

# Journal of Electronic Imaging

JElectronicImaging.org

## **Rotation invariant feature lines transform for image matching**

Zhang Ye  
Qu Hongsong

# Rotation invariant feature lines transform for image matching

Zhang Ye<sup>a,b,\*</sup> and Qu Hongsong<sup>b</sup>

<sup>a</sup>Changchun Institute of Optics, Fine Mechanics and Physics, Key Laboratory of Airborne Optical Imaging and Measurement, Chinese Academy of Sciences, Changchun, China

<sup>b</sup>Changchun Institute of Optics, Fine Mechanics and Physics, National & Local United Engineering Research Center of Small Satellite Technology, Chinese Academy of Sciences, Changchun, China

**Abstract.** Image matching has been one of the most fundamental issues in computer vision over the decades. We propose a method based on utilizing feature lines in order to achieve more robust image matching, which includes feature line detection, feature vector description and matching, and the devised rotation invariant feature line transform. The feature vectors have the properties of rotation and scaling invariance. Experimental results demonstrate the effectiveness and efficiency of the proposed method. Compared with the famous powerful algorithm scale invariant feature transform, the proposed method is more insensitive to noise, and the selected distinctive locations of features are more disperse. For a certain sequence of images, which contain strong lines, the proposed method is more efficient. Using the feature lines obtained by our method, it is possible to match two scene images with different rotation angles, scales, and light distortion, and the steps of matching are simpler. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JEI.23.5.053002](https://doi.org/10.1117/1.JEI.23.5.053002)]

Keywords: scene matching; feature-lines; rotation-scaling invariance.

Paper 13573 received Nov. 5, 2013; revised manuscript received Jul. 26, 2014; accepted for publication Aug. 12, 2014; published online Sep. 5, 2014.

## 1 Introduction

The task of finding correspondences between two images of the same scene or object, taken from different times of a day or year, different sensors, and different viewing geometries under different circumstances, is an important and difficult part of many computer vision applications. In order to match differing images, researchers have been focusing on scale and image rotation invariant detectors and descriptors.

The most widely used detector probably is the Harris corner detector, which has been proposed in 1988; Harris corners are robust to changes in rotation and intensity but are very sensitive to changes in scale, so they do not provide a good basis for matching images of different sizes. Mikolajczyk and Schmid<sup>1</sup> used a Harris–Laplace detector to detect the feature points in scale-space by the Laplacian-of-Gaussian operator, thus feature points are invariant to rotation and scale changes. Lowe<sup>2</sup> used a scale-invariant detector that localizes points at local scale-space maxima of the difference-of-Gaussian (DoG) in 1999, and in 2004, he presented a method called scale invariant feature transform (SIFT) for extracting distinctive invariant features from images that can be used to perform reliable matching between images.<sup>3</sup> This approach transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or three dimensional (3-D) projection. SIFT has been proven by Mikolajczyk and Schmid<sup>1</sup> in 2003 to be the most robust among the local invariant feature descriptors with respect to different geometrical changes.

SIFT detects feature points by searching over all scales and image locations. It is implemented efficiently by using a DoG function pyramid to identify potential interest points that are invariant to scale. By rejecting the candidate feature points that have low contrast or that are poorly localized along an edge, the dominant orientation is assigned to each feature point based on local image gradient directions. The feature descriptor is created with  $4 \times 4$  arrays of histograms in the neighboring region around the point, each of which has eight orientation bins, or 128 elements in sum. The matching is often based on the distance between the vectors, e.g., the Mahalanobis or Euclidean distance. Because the dimension of the descriptor and the number of the candidate key points for matching directly contribute to the time the algorithm takes, there are algorithms<sup>4–8</sup> that can accelerate the process by increasing the performance of descriptor of the robust features, such as speed up robust features, proposed by Bay et al.<sup>9,10</sup>

All above descriptors are distinctive and invariant to some image transformations. The feature-based matching algorithms can be divided into three main steps. First, feature detection—for example of feature point, the interest points with the property of high repeatability are selected at distinctive locations in the image. Second, feature points description—the neighborhood of every interest point is represented by a feature vector, and there are lots of possible descriptors that emphasize a diverse set of image properties such as pixel intensity, gradient, color, texture, contour, edge, and so on. Furthermore, the descriptor has to be distinctive and robust to noise. Finally, the descriptor vectors are matched between different images. A fewer number of dimensions are therefore desirable.<sup>9</sup> But the matched number of the key points cannot be too few or too many. A lack of matched key points will decrease the accuracy of matching, whereas too many

\*Address all correspondence to: Zhang Ye, E-mail: [zhangye@ciomp.ac.cn](mailto:zhangye@ciomp.ac.cn)

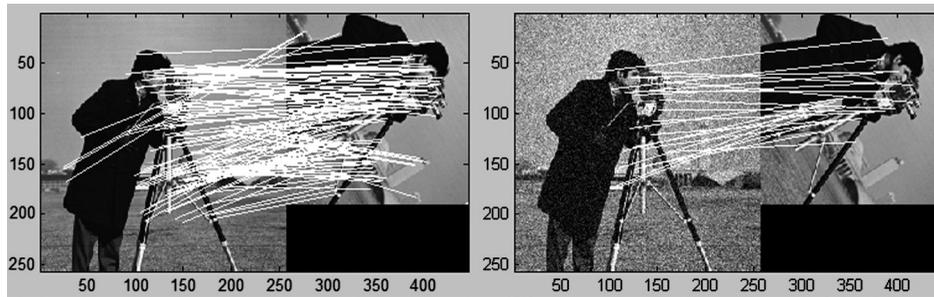


Fig. 1 Scale invariant feature transform (SIFT) matching result.

will cost computational time. Because of these facts, one has to strike a balance between the above requirements, for example, by reducing the descriptor's dimension and complexity, while keeping it sufficiently distinctive.

