

Stripe Model: An Efficient Method to Detect Multi-form Stripe Structures

Yi Liu^{1,2}, Dongming Zhang¹, Junbo Guo¹, and Shouxun Lin¹

¹ Advanced Computing Research Laboratory, Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

² Graduate University of Chinese Academy of Sciences, Beijing 100049, China
{liuyi, dmzhang, guojunbo, sxlin}@ict.ac.cn

Abstract. We present a general mathematical model for multiple forms of stripes. Based on the model, we propose a method to detect stripes built on scale-space. This method generates difference of Gaussian (DoG) maps by subtracting neighbor Gaussian layers, and reserves extremal responses in each DoG map by comparing to its neighbors. Candidate stripe regions are then formed from connected extremal responses. After that, approximate centerlines of stripes are extracted from candidate stripe regions using non-maximum suppression, which eliminates undesired edge responses simultaneously. And stripe masks could be restored from those centerlines with the estimated stripe width. Owing to the ability of extracting candidate regions, our method avoids traversing to do costly directional calculation on all pixels, so it is very efficient. Experiments show the robustness and efficiency of the proposed method, and demonstrate its ability to be applied to different kinds of applications in the image processing stage.

Keywords: stripe model, scale-space, difference of Gaussian, non-maximum suppression.

1 Introduction

The stripe, curvilinear structure of a certain width, is a fundamental component of many objects, such as character strokes, balustrades, roads, pipelines etc. So stripe detection plays an important part in the image processing stage of computer vision applications, e.g., it could be used to extract strokes in text detection as preprocessing, find roads in aerial images, or detect tubular objects in medical and industrial fields.

Early methods detect stripes by considering the gray values of the image only and using purely local criteria [1, 2]. They aim at stripes of a very small width, i.e. lines, and their performance is deeply constrained by stripe direction [1]. In order to detect wide curving stripes, they have to zoom out the image and combine results in all possible directions, which leads to high computational cost and low robustness.

Moreover, the stripe body could even consist of dense subcomponents as shown in Fig. 1, and they could be blurred due to the compression in real images and videos.

Under such a condition, a group of generalized methods based on derivatives of Gaussian kernels [3, 4] could be applied.



Fig. 1. Different stripe forms from left to right: narrow, different brightness, wide, blur, and dense subcomponent

Koller et al. proposed a multi-scale filter [3] that detects both the left and right edge responses of stripes, combines them nonlinearly and iterates in scale-space to detect stripes of arbitrary widths. The advantage of this approach is that, since the particular nonlinear scheme is used, it can subtly resolve the side effects of edges, as occurs for every linear filter based on derivatives of Gaussian kernels. However, due to its inability of locating candidate stripe regions, it has to apply the whole procedure on every pixel while extending 1D theory to 2D situation, which is redundant and takes high computational cost.

Steger gave a comprehensive elucidation on behavior of stripes under derivatives of Gaussian kernels and introduced an unbiased detector [4] building upon previous works on multi-scale ridge detection [3, 5, 6, 7]. The detector locates stripes according to the first directional derivative and the Laplacian of Gaussian (LoG) response, and it generates meaningful results in consideration of asymmetry of edges. But it is also computationally expensive to traverse all pixels the same as [3].

Most recently, Epshtein et al. put forward the stroke width transform (SWT) to detect character strokes [8], which uses Canny filter to find edges and retrieves the stripe width information by concatenating edge points along the gradient direction. Though making a great progress on the text detection task in nature scenes, it is unfit for noisy or blurred condition as reported, because SWT heavily depends on the edge detection.

We give a summary in Table 1 to compare the ability of three popular methods in different condition.

Table 1. Ability comparison among different methods

Method	Width selection	Blur condition	Subcomponents	Less sensitive to noise	Efficiency
Koller [3]	√	√	√	√	×
Steger [4]	×	√	√	√	×
Epshtein [8]	√	×	×	×	√

Consequently, there lacks of a method that is efficient and can fulfill multi-form stripes. In this paper, by taking advantage of most recent achievements, we define the stripe structure and build a mathematical model to illustrate its characteristics. Then, we propose an efficient and robust scheme to detect stripes of diverse forms.

