



Fast curvilinear structure extraction and delineation using density estimation

Shu-Xiao Li^{*}, Hong-Xing Chang, Cheng-Fei Zhu

Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100190, China

ARTICLE INFO

Article history:

Received 6 March 2008

Accepted 21 January 2009

Available online 31 January 2009

Keywords:

Line detection

Density estimation

LWF

Small object detector

Centerline detector

ABSTRACT

Detection and delineation of lines is important for many applications. However, most of the existing algorithms have the shortcoming of high computational cost and can not meet the on-board real-time processing requirement. This paper presents a novel method for curvilinear structure extraction and delineation by using kernel-based density estimation. The method is based on efficient calculation of pixel-wise density estimation for an input feature image, which is termed as local weighted features (LWF). For gray and binary images, the LWF can be efficiently calculated by integral image and accumulated image, respectively. Detectors for small objects and centerlines based on LWF are developed and the selection of density estimation kernels is also illustrated. The algorithm is very fast and achieves 50 *fps* on a PIV2.4G processor. Evaluation results on a number of images and videos are given to demonstrate the satisfactory performances of the proposed method with its high stability and adaptability.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Detection and delineation of lines, also called curvilinear structures, is important for many applications, such as biometric trait detection for personal authentication, human computer interaction, and road detection in aerial images for UAV (Unmanned Aerial Vehicles) navigation. So far, many line detection algorithms have been developed for various applications, which can be categorized into three groups based on the characteristics of detection results: (1) edge based method; (2) centerline based method and (3) region based method. *Edge based method* is based on edge extraction and treats lines as objects with parallel edges [1–4]. An edge extraction algorithm is usually used to get the edge image, which is then analyzed to find the particular lines. The main weakness of this method lies in the fact that the line thickness is not considered. As a result, the detection result only contains edges, not the required regions or centerlines of curvilinear structures. To overcome this problem, Koller et al. [5] presented an edge based line finder for extracting curvilinear structures as well as their widths by iterating in scale space. It selects the line width as the scale that yields the maximum of a scale normalized response. However, due to the iterative scale space analysis, this approach is too computationally expensive for real-time applications.

Centerline based method intends to get the delineation of curvilinear structures, which mainly includes three aspects: (1) ridge detection; (2) region thinning and (3) line delineation. *Ridge detection* uses differential geometric properties to extract centerline image as ridges and valleys in the input gray image [6–7]. Eberly et al.

[8] proposed a method to extract the ridges as points for which the intensities are maxima or minima in the direction of the maximum eigenvalue of the Hessian matrix. Nevertheless, due to the use of second order of derivatives, this approach is sensitive to noise. A DoG (Difference of Gaussian) operator which is originally used in SIFT (Scale Invariant Feature Transform) feature detector [9] can solve this problem to some extent, but we leave it for further investigation because the DoG operator cannot meet our on-board processing requirement either. Aggarwal et al. [10] employed regularized Hough transform to determine the location and orientation of straight lines in gray images. This method combines all the three aspects to one stage and this makes it easy to use in practices. The main disadvantage is the high computational cost and its restriction on straight lines.

Region thinning utilizes thinning techniques to get one-pixel-wide centerlines from the input region image of curvilinear structures. A classical approach is based on edge distance transform map by using iterative morphological analysis [11] and non-maximum suppression technique. The iteration times depend on the maximum line width in the region image and this makes it relatively slow. Moreover, this approach is also sensitive to noise. So a closing operation is usually executed to remove holes in the pre-processing step, which further increases the computational cost. Zhang and Couloigner [12] presented a method to detect accurate centerlines and estimate line widths using the Radon transform. They proposed a mean filter to locate the true peak in the Radon image and a profile analysis technique to refine the line parameters. This approach combines the later two aspects, which makes it convenient to use. However, the method can only deal with straight lines and is not fast enough for real-time applications.

^{*} Corresponding author.

E-mail addresses: shuxiao.li@ia.ac.cn, shower140@163.com (S.-X. Li).

Line delineation refers to get line parameters from the input centerline image. The Hough transform [13–15] and the RANSAC (Random Sample Consensus) theory [16–17] are good tools for defined curvilinear structure parameter estimation in discrete centerline images. Ballard [18] proposed the generalized Hough transform to detect arbitrary lines in the centerline image, but it requires a complete specification of the exact line shape which is very difficult and even unfeasible to get for complex curvilinear structures in practices. For continuous centerline image, incremental line fitting by TLS (Total Least Squares) [17] or by ADSS (Approximate Digital Straight line Segments) [19], which is fast and easy to be implemented, can be employed to get the vector representations of arbitrary lines. Once the continuous centerline image is available, these algorithms work well and can meet most requirements in practical applications.