Figure 1 shows the SIFT matching result. There are 129 pairs of key points being matched correctly, as is shown by Fig. 1(a). However, when we add Gaussian noise with  $\sigma = 0.03$  to the image on the left, the number of the correct matching pairs is reduced to 26, which implies that the SIFT is somewhat sensitive to noise. In addition, the locations of the key points are concentrated.

This paper proposes a method for image matching based on feature lines rather than feature points. Kostadin et al.<sup>11</sup> has proposed a novel image denoising strategy based on an enhanced sparse representation in transform domain, by grouping similar two-dimensional image fragments into 3-D data arrays; his algorithm adapts for natural scenes. A strong similarity between small image blocks at different spatial locations is indeed very common in natural images, but our method adapts for artifices, which always contain abundant lines and clear boundaries. This paper is organized as follows: Sec. 2 describes the scheme of feature lines detection, in which distinctive feature lines with specific orientations can be detected; Sec. 3 explains polar transform of the neighborhood region of the feature lines, and a new rotation/scale invariant descriptor is also presented. Section 4 describes the matching work. Section 5 presents the experimental results, and Sec. 6 concludes the paper.

## 2 Feature Lines Detection

Human vision system (HVS) includes human eyes system and human visual cortex filter system. The former can be modeled as log-polar transform, and the latter can be viewed as a range of spot and bar filters. The elementary processing method of HVS is the collected image by the human eyes system goes through the human cortex cells. These cells are not sensitive to the light of the image, but are sensitive to certain oriented structure. Thus, they are also called orientation harmonious cells, whose function, simply put, is filtering. Just like a range of spot and bar filters, some of the cortex cells respond strongly to oriented structure and weakly to other patterns. Hubel and Wiesel reason that these cortex cells function on the basis of lines or edges.<sup>10</sup> By analogy with the human visual cortex, it is common to use at least one spot filter and a collection of oriented bar filters at different orientations, scales, and phases. After filtering, the HVS synthesizes the information resulted from all the cortex cells to make the final decision.

This work was inspired by the HVS. A new method for scene matching based on feature lines orientations,

description, and matching is presented in this paper. We filter the image utilizing oriented linear Gaussian filter banks convolution to imitate the function of the cortex cells. The Gaussian filter bank has been shown in Fig. 2. Mid-grey level is represented by zero, with brighter values being positive and darker values being negative.

It is still unknown how many filters are "best" for useful texture algorithms. Perona listed the number of scales and orientation used in a variety of systems, ranging from 4 to 11 scales and from 2 to 18 orientations. The number of orientations varies from application to application and does not seem to make a big difference, as long as there are at least six orientations. Typically, the "spot" filters are Gaussians, and the "bar" filters are obtained by differentiating oriented Gaussians.<sup>12</sup> Using more filters leads to more details, but we must also convolve the image with more filters, which can be more expensive.

In this article, we utilize a range of six oriented filters; the orientations are 0, 30, 60, 90, 120, and 150 deg. There are six versions of these bars, each of which is a rotated version of a horizontal bar.

The Gaussians in the horizontal bar have weights  $-1$ ,  $2$ , and  $-1$ . They have different  $\sigma$  in the  $x$  and  $y$  directions; the  $\sigma_x$  values are all 2, and the  $\sigma_y$  values are all 1. The centers are offset along the  $y$  axis, lying at points  $(0, 1)$ ,  $(0, 0)$ , and  $(0, -1)$ . This filters' bank has been used by Malik and Perona in 1990.<sup>12</sup>

The results after applying each of the filters in Figs. 2 to 3(a) are shown in Fig. 3(b) as absolute values of the output. Brighter pixels represent stronger responses, and the images are laid out in correspondence with each of the filter position in Fig. 2.

After the image is convolution filtered by the oriented linear filters, there is a strong response when the image