Comparing to the analogue and most popular ones [3, 4, 8], our method does the directional calculation only on candidate stripe regions which indeed speeds up the processing, and the scale-space based scheme makes it flexible to screen or select stripes of certain widths.

2 Stripe Detection

2.1 Stripe Model

Our stripe model is motivated by the characteristics of the difference of Gaussian (DoG) response of an edge, which we will illustrate with 1D signal, and then extend to 2D. The DoG function provides a close approximation to the scale normalized LoG, $\sigma^2 \nabla^2 G$ [9], as studied by Lindeberg, and Lowe use it to build the image pyramid [10], which is so efficient and will be utilized in our scheme later.

Here, we define the stripe structure as the curvilinear one of a certain width, and the ideal 1D profile is given by

$$f(x) = au(x + \frac{w}{2}) + bu(-x + \frac{w}{2}) + c \tag{1}$$

where $u(x)$ is the unit step signal, w is the stripe width, $a, b, c \in R$ and $ab > 0$.

For $u(x)$ could be seen as the profile of an ideal edge in 1D, to exemplify the features of the edge response under DoG filter, we construct $g(x) = au(x)$ with amplitude a and convolve it with DoG function

$$h_{step}(x; \sigma) = DoG(x; \sigma) * g(x) = a \int_{-\infty}^x DoG(t; \sigma) dt \tag{2}$$

where

$$DoG(x; \sigma) = G(x; k\sigma) - G(x; \sigma) \tag{3}$$

$$G(x; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{4}$$

$k > 1$ is a constant multiplicative factor of two nearby scales.

Take the derivative of $h_{step}(x; \sigma)$ with respect to x and set it to zero, giving

$$x_1 = k\sigma \sqrt{\frac{2 \ln k}{k^2 - 1}}, \quad x_2 = -k\sigma \sqrt{\frac{2 \ln k}{k^2 - 1}} \tag{5}$$

In consideration of the second derivative of $h_{step}(x; \sigma)$, we can conclude the offsets of the extremal DoG responses of a step signal are x_1 and x_2 , which only depend on scale σ but are irrelevant to amplitude a , shown in Fig. 2.

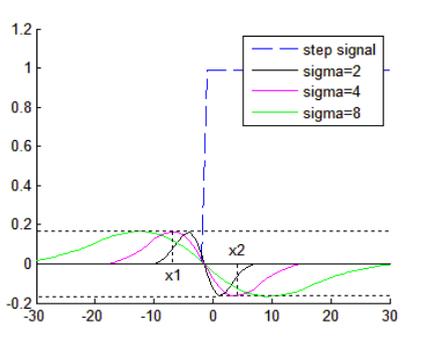


Fig. 2. DoG responses of a step signal in multi-scale with $a = 1$

Proposition 1. The offsets of the extremal DoG responses of a step signal only depend on the scale of DoG filter but are irrelevant to the signal amplitude.

Bring x_1 back to Eq. 2, we get the extremal response

$$\hat{h}_{step}(x_1; \sigma) = \frac{a}{2} [\phi(\sqrt{\frac{\ln k}{k^2 - 1}}) - \phi(k\sqrt{\frac{\ln k}{k^2 - 1}})] \tag{6}$$

where

$$\phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{7}$$

So, we find \hat{h}_{step} only depends on amplitude a and is irrelevant of scale σ .

Proposition 2. The values of the extremal DoG responses of a step signal only depend on the signal amplitude but are irrelevant to the scale.