Region based method aims at extracting all the pixels of curvilinear structures which correspond to salient regions from the input gray image. Wavelet transform [20], morphological top-hat and bottom-hat operators [21] are good tools for high frequency component extraction. Liu et al. [22] presented a wide line detector based on the isotropic responses via circular masks using the idea of the SUSAN (Smallest Uni-value Segment Assimilating Nucleus) corner detector [23], which they call INF (Isotropic Nonlinear Filtering). This approach is robust against noise because the detection is not based on derivatives of images. It also works well for a range of images containing lines of different widths, especially when the width of lines varies greatly. However, this algorithm is slow since it involves large mask comparison operations for every image pixel. Background estimation from a single image is another tool for detecting salient regions through background subtraction. Chao et al. [24] proposed an anisotropic diffusion scheme to detect defects in low-contrast surface images. This approach can simultaneously carry out the smoothing and sharpening operations so that a simple threshold can be used to segment the intensified defects in the result image. The main shortcoming of this algorithm is the high computational cost due to the iterative filtering scheme.

In this paper, we present a fast line detection and delineation method which is suitable for on-board real-time processing. The real-time performance is due to using the idea of fast kernel-based density estimation, which is termed as LWF (Local Weighted Features). Fundamentals of LWF including definition and calculation methods are discussed extensively. The proposed method subsequently extracts the whole region pixels, centerlines and parameters of curvilinear structures from input images. It mainly consists of two components: small object detector and centerline detector. Our small object detector is implemented by employing LWF based background estimation techniques followed by a noise removal scheme. We also design a performance evaluation criterion of estimated background image for selecting LWF kernel. The proposed centerline detector utilizes LWF based evidence map estimation method, followed by non-maximum suppression and incremental line fitting to get one-pixel wide centerlines and line parameters, respectively. The estimated line parameters, as well as centerlines and line regions, are all useful in many applications. Compared to existing line detectors, the LWF based method is faster and more robust. In summary, our main contributions comprise the following:

- Kernel-based density estimation is first introduced to line detection and accumulated image is developed to efficiently compute the LWF for binary images;
- Detectors for small objects and centerlines based on LWF are proposed and the selection of density estimation kernels is discussed;
- The fast line detection and delineation method for on-board real-time visual applications takes shape since the LWF with selected kernels can be efficiently computed by using the tools of integral image and accumulated image, respectively.

This paper is outlined as follows. We give a thorough discuss of the definition and calculation methods for LWF in Section 2. The selection of the LWF kernel for background estimation and the LWF based small object detector are described in Section 3. Section 4 presents the centerline extraction and delineation method based on LWF and incremental line fitting. In Section 5, detection results and evaluations are presented. Conclusions are summarized in Section 6.

2. Fundamentals of LWF

Let $I(\mathbf{x})$ denote the input image where \mathbf{x} is the 2-D pixel position. The problem for object detection can be viewed as a background estimation problem:

$$I(\mathbf{x}) = I_b(\mathbf{x}) + I_o(\mathbf{x}) + I_n(\mathbf{x}) \quad (1)$$

where $I_b(\mathbf{x})$, $I_o(\mathbf{x})$ and $I_n(\mathbf{x})$ are background image, object image, and noise image, respectively, and the goal is to estimate $I_o(\mathbf{x})$ from the single input image $I(\mathbf{x})$ which can be viewed as a classical density estimation problem in 2-D image space.

Another problem that we usually confront is object centerline extraction from the input region image, in which the central task is to estimate the evidence map. The value of evidence map at each pixel corresponds to the “distance” from the edges, indicating that the ridges of it are the centerlines. The estimation of evidence map is actually a density estimation problem too.

In this section, we will discuss the definition and applicable calculation methods of a special density estimation problem, which is termed as LWF. This is the basic for the proposed small object detector and centerline detector. It can also provide guidelines for applying LWF to other applications.

2.1. Definition of LWF

LWF uses feature values of neighborhood pixels to estimate the density of the current pixel, which can be expressed as:

$$I_{LWF}(\mathbf{x}_c) = \sum_{i=1}^{N_h} I(\mathbf{x}_c^i) k\left(\left\|\frac{\mathbf{x}_c^i - \mathbf{x}_c}{h}\right\|\right) \quad (2)$$

where $I_{LWF}(\mathbf{x}_c)$ is the estimated LWF value of the current pixel \mathbf{x}_c , h is the bandwidth (a value of 10 means a 21×21 window), N_h is the pixel numbers in bandwidth h , and $k(x)$ is a profile function [25] to generate the density estimation template. The “local”, “weighted” and “features” correspond to bandwidth h , profile function $k(x)$ and the input feature images $I(\mathbf{x})$, respectively. Similar expressions have been applied in image segmentation [26] and object tracking [27] in a gradient decent manner, but the definition and application to small object detection and centerline extraction in an exhaustive manner are original. Additionally, we give a thorough discussion on the calculation methods of LWF. Three candidate profile functions we may use frequently are:

- CPF (Constant Profile Function)

$$k_c(x) = \begin{cases} 1, & x \leq 1 + \varepsilon - x\varepsilon \\ 0, & x > 1 + \varepsilon \end{cases} \quad (3)$$

- EPF (Epanechinikov Profile Function)

$$k_e(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-x), & x \leq 1 \\ 0, & x > 1 \end{cases} \quad (4)$$

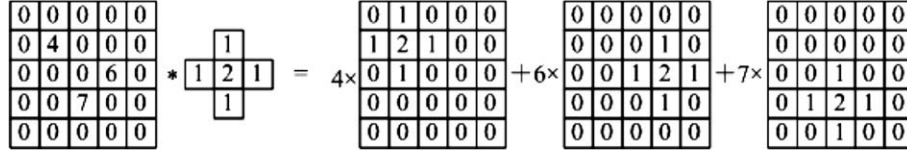


Fig. 1. Illustration of accumulated image for LWF.