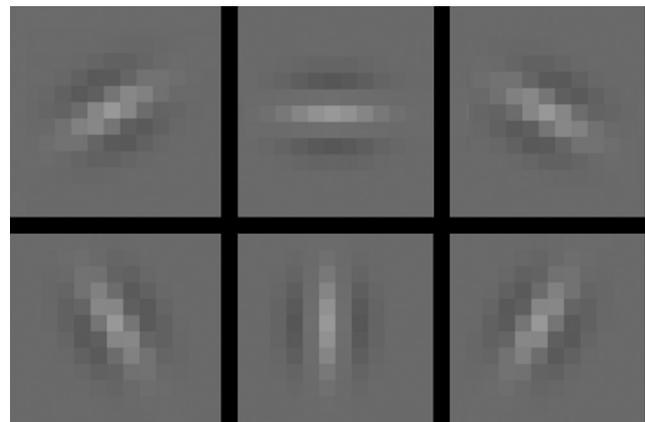
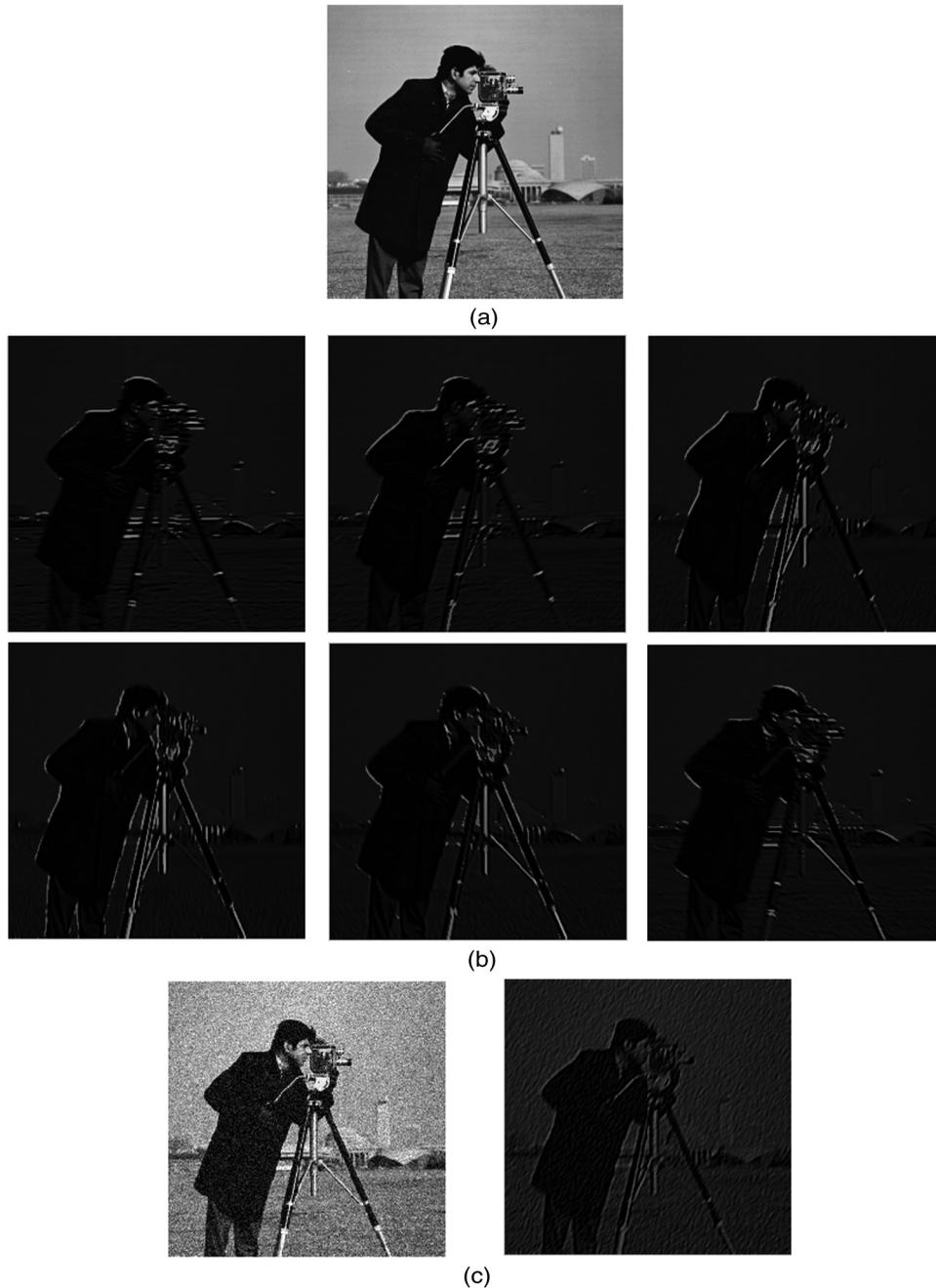


Fig. 2 Oriented Gaussian filters bank.



**Fig. 3** (a) Original image, (b) the responses of the filters of Fig. 2, (c) the responses of the filters under noise.

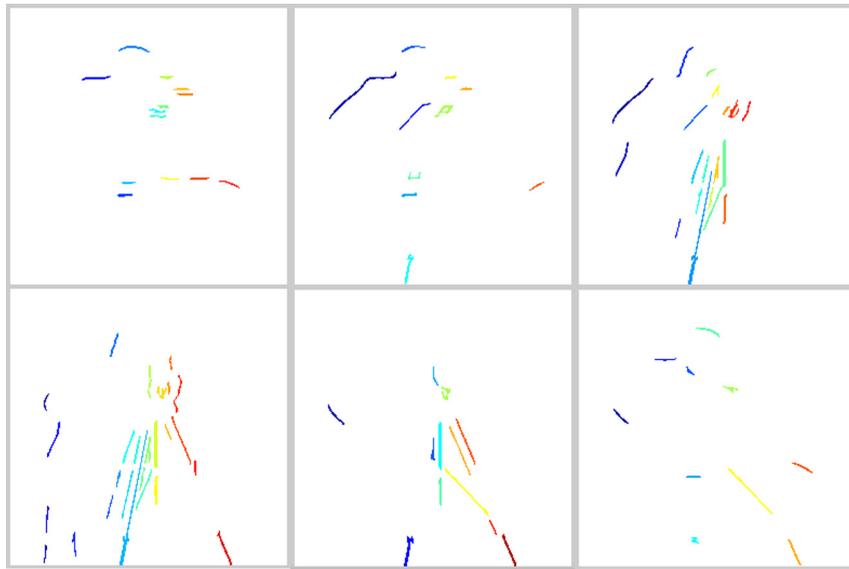
pattern in a neighborhood looks similar to the filter kernel and a weak response when it does not, as shown in Fig. 3, and it can be hardly affected by the noise, as shown in Fig. 3(c). The advantage of this approach is that it is easy to search simple pattern elements, bars, by filtering an image. The brightness of pixels expresses responses. Then, we threshold<sup>13</sup> to extract the light feature lines. The threshold value can be automatically computed by analyzing the image histogram, by obtaining the entire information contained in the histogram, the light lines with certain orientation can be extracted. Figure 4 shows the thresholding results.

After automatically thresholding,<sup>13,14</sup> we can get six pictures with extracted feature lines by six oriented linear filters, as shown in Fig. 4.<sup>15</sup>

The number of the Oriented Gaussian filters can also be 4 as Fig. 5 shows and the extracted features lines showed in Fig. 6. One image has a range of four or six pictures of feature lines; take the feature lines in one picture as one group. If we find the correspondingly oriented pictures of the other image, the matching work will be simpler and more credible. In other words, if the approximate rotation angle of the two images could be estimated first, or one group of feature lines has been matched to another image, the matching order will be definite.

One way to estimate the approximate rotation angle of two images is using Fourier–Mellin transform (FMT).