Now, given a square signal, the 1D profile of an ideal stripe structure with a width w , it could be seen as a linear combination of two shifted step signals, as Eq. 1. For DoG filter is a linear operator, the final response of the square signal is also a linear combination of that two step signal responses

$$\begin{aligned} h_{square}(x, \sigma) &= DoG(x; \sigma) * f(x) \\ &= aDoG(x; \sigma) * u(x + \frac{w}{2}) + bDoG(x; \sigma) * u(-x + \frac{w}{2}) \end{aligned} \tag{8}$$

Take the result into account along with proposition 1 and 2, we infer that there must be a scale σ , at which the DoG response in the center of the square signal reaches the extremum as shown in Fig. 3. And bring $x_1 = w / 2$, the extremum offset, back to (5), we get the proper scale

$$\hat{\sigma} = \frac{w}{2k} \sqrt{\frac{k^2 - 1}{2 \ln k}} \tag{9}$$

At the same time, the values in the center neighborhood also reach their extrema due to the continuity property of the response. So we can summarize the feature of the stripe model in 1D as:

Proposition 3. Given a square signal of the width w , DoG responses in the center and its neighborhood reach extrema at scale $\sigma = \hat{\sigma}$.

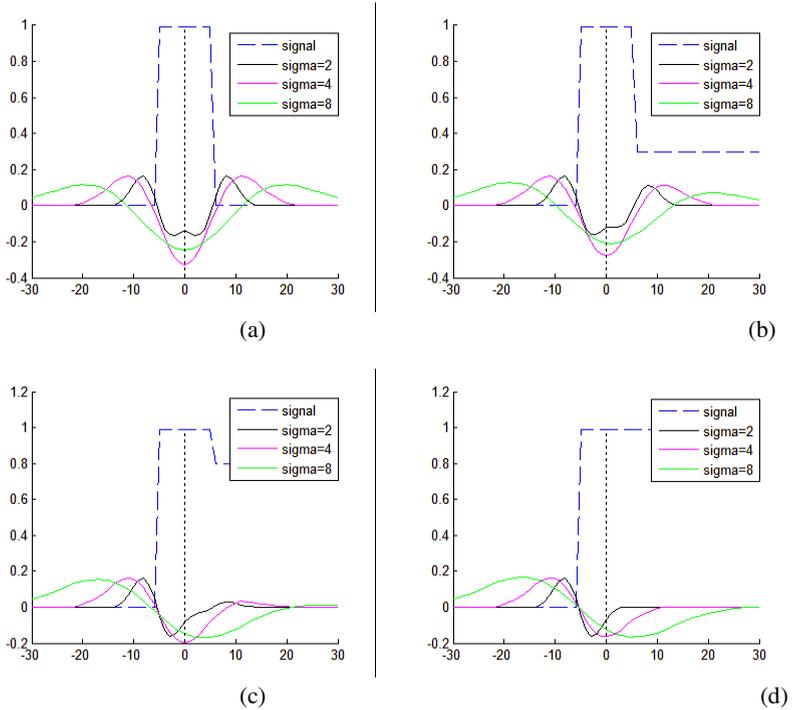


Fig. 3. DoG responses of a square signal in multi-scale with different a and b configurations. (a) shows that symmetrical edges generate corresponding responses. When $a \neq b$, though do not affect the extrema, they turns biased (b, c). (d) gives a limit case that $b = 0$ which degrades to the step signal situation.

Now, we will discuss the blurred condition. As mentioned before, this model fulfills blurred stripes which could be seen as a Gaussian smoothing ones

$$f_s(x; \sigma_s) = G(x; \sigma_s) * f(x) \tag{10}$$

According to the semi-group property of Gaussian kernel [11], the convolution of two Gaussian kernels is the same as that of one kernel with different scale

$$G(\cdot; \sigma) * G(\cdot; \sigma_s) = G(\cdot; \sigma + \sigma_s) \tag{11}$$

So the final response turns to be $h_{square}(x; \sigma + \sigma_s)$ in accordance with Eq. 11, and all properties described above still hold with the corresponding scale rising to $\sigma + \sigma_s$ concomitantly.