- GPF (Gauss Profile Function)

$$k_g(x) = \begin{cases} c_d e^{-\alpha x^2}, & x \leq 1 \\ c_d e^{-\alpha \frac{1+\varepsilon-x}{\varepsilon}}, & 1 < x \leq 1 + \varepsilon \\ 0, & x > 1 + \varepsilon \end{cases} \quad (5)$$

where d , c_d , and c_α are normalization constants. α is the cut coefficient. Cut values of 1% and 10% correspond to values of 4.6 and 2.3 for α , respectively, and we set α be 2.3 in this paper. Since a profile function should keep continuous, we add a term containing ε which is a sufficient small positive number. However, this term actually does not influence the calculation process in image domain.

2.2. Calculation methods

A novel approach for calculating LWF which is termed as accumulated image and is especially useful for binary images is exploited in this subsection. For completeness, we also summarize the existing methods such as 2-D template, sequential 1-D template, and integral image below.

- 2-D Template

2-D template is the direct calculation for LWF. Let c_a and c_m be the cost of an addition and a multiplication, respectively. The cost with EPF or GPF of LWF using 2-D template is appropriately $(2h+1)^2 (c_a + c_m)$, while that with CPF is appropriately $(2h+1)^2 c_a$.

- Sequential 1-D Template

Sequential 1-D template is feasible due to the use of profile functions which makes 2-D convolutions separable. The cost with EPF or GPF of LWF using sequential 1-D template is appropriately $2(2h+1) (c_a + c_m)$, while that with CPF is appropriately $2(2h+1) c_a$.

- Integral Image

The concept of integral image is proposed by Viola and Jones [28] and has been widely used in rectangle feature extraction [28–29]. It can efficiently sum the values over a rectangle region. Integral image is defined as:

$$I_\Sigma(x, y) = \sum_{i \leq x} \sum_{j \leq y} I(i, j) \quad (6)$$

and can be computed iteratively:

$$I_\Sigma(x, y) = I(x, y) + I_\Sigma(x-1, y) + I_\Sigma(x, y-1) - I_\Sigma(x-1, y-1) \quad (7)$$

Then, LWF with CPF can be calculated iteratively:

$$I_{LWF}^{CPF}(x, y) = \frac{1}{N_h} \left[I_\Sigma(x_1, y_1) + I_\Sigma(x_0-1, y_0-1) - I_\Sigma(x_0-1, y_1) - I_\Sigma(x_1, y_0-1) \right] \quad (8)$$

where (x_0, y_0) and (x_1, y_1) are the 2-D positions of the rectangle's top-left and bottom-right corners, respectively. The cost of LWF with CPF by using integral image is appropriately $8c_a + c_m$.

Since EPF can be expressed as sequential convolutions of CPF, the cost with EPF of LWF using integral image is appropriately $16c_a + c_m$. However, this method is unsuitable for GPF.

- Accumulated Image

The above three methods get LWF values pixel by pixel except for the acquirement of integral image which involves iterative process. Here, we introduce a novel method for computing LWF employing totally batch process. This method is based on the concept of accumulated image and can be expressed as:

$$I_{LWF} = \sum_{i=1}^N I(\mathbf{x}_i) I_i \quad (9)$$

where I_i is the image generated by the template placed at \mathbf{x}_i . Fig. 1 shows an example of the calculation process for this method. It adds the generated image I_i at each non-zero pixel \mathbf{x}_i to an image termed as accumulated image. Let k be the ratio of non-zero pixel numbers to the total pixel numbers in the image. The cost with EPF or GPF of LWF using accumulated image for a gray image is appropriately $k(2h+1)^2 (c_a + c_m)$, while that with CPF is appropriately $k(2h+1)^2 c_a$. For binary images, this method needs no multiplications and the cost for all profile functions is $k(2h+1)^2 c_a$. Since the value of k is usually sufficiently small for binary images (e.g. 0.05) and is about 1 for gray images, this method is especially suitable for calculating LWF of binary images.

Table 1 summarizes the costs for calculating LWF with CPF, EPF and GPF. The computing strategy can be generated from the comparative results directly. However, the actual costs are also influenced by some other factors such as data type, truncation error and the frequency of reading and writing datum. In many applications, we are more concern about the consumption time of the algorithm with specific profile functions and parameters. The above algorithms have been implemented in C++ code with optimization and tested on a PIV 2.4G processor. The test image size is 640×480 . Some useful conclusions are given below.

2.3. Calculation strategy for gray images

Since the computational time of LWF for gray images using 2-D template and accumulated image is comparable as shown in table

Table 1
Costs for different methods and profile functions.