Consider two images  $f_1(x, y)$  and  $f_2(x, y)$ ,  $f_2(x, y)$  is a rotated, scaled, and translated replica of  $f_1(x, y)$ :



**Fig. 4** Extract feature lines by oriented linear Gaussian filter.

$$f_2(x, y) = f_1[\sigma(x \cos \alpha + y \sin \alpha) - x_0, \sigma(-x \sin \alpha + y \cos \alpha) - y_0], \quad (1)$$

where  $\alpha$  is the rotation angle,  $\sigma$  is the uniform scale factor, and  $x_0$  and  $y_0$  are the translational offsets.

The Fourier transforms  $F_1(u, v)$  and  $F_2(u, v)$  of  $f_1(x, y)$  and  $f_2(x, y)$  are related by

$$|F_2(u, v)| = \sigma^{-2} |F_1[\sigma^{-1}(u \cos \alpha + v \sin \alpha), \sigma^{-1}(-u \sin \alpha + v \cos \alpha)]|, \quad (2)$$

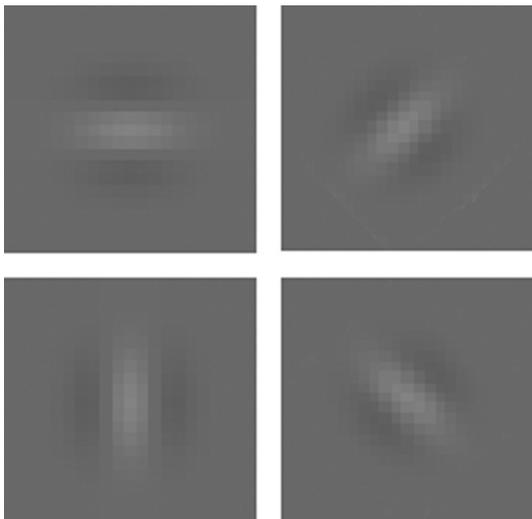
where  $|*|$  presents the spectral magnitude. Equation (3) shows that the image rotates the spectral magnitude by the same angle. This method is called Fourier–Mellin invariant descriptor. Moreover, we can use the log-polar transforms of the two spectral magnitude images to compute the rotation angle  $\alpha$  and the scaled factor  $\sigma$ . Let

$$\ln \rho = \ln(\sqrt{u^2 + v^2}); \quad \theta = \arctan(v/u).$$

Then the relation can be written as follows:

$$F_2(\theta, \log \rho) = F_1(\theta - \alpha, \log \rho - \log \sigma). \quad (3)$$

In this log-polar representation, both rotation and scaling are reduced to translations, as proposed by Pratt<sup>16</sup> and Schalkoff.<sup>17</sup> By Fourier transform, the log-polar representations, the rotation, and scaling appear as phase shifts. The log-polar mapping of the spectral magnitude corresponds to physical realizations of the FMT.<sup>18</sup> Human visual systems appear to have some similarity with this log-polar mapping.



**Fig. 5** Four oriented Gaussian filters bank.



**Fig. 6** Extract feature lines by less linear Gaussian filter.

The phase correlation matrix of the two transformed images is computed by

$$\frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|} = e^{j(u x_0 + v y_0)}, \tag{4}$$

where  $x_0$  and  $y_0$  are the amount of horizontal and vertical translations between the two transformed images. Computing the inverse Fourier transform of the phase correlation matrix, the location of a sharp correlation peak identifies the translation, which represents the rotation and scaling between the two images. Figure 7 shows an example.

According to this method, we can approximately estimate the rotation angle of the two images, and the directional pictures of feature lines can be correspondingly assigned. Then, the matching work will be simpler and more credible.

But FMT is not effective for two images with big range of zooming, rotation, and transformation because FMT is a method that needs to use the whole information contained in the images. Difference in contents of two images will lead to the erroneous matching result, as is shown in Fig. 8, in which the Dirac peak has been submerged in the noises. In this condition, we should match all the six or four oriented ranges without the information of matching order.

### 3 Feature Line Descriptor

Polar transform representations play an important role in image processing and analysis. In the spatial domain, log-polar schemes have been used to model the strongly inhomogeneous sampling of the retinal image by the HVS.<sup>19</sup> Polar transform has two principal advantages: rotation invariance and scale invariance. Lines through the center of the image will be mapped into horizontal lines in the polar image and the concentric circle into vertical lines. Thus, the target rotation results in vertical translation, and the change of the target size results in horizontal translation. In other words, translating the polar image with a near edge causes the rotation of the original image. Therefore, we can transform the rotation variation to translational variation.

In order to describe the feature lines in an invariance way, we take each feature line orientation  $\theta(x, y)$  as 0 deg, establish polar coordinate to transform the neighboring region of the feature line to polar image, take the feature line center as the center of the circle, and make two-thirds of the length of the feature line  $r(x, y)$  as the radius, as shown in Fig. 9.

For each feature line, the orientation  $\theta$  and length  $r$  can be computed as follows:

$$\begin{aligned} \theta &= \tan^{-1}[(y_1 - y_2)/(x_1 - x_2)] \\ r &= \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \end{aligned} \tag{5}$$



Fig. 7 Utilize Fourier–Mellin transform (FMT) to roughly estimate the rotation angle of the two images.

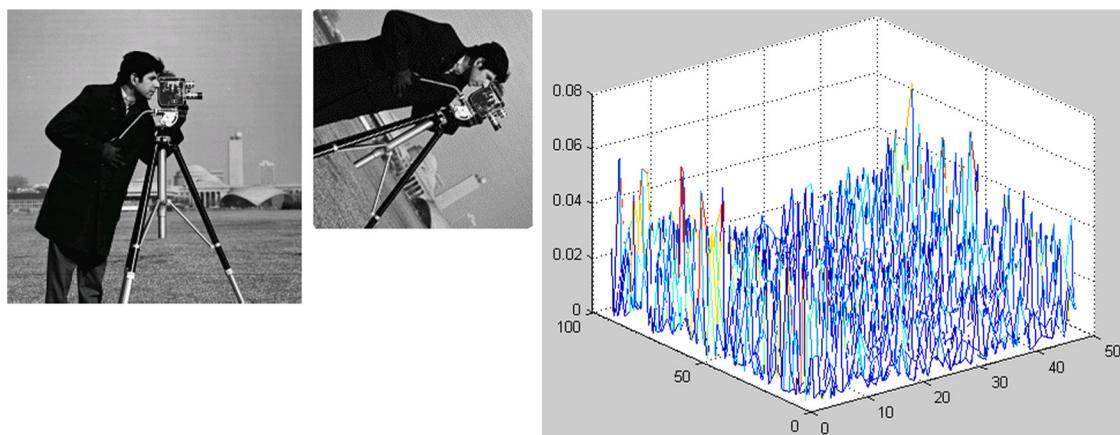


Fig. 8 FMT cannot estimate approximate rotation angles for images with big changes.