Since the previous analysis is based on 1D, we need extend to 2D to apply to images. Using the separability property of 2D Gaussian kernels, convolving the stripe with a 2D Gaussian kernel is the same as convolving it with a 1D one in the stripe direction first and then in the perpendicular direction. For the convolution in the stripe direction seldom changes the stripe itself, and that in the perpendicular direction could be seen as convolving a square signal, the 1D profile of an ideal stripe structure, we can easily extend the 1D theory to a 2D version:

Proposition 4. Given a stripe structure of a width w , DoG responses in the centerline and its neighborhood reach extrema at scale $\sigma = \hat{\sigma}$.

The analysis so far has been carried out for a method to generate maps containing extremal responses without any directional calculation, which supplies the candidate stripe regions to reduce the computation cost and get accelerated as mentioned before. However, not only stripe responses but edge (Fig. 3d) and edge-side ones (Fig. 3a-c) exist in the same map, so we need to screen out the stripe responses.

2.2 Edge Responses Elimination

Applying DoG filter to an image, at a certain scale, not only stripes generate extremal responses, but also edges and edge sides. Moreover, edge responses appear as extrema even though the whole scale range, so it is necessary to separate the stripe responses from the edge ones.

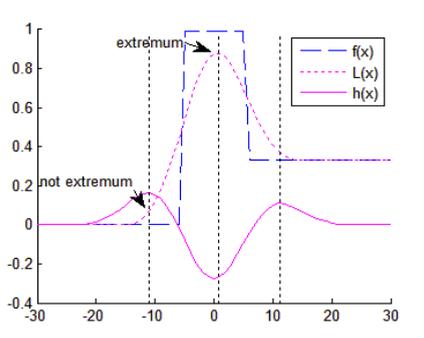


Fig. 4. The response profiles of the edge and stripe in 1D

We draw $f(x)$, $L(x; \sigma)$ and $h_{square}(x; \sigma)$ in Fig. 4 to illustrate their mutual relationships where

$$L(x; \sigma) = G(x; \sigma) * f(x) \tag{12}$$

In Fig. 4, when $h_{square}(x; \sigma)$ reaches local extremum on both sides, $L(x; \sigma)$ is still far from its peak, but when $h_{square}(x; \sigma)$ reaches extremum inside, $L(x; \sigma)$ reaches its peak nearby. This discovery constitutes the foundation of distinguishing the extremal responses of edges and stripes.

Now we will examine the peak location of $L(x; \sigma)$. There is a bias problem while stripe edges are asymmetry [4] that the peak of $L(x; \sigma)$ will shift from the extremum of $h_{square}(x; \sigma)$ when $a \neq b$, as also shown in Fig. 4. And the offset, i.e. the root of the first derivative of $L(x; \sigma)$ is

$$l = \frac{\sigma^2}{w} \ln\left(\frac{a}{b}\right) \tag{13}$$

If l belongs to the $h_{square}(x; \sigma)$ extremal interval, we could easily estimate the center by seeking the extremum of $L(x; \sigma)$ in that interval, corresponding to the candidate region in 2D. We name them salient stripes that satisfy this condition and have extremal responses beyond a certain threshold.

However, it is hard to determine the extremal interval analytically, so we compute a/b and its generated offset ratio $2l/w$ numerically to give some perceptual knowledge.

Table 2. a/b and its generated offset ratio when $k = 1.8$

a/b	40	6	3	2	1.5	1
$2l/w$	1.00	0.48	0.30	0.19	0.11	0

Table 2 shows that when $a/b \leq 3$, the extremum of $L(x; \sigma)$ is very close to that of $h_{square}(x; \sigma)$, i.e. it generates salient stripes. And the condition $a/b \leq 3$ is so loose that it could be satisfied in most common situations.

By shifting along the direction perpendicular to the candidate stripe regions and do non-maximum suppression, we can get the approximate centerlines of stripes.

