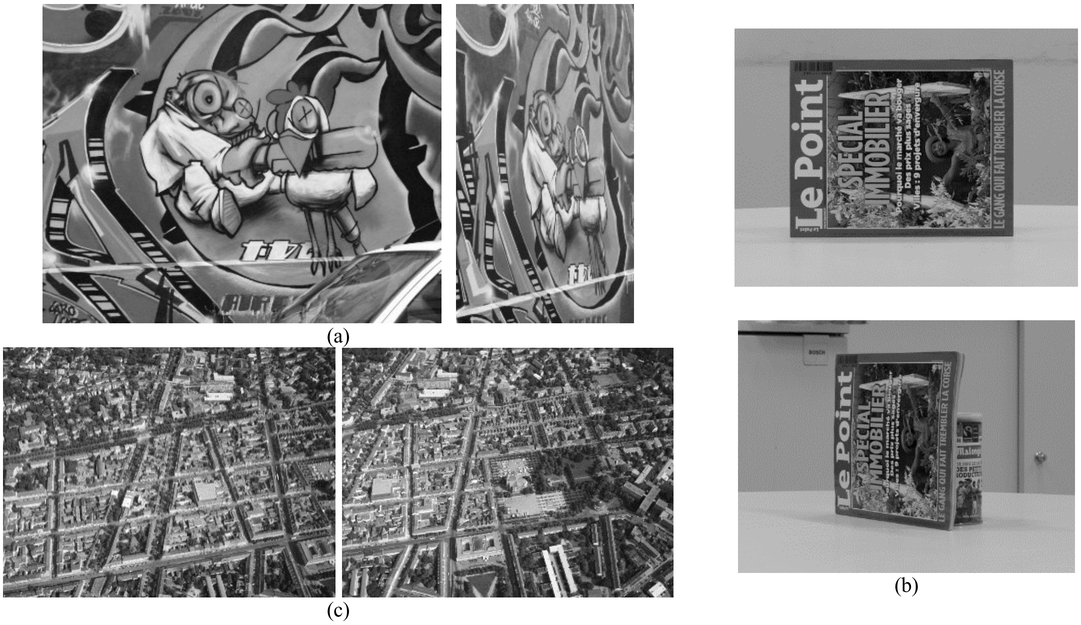
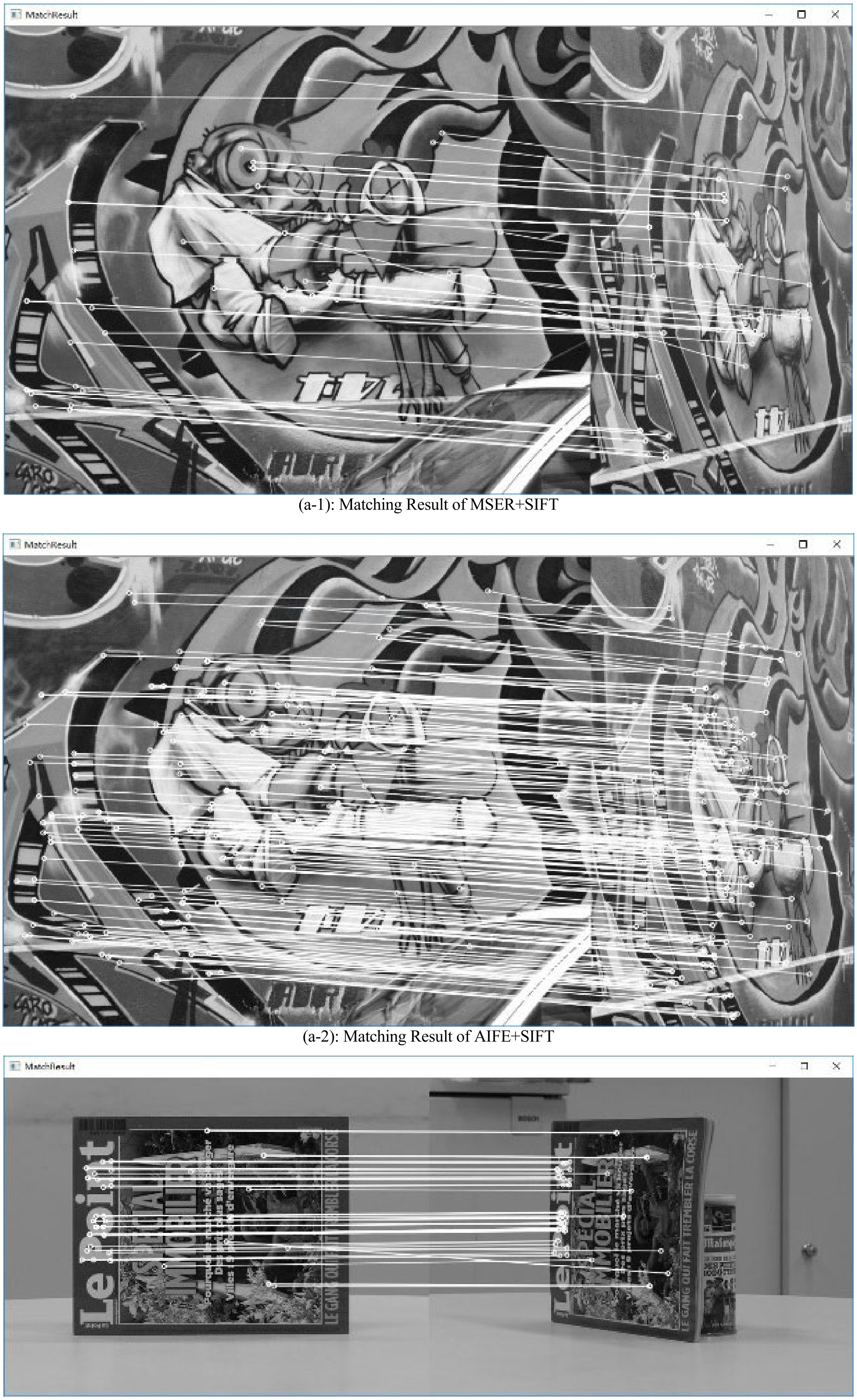