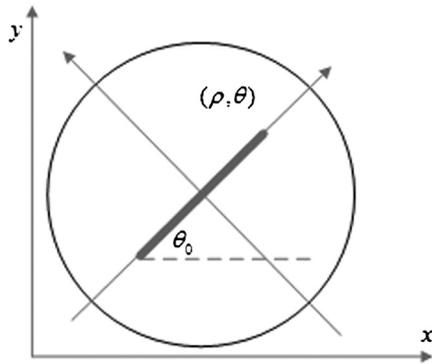


Fig. 9 Establish polar coordinate for each feature line.

where  $(x_1, y_1)$  and  $(x_2, y_2)$  are extreme points of the feature line.

Because the polar coordinate takes the line orientation as the base, the polar transform is invariance to rotation angles. As the neighboring region takes the line's length as the base, it is invariance to scaling as well. Then the circle neighboring region image is utilized to create a vector to describe the feature line. The formulas of polar transform are

$$\rho = (\sqrt{u^2 + v^2}) \quad \theta = \arctan(v/u). \quad (6)$$

Figure 10 shows the transformed polar images of the feature lines. Although the pair of feature lines has different orientations and scales, the normalized transformed polar images are almost the same.

#### 4 Local Polar Image Descriptor

The next step is to compute a descriptor for the local polar image region that is highly distinctive. One obvious approach would be to describe local polar images in a coarse-to-fine iterative strategy (pyramid approach) and to match them using a normalized correlation measure. By casting the polar image into a multiresolution framework, most of the iterations are carried out at the coarsest level, where the amount of data is greatly reduced.<sup>20</sup> This results in a considerable saving of computational time. However, simple correlation of image patches is highly sensitive to changes that may cause wrong registration of samples, such as affine or 3-D viewpoint change or nonrigid deformations.<sup>1,21</sup>

In this article, a feature line descriptor is created by first computing the gradient magnitude and orientation at each polar image. Then, the sample points are accumulated into orientation histograms, as shown in Fig. 11. The

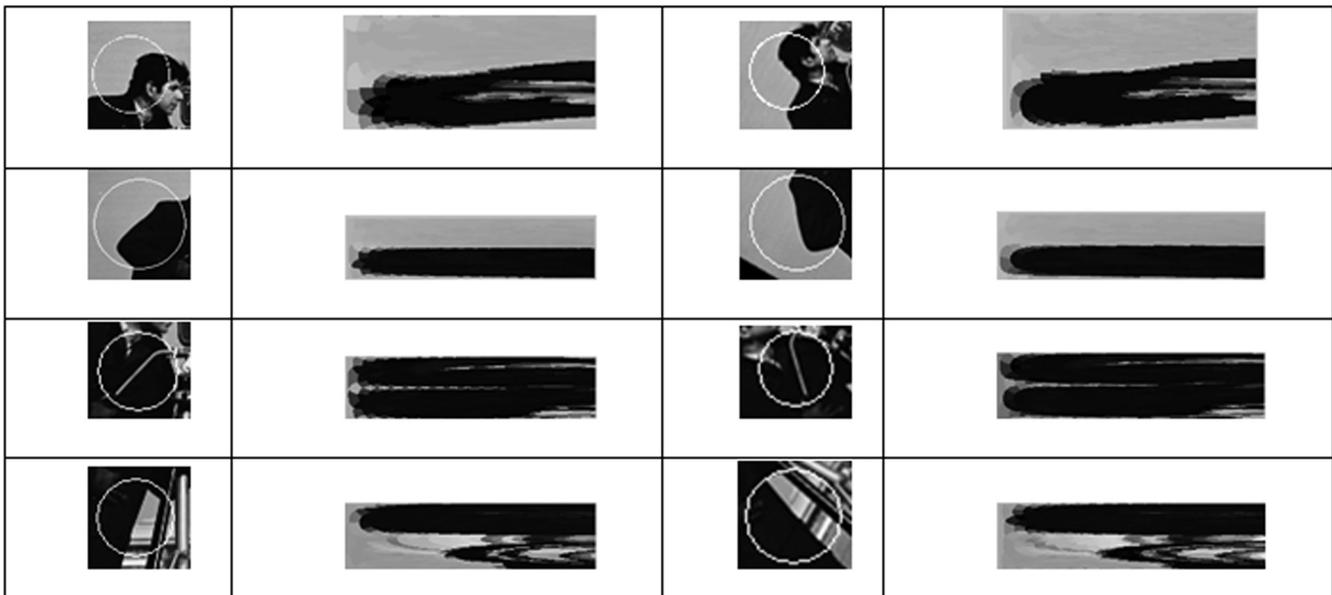


Fig. 10 Four pairs of feature lines neighborhood transform.

length of each arrow corresponds to the sum of the gradient magnitudes approximating that direction within the region.

There are three parameters that can be chosen to vary the complexity of the descriptor: the number of orientations,  $r$ , in the histograms, the width,  $w$ , and the height  $h$  of the  $w \times h$  array of orientation histograms. The size of the resulting descriptor vector is  $r \times w \times h$ . As the complexity of the descriptor grows, it will be able to discriminate better in a large database but will also be more sensitive to shape distortions and occlusion. We can use a  $4 \times 8 \times 4 = 128$  element feature vector for each feature lines, as shown in Fig. 11. As the region of the image near the center is over-sampled by polar transform, there is no more information that can be used to match the left row vector that can be deleted. The eventually used number of the feature vector elements is  $4 \times 7 \times 4 = 112$ .