An equivalent way to carry out the non-maximum suppression is to check zero-crossing. Assuming $l(x)$ indicates the value of $L(\mathbf{x}, \sigma)$ that x pixels away from \mathbf{x} , shifted along the unit direction \mathbf{e} , and \mathbf{x}_0 is the sample point, we get

$$\begin{aligned} l(x) &= L(\mathbf{x}_0 + x\mathbf{e}) \\ &\approx L(\mathbf{x}_0) + x \frac{\partial L(\mathbf{x}_0)^T}{\partial \mathbf{x}} \mathbf{e} + \frac{x^2}{2} \mathbf{e}^T \frac{\partial^2 L(\mathbf{x}_0)}{\partial \mathbf{x}^2} \mathbf{e} \end{aligned} \tag{14}$$

The location of the extremum, is estimated by taking the derivative of this function with respect to x and setting it to zero, giving

$$\hat{x} = (\mathbf{e}^T \frac{\partial^2 L(\mathbf{x}_0)}{\partial \mathbf{x}^2} \mathbf{e})^{-1} \frac{\partial L(\mathbf{x}_0)^T}{\partial \mathbf{x}} \mathbf{e} \quad (15)$$

Consequently, we can discriminate between stripe responses and edge ones by simply checking whether the estimated $\hat{x} \in [-\frac{1}{2}, \frac{1}{2}]$ meets, and the approximate centerline map is obtained.

2.3 Stripe Detection Scheme

Based on the stripe model and false response elimination algorithm, we begin to set up the whole stripe detection scheme.

First, we choose a target stripe width w and build the scale-space around the scale σ according to Eq. 9. The parameter k should be tuned according to different requirements, because it affects the precision of stripe width selection and determines whether the centerlines stay inside the candidate stripe regions along with a/b . A bigger k results in less selection precision and more stay-inside possibility. In general, a recommended value $k = 1.8$ could fulfill most common needs and we use it as default in Sec. 3.

Second, we build the scale-space and generate the DoG maps by subtracting neighbor Gaussian layers. The extremum maps that contain candidate stripe regions are generated by comparing pixel responses to its neighbor DoG maps and keeping the extrema above a certain screen threshold. We could set the screen threshold to select stripes with prominent edges in consideration of the edge contrast as used in [4]. E.g., if stripes with edge contrasts of $a, b \geq 70$ are to be selected, they will have extremal responses about -22.5 from Eq. 6. Therefore, the upper threshold for the absolute value of the response is set to 20. If intersection operation is needed, the lower threshold is set to 3.6 in order to collect adequate responses.

Finally, edge responses are eliminated by the method described in Sec 2.2, so that only approximate one-pixel-width centerlines remain. And we can also estimate stripe masks from centerlines with dilate operation according to the target stripe width.

The flowchart of the whole stripe detection scheme is shown in Fig. 5.

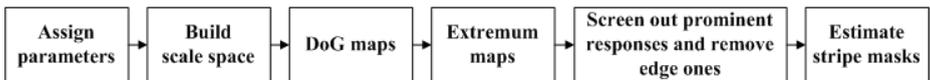


Fig. 5. The flowchart of the proposed scheme

In addition, if stripe is too wide that will add a heavy burden to Gaussian convolution due to the filtering window size, we can use Lowe's work [10] to generate image pyramid and calculate on a smaller image to save computing cost.

3 Experiments and Results

In this section, we compare the proposed method to Koller [3], Steger [4], and Epshtein [8], regarding to the quality, width selection and efficiency.

First, we compare the detection results in an aerial image as Fig. 6a. We find that Koller, Steger and the proposed methods generate similar results. However, the effect of Steger is a little worse due to its inability of width selection, so roads and other components with narrower width all emerge. In order to reduce these noises to the same level of Koller and the proposed for comparison, we have to tune screen threshold and some fractions of roads disappear. Epshtein is deeply trapped by those noises, and it is hard to distinguish roads from them without other cues.