	CPF	EPF	GPF
2-D Template	$(2h+1)^2 c_a$	$(2h+1)^2 (c_a + c_m)$	$(2h+1)^2 (c_a + c_m)$
Sequential 1-D template	$2(2h+1) c_a$	$2(2h+1) (c_a + c_m)$	$2(2h+1) (c_a + c_m)$
Integral image	$8c_a + c_m$	$16c_a + c_m$	unsuitable
Accumulated image for gray images	$k(2h+1)^2 c_a$	$k(2h+1)^2 (c_a + c_m)$	$k(2h+1)^2 (c_a + c_m)$
Accumulated image for binary images	$k(2h+1)^2 c_a$	$k(2h+1)^2 c_a$	$k(2h+1)^2 c_a$

1, we only present the test results using 2-D template, sequential 1-D template and integral image for different profile functions as shown in Fig. 2(a–c). Though there are some unavoidable disturbances in the test environment, the increase rates of consumption time with respect to bandwidth h for these methods are in general of order 2, 1, and 0, respectively, which are consistent with the theoretical analysis results in the above subsection. In addition, the curve of CPF by using integral image consists of two constant parts as shown in Fig. 2(c). This is due to using different data types and substitution strategies for division operator during the optimization process with respect to different values of h . For a fixed value of h , the minimum cost called hybrid cost is selected to generate the calculation strategy for gray images as shown in Fig. 2(d). Some useful conclusions drawn from Fig. 2 are:

1. For the same value of h by using 2-D template or sequential 1-D template, the calculation time with CPF is smaller than that with EPF or GPF, while the cost with EPF is comparable to that with GPF.
2. The calculation time with CPF by means of integral image is about 6 ms when h is smaller than 10 and is about 9 ms when h is larger than 10. For EPF, the bound value for h is 20 and the costs are 12 ms and 18 ms, respectively.
3. The costs by using 2-D template and sequential 1-D template are unbearable for real-time applications when the values of h are larger than 2 and 6, respectively, while integral image based method is applicable for arbitrary values of h .
4. The hybrid method with CPF gets LWF utilizing 2-D template when h equals to 1 and integral image when h is larger than 1.
5. The hybrid method with EPF gets LWF by using 2-D template when h equals to 1 and sequential 1-D template when h takes the value of 2 or 3. When the value of h is larger than 3, integral image based method is employed.
6. The hybrid method with GPF adopts 2-D template when h equals to 1 and uses sequential 1-D template when h is larger than 1.
7. For applications when the selected value of h is smaller than 4, the profile function which gets the best performance can be employed. But for larger values of h , CPF and EPF are more preferred.

2.4. Calculation strategy for binary images

The computational time of LWF for binary images, which are special cases of gray images, obeys the hybrid cost as shown in Fig. 2(d) in the worst cases. However, accumulated image based method, which is very fast with a small value of k , may be more efficient. Fig. 3(a and b) show the calculation time with different profile functions when k equals to 0.01 and 0.03, respectively. The cost with CPF is only a bit smaller than that with EPF or GPF, so we only present the average costs by means of accumulated image with different values of k as illustrated in Fig. 3(c). Only typical values of k are examined because they usually lie below 0.1 and in most cases are around 0.01. For a fixed value of h , the minimum cost between accumulated image based method and that computed as gray images, which is called hybrid cost, is selected to generate the calculation strategy for binary images as shown in Fig. 3(d–f). Some useful conclusions drawn from Fig. 3 are:

1. The cost of LWF by means of accumulated image is applicable when k and h are smaller than 0.03 and 10, respectively, which occurs rather frequently in practices. This method can also meet the real-time requirement when the values of k and h are below 0.09 and 7, respectively.
2. The hybrid method with GPF gets the LWF by means of accumulated image for all values of k and h that are smaller than 0.09 and 10, respectively.
3. The hybrid method with EPF adopts accumulated image when k and h are smaller than 0.03 and 10, respectively. When values of k are larger than 0.05, accumulated image is utilized for small values of h , followed by integral image based method for larger values of h .
4. The hybrid method with CPF calculates LWF using accumulated image when k and h are smaller than 0.01 and 10, respectively. When values of k are larger than 0.03, accumulated image is utilized for small values of h , followed by integral image based method for larger values of h .
5. For applications when the values of k and h are smaller than 0.05 and 10, respectively, the profile function which gets the