FIGURE 12. Test data used in the experiments. (The data (a) are images "graf-image1" and"graf-image5" fromhttp://www.robots.ox.ac.uk/~vgg/research/affine/. The data (b) are images "magazine_V1_front" and "magazine_V1_60deg" from http://www.cmap.polytechnique.fr/~yu/research/ASIFT/demo.html).

image, and the four pairs of small images in the second row show the region growing results. The image on the left in each pair shows the original image content within the radius and the image on the right shows the region growing result. Each shape obtained by region growing can be fitted by an ellipse, and the 4 ellipses are presented in the bottom image.

C. RECTIFY LOCAL IMAGE PATCHES

After we obtain local support regions for each feature, we can use the shape of these regions to rectify local image patches around each feature point. To do this, we should first calculate the shape covariance matrix $\mathbf{C}$ of the area covered by the region as shown in formula (2). In the formula, P indicates a local support region, and (x, y) indicates the pixel coordinates. If a pixel (x, y) belongs to the region P, then $f(x, y)$ equals 1, otherwise, it equals 0. When Matrix $\mathbf{C}$ is obtained, we can use its Cholesky Decomposition to rectify the image, as showed in formula (3). The matrix $\mathbf{L}$ is a lower triangular matrix, the vector $\mathbf{x}$ indicates the pixel coordinates in original image and the vector $\mathbf{x}'$ indicates the pixel coordinates in the