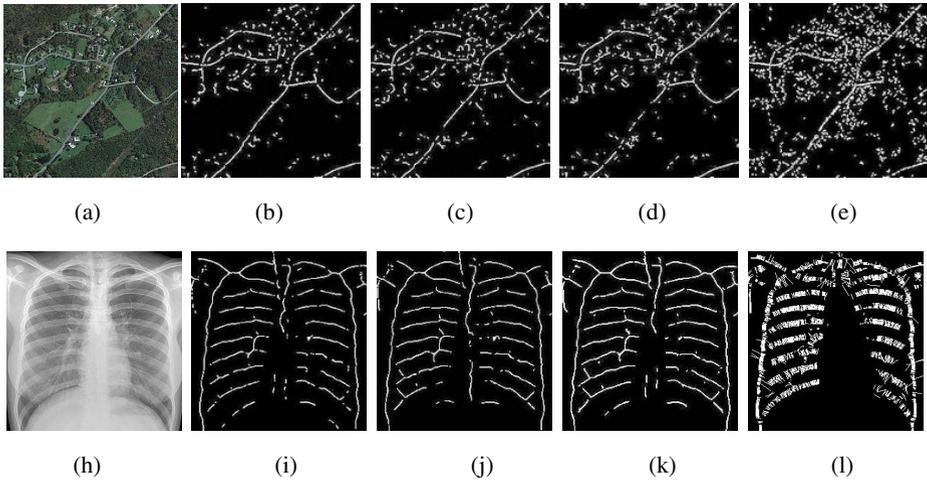


Fig. 6. Results from an aerial image (a) and a medical one (h) by the proposed (b, i), Koller (c, j), Steger (d, k) and Epshtein (e, l)

Table 3. The time cost (s) comparison on Fig. 6a, h

Method	Koller	Steger	Epshtein	Proposed
Fig. 6a	47.2	17.4	2.0	1.63
Fig. 6h	10.8	4.0	0.7	0.8

Fig. 6h gives another example on medical image. Due to the smoothness property of Gaussian kernel, Koller, Steger and the proposed method are applicable in blurred or low edge contrast condition. While Epshtein, depending on edge detection badly, misses some stripes.

Table 3 shows time cost of each method on original image of Fig. 6a with size 803×596 and Fig. 6h with size 365×300 . All methods are carried out in matlab on a PC with 2.4GHz CPU and 3GB memory. The proposed method runs fastest on Fig. 6a and the second fastest on Fig. 6h. The main reason lies in fewer pixels to calculate, for the proposed method only does calculation on pixels within candidate stripe regions

comparing to the traversal scheme used by Koller and Steger. Epshtein is also fast in both examples, and its efficiency is directly related to the number of edge pixels extracted by Canny filter.

In Fig. 7, another common task, scene text detection [12], is examined. As Koller and Steger generate similar results as the proposed, we only compare to Epshtein.

Epshtein aims at scene text detection by stroke transformation and rule based assembly. However, in complex background like trees and overlapped objects, the transformation catches a lot of noisy responses (Fig. 7c) or split the stroke into pieces (Fig. 7f) which changes the actual stroke width. Furthermore, it fails to accurately determine the width of crossing strokes when strokes are thick, for it is determined by the median width of a scanning ray. E.g., in Fig. 7f, left strokes of the letter “D” and “R” in the first row both get screened for incorrectly width determination. While the proposed method uses width selection ability to shield the effects of trees and overlapped objects and gains a better result (Fig. 7b, e).