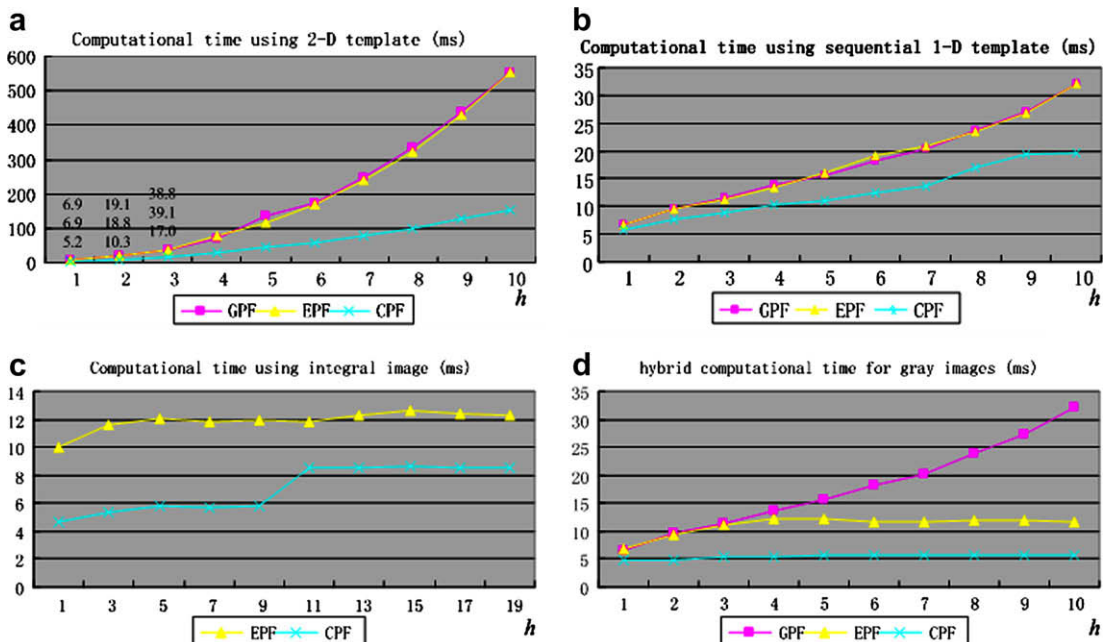


Fig. 2. Calculation time of LWF for gray images with GPF, EPF and CPF by means of (a) 2-D template, (b) sequential 1-D template and (c) integral image, respectively, on a PIV 2.4G processor. The resolution of test images is 640 by 480. The hybrid cost for gray images is shown in (d).

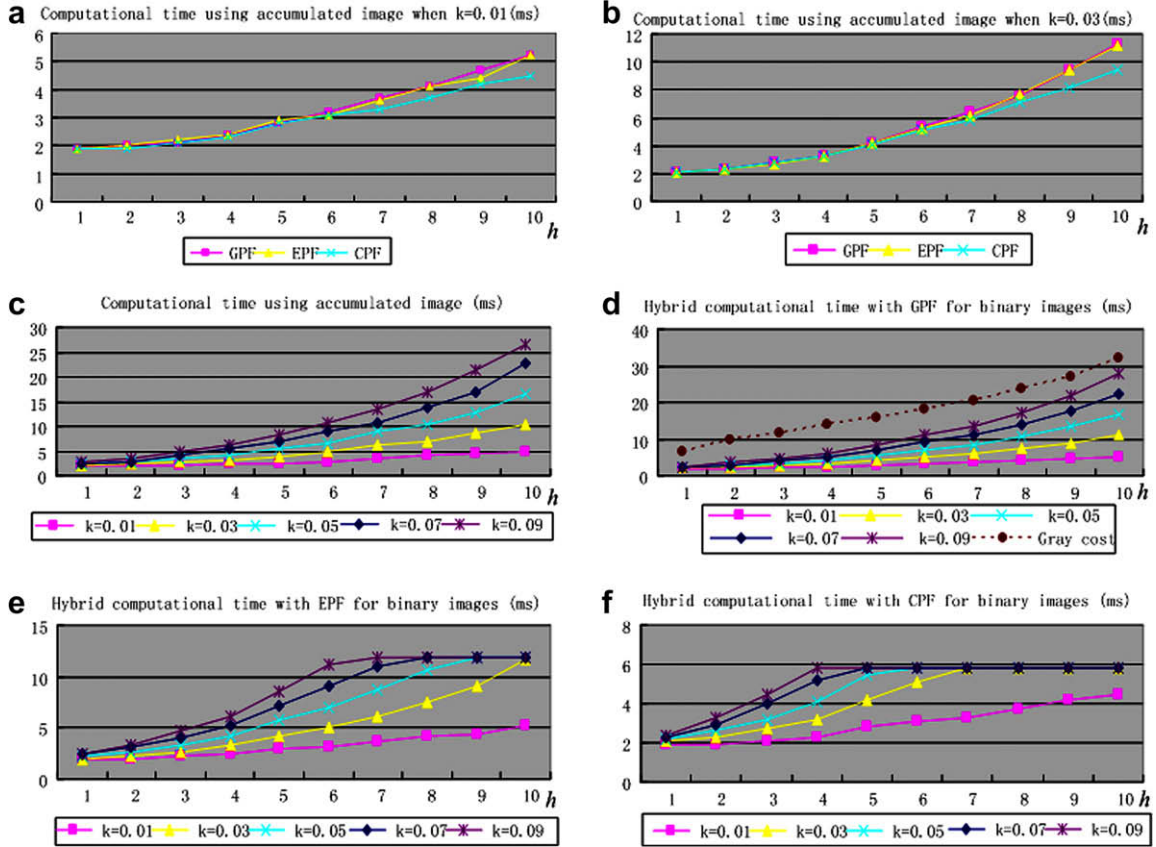


Fig. 3. Calculation time of LWF for binary images with GPF, EPF and CPF by means of accumulated image when (a) $k=0.01$ and (b) $k=0.03$, respectively, on a PIV 2.4G processor. The test image size is 640 by 480. (c) Shows the average computational time by using accumulated image for different values of k . (d)–(f) are the hybrid computational time for binary images with GPF, EPF and CPF, respectively.

best performance can be selected. But for larger values of k and h , CPF and EPF are more preferred.