### 5 Feature Line Matching

The best candidate match for each feature line is found by identifying its nearest neighbor in the database of feature lines from training images. The matching work is similar to SIFT, an effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. Suppose there are multiple training images of the same object, the second-closest neighbor is defined as being the closest neighbor that is known to come from a different object than the first, such as by only using images known to contain different objects. This measure performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances because of the high dimensionality of the feature space. We can consider the second-closest match as providing an estimate of the density of false matches within this portion of

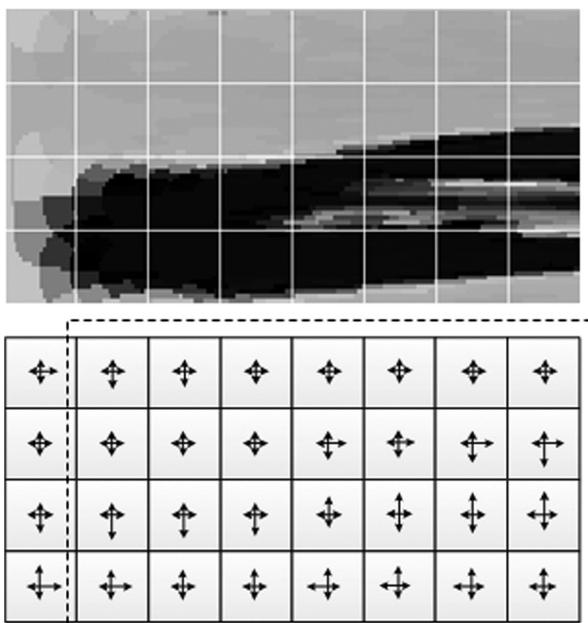


Fig. 11 Feature lines descriptor.

the feature space and at the same time identifying specific instances of feature ambiguity.<sup>1</sup>

## 6 Experiment Results and Algorithms Analysis

### 6.1 Experiment Results

We coined the proposed method rotation invariant feature-line transform (RIFLT). Figure 12 shows the matching result. Figure 12(a) is the SIFT matching result, and Fig. 12(b) is the RIFLT matching result. We can see that the quantities of correct matching pairs are almost the same, but the RIFLT has two benefits: first, the RIFLT matched pairs are lines, not points; it is obvious that one line contains quite a few points; the RIFLT will contain more information. Second, because the matched feature points by SIFT are too concentrated, it is not good for image matching to compute the translation of the two images; the RIFLT feature lines are more disperse. Why the SIFT key points are so concentrated and compared with RIFLT? Does the Gaussian noise make it? Fig. 13 shows the location of the RIFLT feature lines are also disperse, compared with SIFT.

We present an experiment of two simple figures, as shown in Fig. 14, one figure contains a small rectangle, and another is its rotated replica. If SIFT is utilized to match the two figures, there is only one key point, the location of the key point is in the rectangle's center. By contrast, if RIFLT is utilized to match the two figures, there are four feature lines, the locations of the feature lines are on the edge of the rectangle. SIFT finds key points inside the change of gradients, but RIFLT find feature lines on the change of gradients, and the latter selects features that will be more disperse.

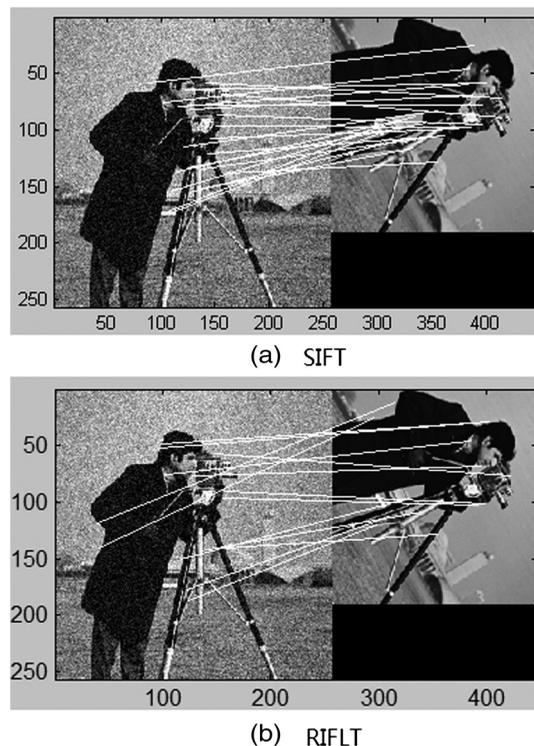
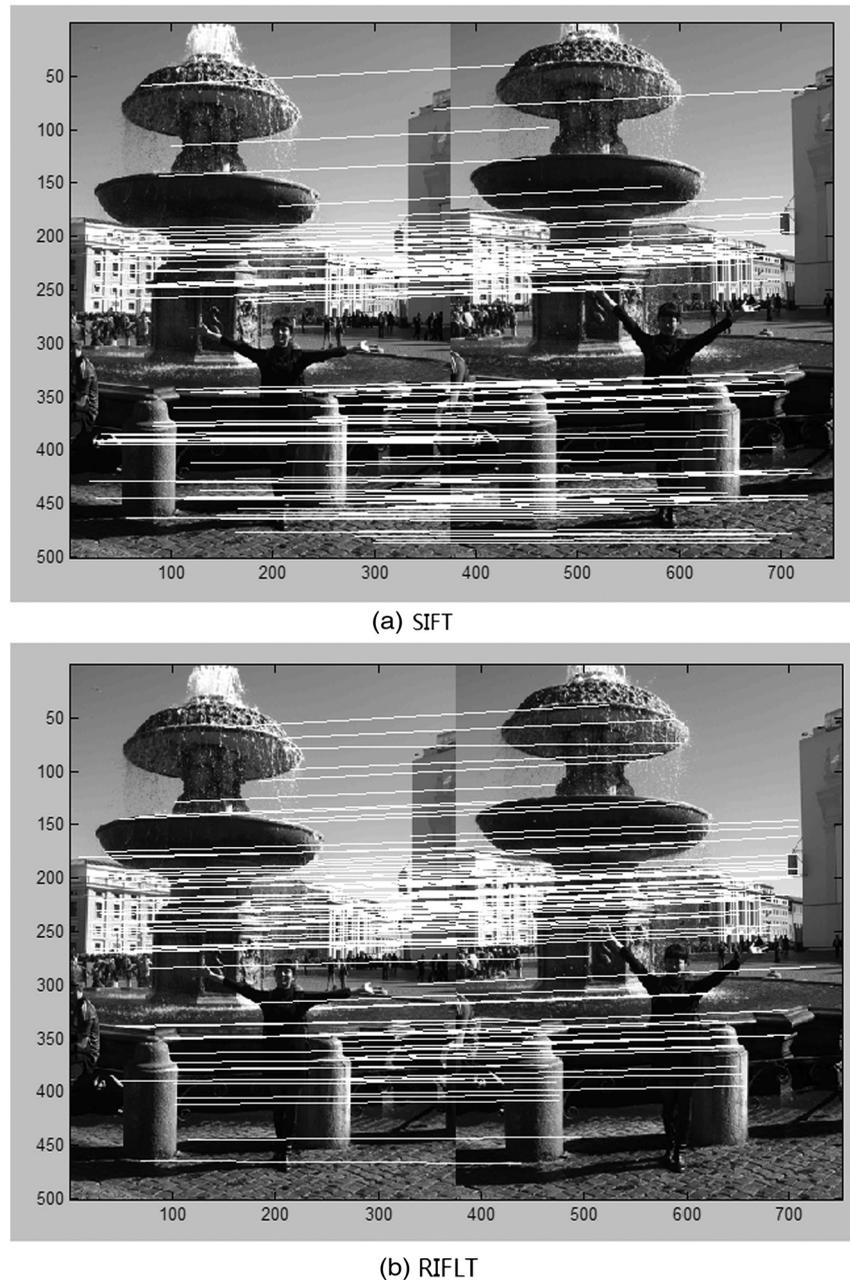