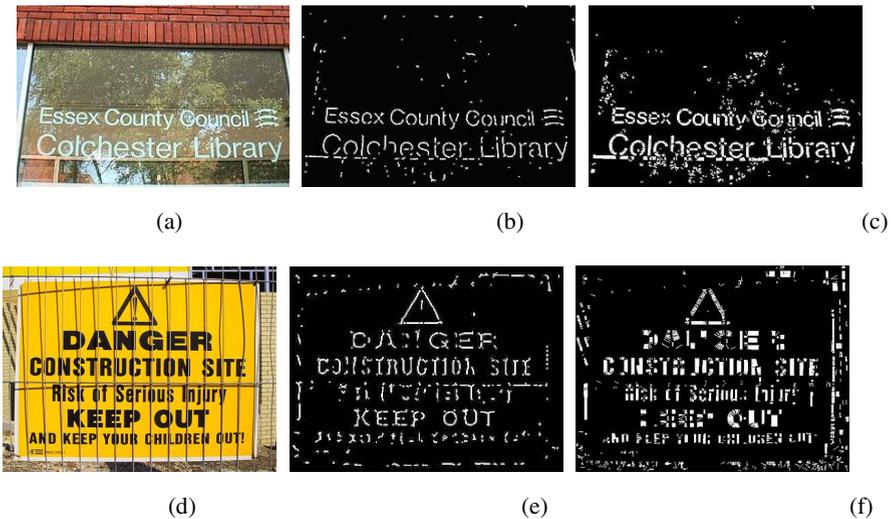


Fig. 7. Comparing results of scene text image (a, d) between the proposed (b, e) and Epshtein (c, f). For we aim at distinguishing thick stripes from the thin balustrade in (d), those thin stroke characters also get screened in (e, f).

4 Conclusion

In this paper, we studied current methods which detect multiple forms of stripes, and introduce a mathematical model to define the common stripe structures by exploiting the characteristics of the edge DoG response. Then, based on the model, we present a scale-space based method to estimate the stripe centerlines and masks efficiently. Due to the peculiar property of the scale-space and the scale specification of extremal maps, our method extracts candidate stripe regions to get accelerated and can detect stripes of diverse forms. Furthermore, we show the method works well on practical images comparing to other popular methods in experiments.

Our method only gives approximate results of centerlines and stripe masks concentrating on detection efficiency. If more precise stripe boundary and bias removal are needed, detailed analysis of asymmetrical stripe profiles [4] may be helpful, and could be integrated into our scheme.

In the future, we will integrate GPU based parallel computing scheme [13, 14] to make the method processing much faster.

Acknowledgements. This work is supported by National Nature Science Foundation of China (61273247, 61271428), National Key Technology Research and Development Program of China (2012BAH39B02), and Co-building Program of Beijing Municipal Education.

References

1. Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd edn., pp. 719–722. Prentice-Hall Publisher (2008)
2. Hough, P.V.C.: Method and means for recognizing complex patterns. U.S. Patent 3,069,654 (December 18, 1962)
3. Koller, T.M., Gerig, G., Szekely, G., Dettwiler, D.: Multiscale detection of curvilinear structures in 2-D and 3-D image data. In: Proc. ICCV, pp. 864–869 (1995)
4. Steger, C.: An unbiased detector of curvilinear structures. IEEE Trans. PAMI 20(2), 113–125 (1998)
5. Eberly, D., Gardner, R., Morse, B., Pizer, S., Scharlach, C.: Ridges for image analysis. J. Math. Imaging Vis. 4(4), 353–373 (1994)
6. Koenderink, J.J., van Doorn, A.J.: Two-plus-one-dimensional differential geometry. Pattern Recognition Letters 15(5), 439–444 (1995)
7. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. In: Proc. CVPR, pp. 465–470 (1996)
8. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proc. CVPR, pp. 2963–2970 (2010)
9. Lindeberg, T.: Scale-space theory: a basic tool for analyzing structures at different scales. J. Appl. Stat. 21(2), 224–270 (1994)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60(2), 91–110 (2004)
11. Lindeberg, T.: Generalized Gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space. J. Math. Imaging Vis. 40(1), 36–81 (2011)
12. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: reading text in scene images. In: Proc. ICDAR, pp. 1491–1496 (2011)
13. Wang, W.Y., Zhang, D.M., Zhang, Y.D., Li, J.T., Gu, X.G.: Robust spatial matching for object retrieval and its parallel implementation on GPU. IEEE Trans. on Multimedia 13(6), 1308–1318 (2011)
14. Xie, H.T., Gao, K., Zhang, Y.D., Tang, S., Li, J.T., Liu, Y.Z.: Efficient feature detection and effective post-verification for large scale near-duplicate image search. IEEE Trans. on Multimedia 13(6), 1319–1332 (2011)