3. LWF based small object detector

In this section, we first propose a criterion to determine background estimation performance by using LWF with different profile functions and then describe details of the proposed small object detector.

3.1. Profile function selection

The quality of estimated background image, which is critical to object detection, should reflect large differences in object areas and similar values in background regions with respect to the input image. This can be characterized by:

$$C_d(h) = \frac{\sum_{i=1}^{N_o} |I_{LWT}(\mathbf{x}_{i,o}) - I(\mathbf{x}_{i,o})|}{N_o} - \frac{\sum_{i=1}^{N_b} |I_{LWT}(\mathbf{x}_{i,b}) - I(\mathbf{x}_{i,b})|}{N_{bc}(h)} \quad (10)$$

where $\mathbf{x}_{i,o}$ and N_o are pixel locations in object regions and the number of pixels in them, $\mathbf{x}_{i,b}$ and N_b are pixel coordinates in background areas and the corresponding number of pixels, $N_{bc}(h)$ is the number of pixels surrounding object regions in background areas within bandwidth h , and $C_d(h)$ is the quality value of the estimated background image. The first term at the right hand of Eq. (10) reflects the differences between estimated background image and input image in object regions, and the second term represents the similarity in background areas surrounding object regions within bandwidth h . In addition, this criterion is normalized with respect to object

size, valid background size and bandwidth of profile functions, and so it's appropriate for characterizing the quality of estimated background images.

Fig. 4 shows the quality measure value $C_d(h)$ of the estimated background image acquired by LWF for different profile functions with respect to bandwidth h . The test image contains a curvilinear object whose width is about 5 pixels, indicating that the object bandwidth h_g equals to 2. The brightness value of the object is 255 and that of background is 0 in the test image. The local disturbances in the quality curves in Fig. 4 are due to data truncations, data types and optimization strategies in implementation of LWF. It can be indicated that the quality measure can achieve a considerable value of about 150 when the bandwidths for calculating LWF are approximately above $4h_g$, $6h_g$ and $7h_g$ for CPF, EPF and GPF, respectively. In practices, the quality measures are usually

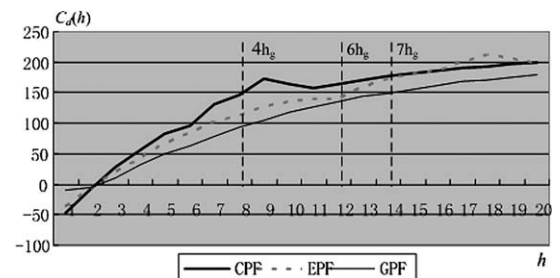


Fig. 4. Quality analysis of estimated background images with respect to bandwidth h for profile function selection.

not monotonously incremental with respect to bandwidth h because of the disturbances in background areas and nearby object regions. A larger value of bandwidth h indicates that the quality of the estimated background image is more dependent on the surrounding circumstances. This implies that LWF using CPF with the bandwidth of about twice the maximal curvilinear structure widths can get the best performance for background estimation among these three profile functions. This conclusion coincides with the experience that flat profile functions can smooth out objects better and thus can estimate the background image with higher quality compared to sharp ones. In addition, LWF with CPF is more efficient than those with EPF and GPF as described in Section 2, and so is most appropriate for background image estimation and object detection tasks.

3.2. Proposed small object detector

Small objects in this paper refer to targets with at least one small scale in image space, which mainly include point features and curvilinear structures. Detecting small objects effectively and efficiently is very important for motion estimation, point feature extraction, personal authentication and UAV navigation. In this section, a novel small object detector is proposed. It consists of the following two steps:

3.2.1. Background subtraction

Denote the estimated background image as $I_{LWF}(\mathbf{x})$, which can be efficiently computed by using LWF with CPF of bandwidth h_o . The default value of h_o is 20 in this paper, which means that we are only interested in objects whose widths are less than 10 pixels. Once the background image is available, the candidate object regions can be acquired by the following simple manipulations:

$$I_b(\mathbf{x}_i) = \begin{cases} 1 & , I(\mathbf{x}_i) > I_{LWF}(\mathbf{x}_i) + T \\ 0 & , \text{otherwise} \end{cases} \quad (11)$$

$$I_d(\mathbf{x}_i) = \begin{cases} 1 & , I(\mathbf{x}_i) < I_{LWF}(\mathbf{x}_i) - T \\ 0 & , \text{otherwise} \end{cases} \quad (12)$$

$$I_o(\mathbf{x}_i) = I_b(\mathbf{x}_i) + I_d(\mathbf{x}_i) \quad (13)$$

where $I_b(\mathbf{x})$ or $I_d(\mathbf{x})$ is candidate object regions that are brighter or darker than surroundings, and $I_o(\mathbf{x})$ represents points that are dissimilar from the estimated background values. T is the segmentation threshold, which can be determined by the standard deviation of input images as is used in reference [22]. However, a constant threshold is employed more often in practices because it can meet the needs for most applications and can reduce the computational cost at the same time. A default value of 15 for T is competent for all the test images and sequences used in this paper.