Fig. 12 Experiment results (a) utilize SIFT algorithm and (b) utilize rotation invariant feature line transform (RIFLT) algorithm proposed in this paper.



**Fig. 13** Experiment results (a) utilize SIFT algorithm and (b) utilize RIFLT algorithm proposed by this paper.

If the rectangle has been drew larger, SIFT will find more key points; the key points are under different scale spaces, but the key points are all selected by the surrounding gradient changes—the four lines. Although the quantities of the selected key points are more than the feature lines, it will make a mistake if the gradient changes are similar, as shown in Fig. 15.

## 6.2 Complexity Analysis

RIFLT includes four steps: (1) feature line extraction, (2) feature line description, (3) polar image description, and (4) feature line matching. There are three benefits of utilizing feature lines to match images. First, lines are always strong features, i.e., they are insensitive to noise; second, lines

contain size and orientation information, and the line description steps may be simpler; third, the matched number of the feature lines can be fewer compared with the method based on key points, but the feature lines can be more effective because they are composed of numerous points. What we do is describing each line and its neighborhood by a rotation and scaling invariant feature vector. If we match two images according to key lines, each line contains orientation and scale information. We need not to match too many pairs of lines. If the candidate lines are too many, we can threshold the length of the lines and reduce the short lines and the lines too long. For the picture “cameraman,” SIFT may match 129 pairs of key points, but RIFLT may only match 13 key lines of the picture. The estimating of the transform between the two images by both the methods is almost the same.

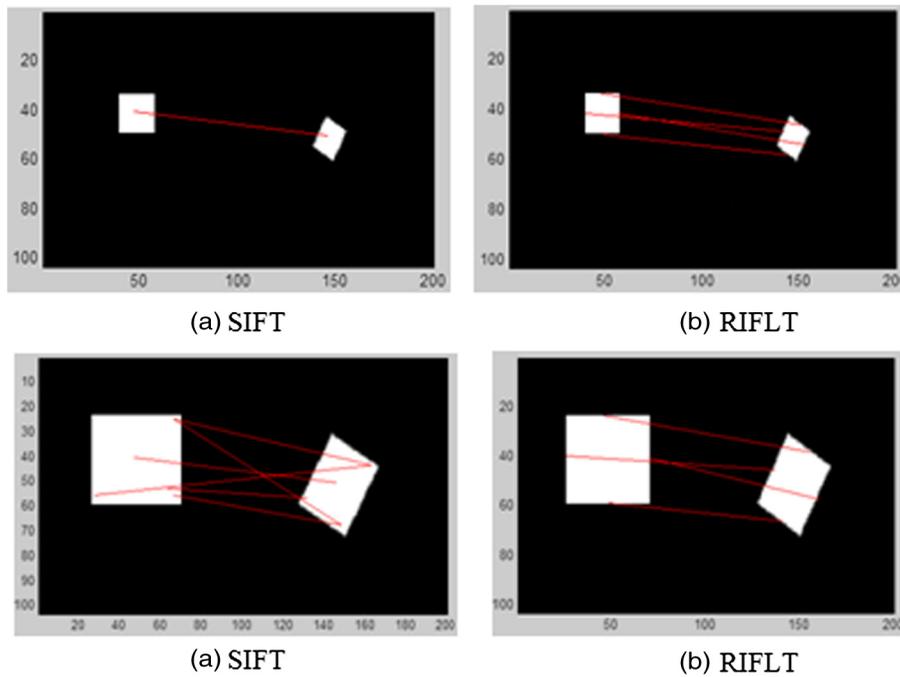


Fig. 14 (a) SIFT and (b) RIFLT for simple figure matching.