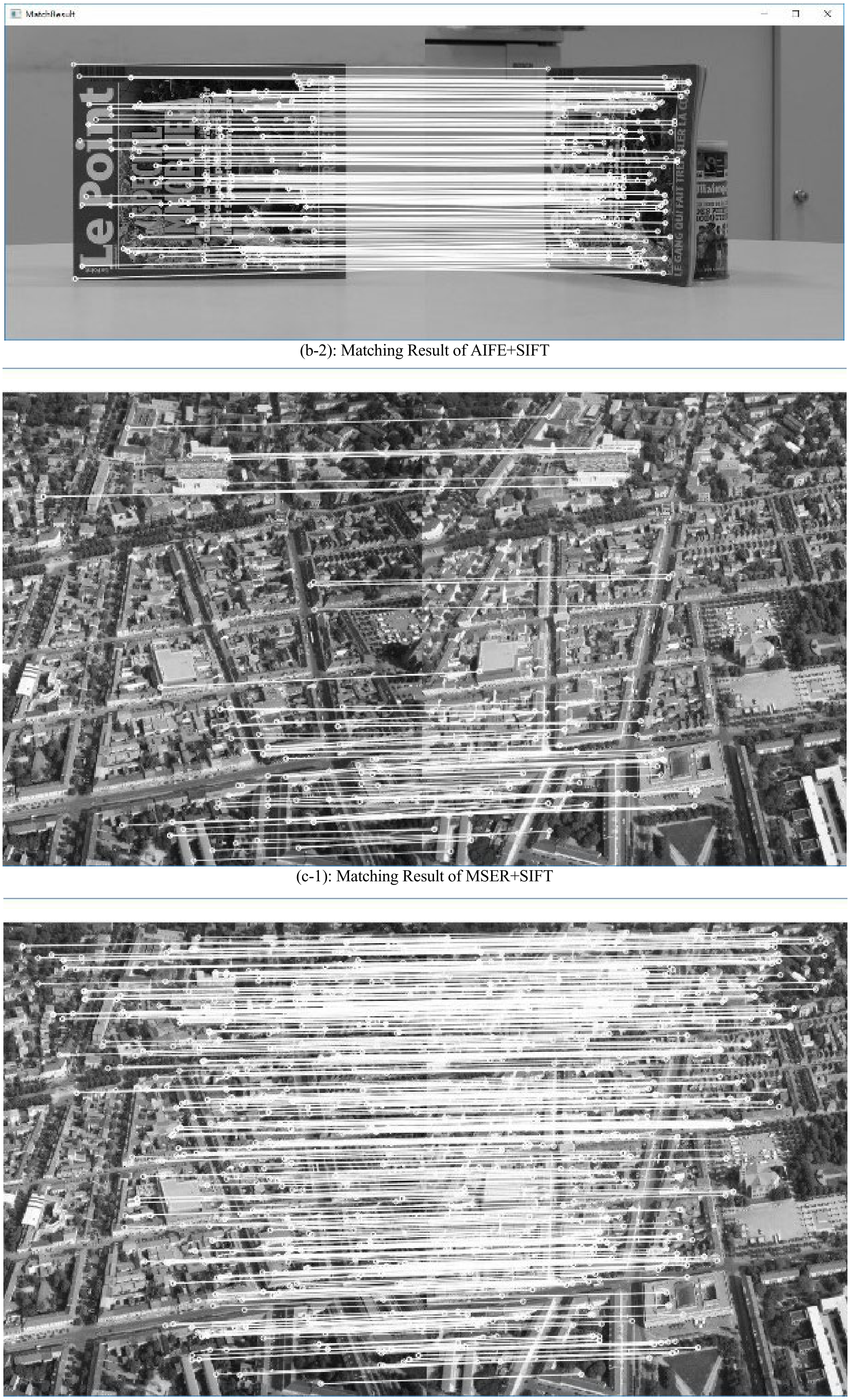


(a-1): Matching Result of MSER+SIFT

(a-2): Matching Result of AIFE+SIFT

(b-1): Matching Result of MSER+SIFT

FIGURE 13. **Comparison of matching results between MSER+SIFT and AIFE+SIFT.**



(b-2): Matching Result of AIFE+SIFT

(c-1): Matching Result of MSER+SIFT

(c-2): Matching Result of AIFE+SIFT

FIGURE 13. ***(Continued.)*** **Comparison of matching results between MSER+SIFT and AIFE+SIFT.**

rectified image. For each local support region, we use it to rectify a local image patch whose size is 100 × 100 pixels around the seed point.