3.2.2. Noise removal

The detection results acquired by background subtraction scheme include point features, curvilinear structures, wide borders of large objects and background regions, and some other noises. Point features and curvilinear structures, which are termed as small objects in this paper, are the intention of the detector. So a noise removal process must be put forward to eliminate the wide borders generated by large objects and background areas. Circular analysis, which has been successfully used in LBP (Local Binary Patterns) based texture analysis [30–31] and FAST (Features from Accelerated Segment Test) corner extraction [32–33], is adopted and extended to remove the wide borders in the candidate object or background regions. This noise removal strategy is based on the fact that pixels on the test circle in background areas or large object regions contain more points which have similar brightness values to the nucleus, while pixels in small object regions have

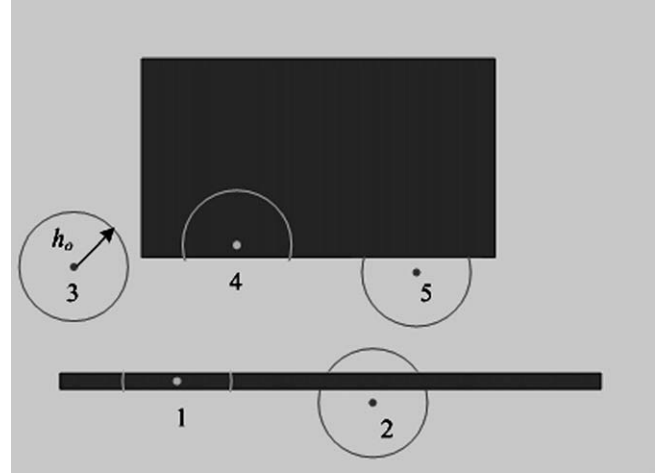


Fig. 5. Circular analysis for noise removal is illustrated. Five points and their circular parts which have similar brightness values to the nucleus are shown.

fewer ones as shown in Fig. 5. For efficiency, a number of N pixels equally spaced on the test circle with radius h_o are selected to test the validity of the acquired candidate pixels. The default value for N is 8 in this paper. A value of N above 8 might slightly improve the performance with the expense of increasing computational cost. Meanwhile, the number of pixels among the N test locations whose intensities differ from that of the nucleus by less than threshold T , denoted as n , is recorded. Finally, candidate pixels which does not satisfy $n < 0.3 N$ are considered as noises and are removed from the candidate regions.

Fig. 6 shows a typical process of the proposed small object detector. The input image in Fig. 6(a) contains curvilinear structures with different widths and brightness, point features, and a larger object on a non-uniform background. To detect the small objects in the original image, background image must be estimated firstly. The estimated background image by using LWF with CPF is presented in Fig. 6(b). Fig. 6(c) displays the detected candidate object regions through background subtraction, which contains wide borders of the large object and certain background regions near it. A noise removal strategy by using the idea of circular analysis is then employed to discard these noises and the final detection result is shown in Fig. 6(d). It can be seen that the proposed small object detector can extract the interested object regions with high precision.

4. LWF based centerline detector

Delineation of curvilinear structures efficiently and effectively is also an important task for many applications. Take UAV navigation using vision and GIS (Geographical Information System) as an example. Lines such as roads and rivers are represented as vectors in GIS. As a result, vector representations for curvilinear structures are expected to be extracted from the input image to achieve convenient and successful matches for UAV navigation. The key difficulty to delineate curvilinear structures from binary images is how to get the evidence map efficiently. In this section, a novel fast centerline detector is presented, which consists of the following three steps:

4.1. A. Evidence map acquisition

The evidence map for centerline detector, which should have large values on centerlines and gradually decreases along with