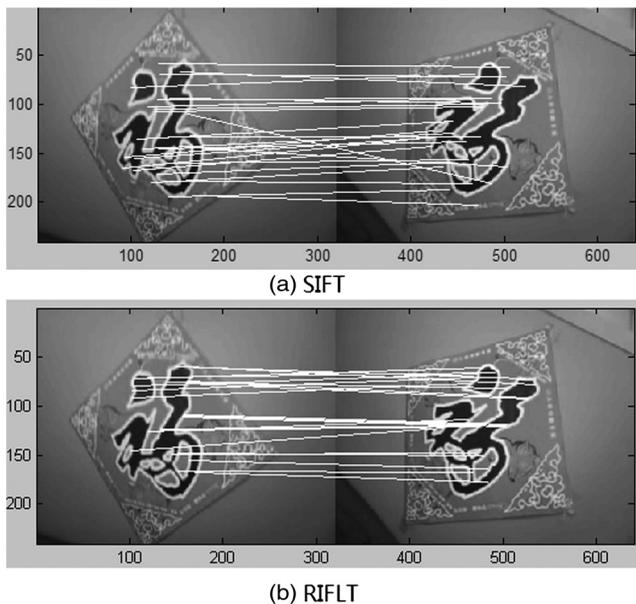


Fig. 15 Experiment results (a) utilize the SIFT algorithm and (b) utilize the RIFLT algorithm proposed in this paper.

### 7 Conclusions

In this paper, we present a feature line detection and description method for image matching. The feature lines have been shown to be invariant to image rotation and scale, and robust across a substantial range of affine distortion, addition of noise, and change in illumination. According to the position of the feature lines, the parameters that represent rotation angle, scaling quantity, and the transformation value can be computed. Following are the major stages of computation used to generate the set of image features:

1. Feature line detection—utilize oriented bar filters;
2. Feature line descriptor computation—utilize polar transform and local gradient histogram;
3. Feature line vectors matching—utilize the information of closest and second closest matching;
4. Estimate image transform—utilize the locations of the feature lines.

Future work will aim at optimizing the line descriptor computation work for speed up, uniting feature points and feature lines for more robust image matching, and imitating HVS.

### Acknowledgments

The authors would particularly like to thank Yanjie Wang, Hongsong Qu, Jiade Li, and Ning Xu for their numerous suggestions to both the content and presentation of this paper. They also thank the National Natural Science Foundation of China (Grant No. 60902067), and the Foundation of Key Laboratory of Airborne Optical Imaging and Measurement, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences for supporting this research.

### References

1. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005).
2. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Seventh IEEE International Conf. on Computer Vision*, Vol. 2, pp. 1150–1157 (1999).
3. D. G. Lowe, "Distinctive image feature from scale-invariant keypoints," *Int. J. Computer Vision* **60**(2), 91–110 (2004).
4. C. Li and L. Ma, "A new framework for feature descriptor based on SIFT," *Pattern Recognit. Lett.* **30**(5), 544–557 (2009).
5. Z. Luan, W. Yuan-qin, and T. Jiu-Bin, "Improved algorithm for SIFT feature extraction and matching," *Optics Precis. Eng.* **19**(6), 1391–1397 (2011).

6. H. Bai-Gen and Z. Ming, "Improved fully affine invariant SIFT-based image matching algorithm," *Optics Precis. Eng.* **19**(10), 2472–2477 (2011).
7. W. Mei, T. Da-Wei, and Z. Xuchao, "Moving object detection by combining SIFT and differential multiplication," *Optics Precis. Eng.* **19**(4), 892–899 (2011).
8. Z. Qidan et al., "Calculation of object rotation angle by improved SIFT," *Optics Precis. Eng.* **19**(7), 1669–1676 (2011).
9. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," *Computer Vision and Image Understanding* **110**(3), 346–359 (2008).
10. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.* **160**, 106–154 (1962).
11. K. Dabov et al., "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007).
12. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey (2002).
13. Z. Ye, Q. Hongsong, and W. Yanjie, "Adaptive image segmentation based on fast thresholding and image merging," in *16th International Conference on Artificial Reality and Telexistence-Workshops (ICAT)*, pp. 308–311, IEEE (2006).
14. Z. Ye and W. Yanjie, "High accuracy real-time automatic thresholding for centroid tracker," *Proc. SPIE* **6027**, 60272P (2005).
15. Z. Ye, Q. Hongsong, and W. Yanjie, "Implementation of scene matching based on rotation invariant feature lines," *Optics Precis. Eng.* **7**, 1759–1765 (2009).
16. W. K. Pratt, *Digital Image Processing*, pp. 526–566, John Wiley & Sons, New York (1978).
17. R. J. Schalkoff, *Digital Image Processing and Computer Vision*, pp. 279–286, John Wiley & Sons, New York (1989).
18. Q.-S. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(12), 1156–1168 (1994).
19. A. Taberero, J. Portilla, and R. Navarro, "Duality of log-polar image representations in the space and spatial-frequency domains," *IEEE Trans. Signal Process.* **47**(9), 2469–2479 (1999).
20. P. Thevenaz, U. E. Ruttimann, and M. Unser, "A pyramid approach to subpixel registration based on intensity," *IEEE Trans. Image Process.* **7**(1), 27–41 (1998).
21. C. Li and L. Ma, "A new framework for feature descriptor based on SIFT," *Pattern Recognit. Lett.* **30**, 544–557 (2009).

**Zhang Ye** received her PhD degree in mechatronic engineering from the Chinese Academy of Sciences in 2008. She is currently an associate researcher at the Key Laboratory of Airborne Optical Imaging and Measurement, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. Her research interests lie in the real-time digital image processing and pattern recognition.

**Qu Hongsong** received his PhD degree in mechatronic engineering from the Chinese Academy of Sciences in 2008. He is currently an associate researcher at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests lie in image optics and remote sensing.