$$
\begin{aligned}
f(x,y) &= \begin{cases} 1, (x,y) \in P \\ 0, (x,y) \notin P \end{cases} \\
m_{10} &= \frac{\sum_x \sum_y x^1 y^0 f(x,y)}{\sum_x \sum_y f(x,y)} \\
m_{01} &= \frac{\sum_x \sum_y x^0 y^1 f(x,y)}{\sum_x \sum_y f(x,y)} \\
m_{11} &= \frac{\sum_x \sum_y (x-m_{10})^1 (y-m_{01})^1 f(x,y)}{\sum_x \sum_y f(x,y)} \\
m_{20} &= \frac{\sum_x \sum_y (x-m_{10})^2 (y-m_{01})^0 f(x,y)}{\sum_x \sum_y f(x,y)} \\
m_{02} &= \frac{\sum_x \sum_y (x-m_{10})^0 (y-m_{01})^2 f(x,y)}{\sum_x \sum_y f(x,y)} \\
\mathbf{C} &= \begin{bmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix} \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{C} &= \mathbf{L}\mathbf{L}^T \\
\mathbf{x}' &= \mathbf{L}^{-1}\mathbf{x} \qquad (3)
\end{aligned}
$$

Examples of image rectification based on the shape of local support region are presented in Figure 9. In this figure, the images in column (a) present four synthetic images of a black desk corner seeing from four different view directions. According to the view direction, the angle of the corner varies. Images in Column (b) present local image contents around each "inside corner point" extracted from images in column (a). Images in Column (c) present the seeded region growing results within the circular mask, and these results become the local support regions of the feature points. In these cases, the local support regions are fan shapes. Images in Column (d) present the corresponding rectification results of each image in column (a). The rectification parameters are calculated based on the shape of the support regions in column (c). It can be seen that after the rectification, the four images of the desk corner are much similar to each other (the angles of the corner are much closer to each other).

More examples of image rectification results on real images are presented in the next section. All rectified image patches generated by all local support regions are used to do the image matching.

Till now, the main steps of the proposed AIFE algorithm have been introduced, and the whole procedure is outlined in Algorithm. 1. Let's recapitulate our considerations when we designed the algorithm: (1) we would like to choose point as the basic feature primitives in matching. Point is the simplest type of feature, so it has a wide adaptability to various scenes. This is important to ensure the quantity of features and their distributions in the images. (2) A feature point needs its support region. Our goal is to make corresponding support regions detected from different images contain exactly the same contents of the feature. Only then we can use their shapes to normalize the image patches. (3) The support region of a feature point needs to be local. On the one hand, it can decrease the influence of image occlusion, on the other hand, it's easier to find a small support region than a big one. (4) Image segmentation is a good way to obtain the support regions. And the seeded region growing is a handy way to do image segmentation. More importantly, it has two elements we want: a seed point and a region. We can control the size of the region by setting some scale parameters. To the seed point, we discover the strategy to extract the 'inside corner point', and it meets the requirement that a seed point should locate in a stable image intensity area.

Algorithm 1 The AIFE Algorithm

```
Input: Two images
Output: The image matching results
Workflow:
  1. Detect "inside corner points" for each input image;
  2. Do multi-scale seeded region growing for each feature point to obtain local support regions;
  3. Do image rectification for each local support region;
  4. Do feature extraction and description on each rectified image patches using the SIFT algorithm.
  5. Find correct matches between two images.
```

IV. EXPERIMENTS AND RESULTS

A. EXAMPLES OF IMAGE RECTIFICATION RESULTS ON REAL IMAGES

In this section, we will present more examples of image rectification results on real images, as shown in Figure 10. From these examples, the readers can have a more intuitive understanding of our algorithm. Six examples are presented, and in each of these examples, a pair of original image patches are given on the left. They contain the same part of an object but with different projective deformations. The image around an "inside corner point" and the corresponding seeded region growing results are given in the middle. The corresponding rectified image patches are given on the right.

It can be seen that the similarities between the rectified image patches are much greater than the similarities between the two original image patches. Then feature extraction and feature description can be done on the rectified image patches. The SIFT algorithm is used in our algorithm. Since the coordinates of the rectified image patches can be rigorously mapped to the coordinates of the original images, these new extracted feature points can be used to match the original images directly.

B. COMPARISON WITH MSER METHOD

We did many tests to evaluate the performance of our algorithm. Here three of them are presented. The performance of the algorithm is evaluated according to its image matching results. The workflow of the image matching is introduced

TABLE 1. Processing time and number of correct matches.

Test data	Image size(pixel)	Processing time(ms)		Number of matches	
		MSER+SIFT	AIFE+SIFT	MSER+SIFT	AIFE+SIFT
data (a)	800x640	2193	13296	71	304
data (b)	600x450	1204	8200	59	313
data (c)	803x603	2704	59515	103	687

first. To the images we are going to match, we first use AIFE algorithm to obtain many rectified image patches, then we use SIFT algorithm to do feature extraction and feature description on these rectified image patches. Then the feature points are matched by the brute force + RANSAC method. This process is denoted as AIFE+SIFT in the following. As a comparison, we also use MSER algorithm to obtain rectified image patches, and the other operations remain the same. The process is denoted as MSER+SIFT. The MSER code we used is from the VLFeat library [21]. Comparisons about the performance of different affine covariant feature extraction algorithms can be seen from [14], [15], [17]. From their experiments, we can see that regarding the viewpoint change, the repeatability and matching score of MSER is best in most cases. And from the design of the algorithm, our method is most related to the MSER algorithm. So in this paper, we choose to compare our method with the MSER method.

The number of correct matches is a good indication of the performance of the image matching algorithm. Besides, if the rectification can remove most of the anisotropic scaling deformation, the distance between feature descriptors of corresponding points will be small. This is a good indication of to what extent the image deformations have been removed. So we also compared the distance between feature descriptors of corresponding points. More correct matches and smaller distance between feature descriptors indicate better algorithm performances.

The three test data are presented in Figure 12. The data (a) and data (b) are obtained from the internet. They both contain planer objects. And data (c) are aerial images with non-planar objects. The program was executed on a laptop with an Intel Core i5 7500 CPU and 8G RAM. No GPU nor CPU acceleration techniques were used. The processing time and the number of correct matches are presented in Table 1. The processing time of our algorithm is longer than MSER method, but the number of correct matches is much more than MSER method. The Euclidian distance between feature descriptors of each corresponding point is presented in Figure 11. These distances are arranged in ascending order for each of the six matching results. So they form six curves as presented in Figure 11. The three distance curves of AIFE+SIFT are all under the distance curves of MSER+SIFT. This means the similarities of feature descriptors generated by our algorithm is better than the MSER method. In other words, it indicates the rectification results of our method are better than MSER method. The matching results are presented in Figure 13. It can be seen that the distributions of the matches of our algorithm are better, too.

V. CONCLUSIONS

An algorithm for extracting feature points and covariant regions is introduced in this paper. We proposed using a new type of feature points, the "inside corner points" as seed points. Point is the simplest type of feature, so it has a wide adaptability to various scenes. This is important to ensure the quantity of features and their distributions in the images. Although we use "inside corner points" in our implementation, the blob feature points can be used as well. And we proposed using a multi-scale seeded region growing method to find the local support regions for feature points. Our support regions are pretty local. On the one hand, it's easier to find a smaller support region than a big one, on the other hand, smaller support regions suffer less the influence of image occlusions. What's more, a local small region is enough to reveal the local deformations of the image. So far very promising matching results have been obtained by our algorithm. But the whole process is a little time consuming and acceleration strategies should be considered in the future.

REFERENCES

[1] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, 2000.

[2] J.-G. Leu, "Shape normalization through compacting," *Pattern Recognit. Lett.*, vol. 10, no. 4, pp. 243–250, Oct. 1989.

[3] S.-C. Pei and C.-N. Lin, "Image normalization for pattern recognition," *Image Vis. Comput.*, vol. 13, no. 10, pp. 711–723, Dec. 1995.

[4] T. Cohignac, C. Lopez, and J. M. Morel, "Integral and local affine invariant parameter and application to shape recognition," in *Proc. 12th Int. Conf. Pattern Recognit.*, Oct. 1994, pp. 164–168.

[5] N. J. Higham, *Analysis of the Cholesky Decomposition of a Semi-Definite Matrix*. Oxford, U.K.: Oxford Univ. Press, 1990.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] Y. Li, S. Wang, Q. Tian, and X. Ding, "A survey of recent advances in visual feature detection," *Neurocomputing*, vol. 149, pp. 736–751, Feb. 2015.

[8] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He, "Local feature descriptor for image matching: A survey," *IEEE Access*, vol. 7, pp. 6424–6434, 2018.

[9] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Aug. 2019.

[10] L. Lu, Y. Zhang, and P. Tao, "Geometrical consistency voting strategy for outlier detection in image matching," *Photogramm. Eng. Remote Sens.*, vol. 82, no. 7, pp. 559–570, Jul. 2016.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2004.

[12] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Comput. Vis.*, vol. 59, no. 1, pp. 61–85, Aug. 2004.

[13] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Nov. 2005.

[15] Z. Pusztai and L. Hajder, "Quantitative affine feature detector comparison based on real-world images taken by a quadcopter," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 704–715.

[16] Z. Pusztai and L. Hajder, "Quantitative comparison of affine invariant feature matching," in *Proc. 12th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2017, pp. 515–522.

[17] L. Ferraz and X. Binefa, "A sparse curvature-based detector of affine invariant blobs," *Comput. Vis. Image Understand.*, vol. 116, no. 4, pp. 524–537, Apr. 2012.

[18] V. Fedorov, P. Arias, R. Sadek, G. Facciolo, and C. Ballester, "Linear multiscale analysis of similarities between images on Riemannian manifolds: Practical formula and affine covariant metrics," *SIAM J. Imag. Sci.*, vol. 8, no. 3, pp. 2021–2069, Jan. 2015.

[19] V. Fedorov and C. Ballester, "An affine invariant patch similarity," *Image Process. Line*, vol. 8, pp. 490–513, Dec. 2018.

[20] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Procedings Alvey Vis. Conf.*, 1988, pp. 10–5244.

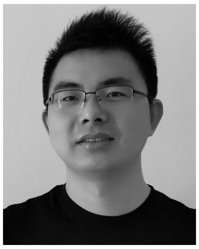
[21] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 1469–1472.

LUPING LU was born in Jiangxi, China, in 1988. She received the B.S. degree in remote sensing science and technology and the Ph.D. degree in digital photogrammetry and remote sensing from Wuhan University, China, in 2010 and 2015, respectively.

From 2015 to 2018, she held a postdoctoral position and a Research Assistant with the Collaborative Innovation Center of Geospatial Technology, Wuhan University. Her research interests include feature extraction, image matching, structure from motion, and the 3D reconstruction.

YONG ZHANG (Associate Member, IEEE) received the B.S. degree from the Department of Electrical Engineering, Southwest Jiaotong University, China, in 1993, the M.S. degree from the Applied Physics Department, Southwest Jiaotong University, in 1996, and the Ph.D. degree in digital photogrammetry and remote sensing from Wuhan University, China, in 2006.

Since 2007, he has been an Assistant Professor with the School of Remote Sensing and Information Engineering, Wuhan University. His research interests include imaging model and bundle adjustment of high resolution satellite images, dense matching algorithms, aerial triangulation of aerial images, and the 3D reconstruction of computer vision.



KAI LIU received the B.S. degree from the College of Science, Wuhan University of Technology, China, in 2012.

From 2015 to 2018, he was a Research and Development Engineer with Farsee2 Technology Company Ltd., Wuhan, China. His research interests include feature extraction, image matching, structure from motion, and the 3D reconstruction.