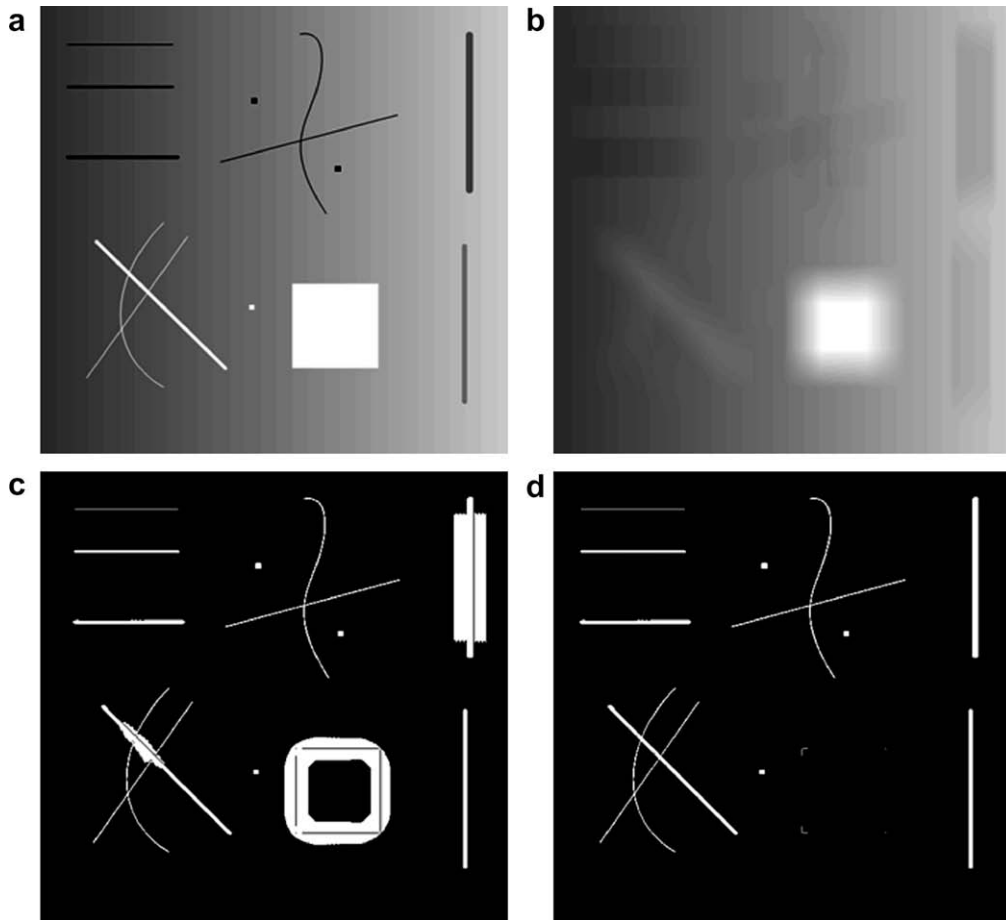


Fig. 6. Illustration of detection process of the proposed small object detector: (a) input image; (b) estimated background image by using LWF with CPF; (c) detected candidate regions through background subtraction; (d) detection result after noise removal.

the distance between the current point and its nearest centerline, can be efficiently estimated by means of LWF with GPF of bandwidth h_c , which takes half value of maximum interested line width. The default value of h_c is 5 in this paper according to the default setting of h_0 . For binary images, the value of k is usually very small and this operator takes less than 5 ms to estimate evidence maps from line region images sized of 640×480 in most cases.

4.2. B. Centerline generation

Centerlines of curvilinear structures correspond to ridges in the evidence map. The centerline extraction algorithm, which is similar to the idea of Canny edge detector [34], mainly consists of two steps: seed generation and line following. *Seed generation* utilizes non-maximum suppression techniques to acquire candidate seed pixels of centerlines. Pixels whose evidence values are above threshold T_1 and are not smaller than those of their 4-neighbors are selected as seeds. The default value of T_1 is 60 in this paper to avoid the influence of noisy pixels and background regions. For lines with even widths, two-pixel width seed sets may be acquired and a further suppression strategy has to be employed. For current seed which has more than one candidate seed in its 4-neighbors, we only keep the seed pixel which is furthest from the head of this seed sequence. Through this step, seed image which consists of isolated seeds and 4-neighborhood connected seed sequences is available.

Line following refers to acquire the 4-neighborhood connected centerlines from seed image and evidence map. A centerline be-

gins with a seed pixel which has not been processed. The 4-neighborhood pixel, which is not on the centerline and has the largest value in the evidence map among these neighbors, is added to the centerline gradually. For candidates with same evidence values, the candidate which is further from the head of the centerline is selected. This process ends until the evidence value of the added pixel is below threshold T_2 . Our experiments on several video clips show that a value between 1 and 25 for T_2 works well and a smaller value results in more weak lines. This parameter can be specified by users and we set it be 20 to get relatively strong lines in this paper. Once a candidate centerline takes shape, seeds with a distance less than 2 pixels from the centerline are suppressed, and then a new centerline is expected until there is no candidate seed to be processed. Finally, we discard candidate centerlines which not only contain less than 10 seeds but also have a length of less than 50 pixels because they usually correspond to noises or point features.

4.3. C. Incremental line fitting

Vector representations of the available 4-neighborhood connected centerlines, which are directly utilized in many applications, can be estimated by incremental line fitting efficiently. For sequential points $\{(x_i, y_i), i = 0, 1, 2, \dots, t\}$ on part of a centerline, they can be represented by a single vector pointing from (x_0, y_0) to (x_t, y_t) if both the absolute mean error e and the maximum error e_{\max} satisfy:

$$e < E, \quad e_{\max} < E_{\max} \quad (14)$$

$$e = \frac{1}{t+1} \left| \sum_{i=0}^t (ax_i + by_i + c) \right| \quad (15)$$

$$e_{\max} = \max\{|ax_i + by_i + c|, i = 0, 1, \dots, t\} \quad (16)$$

where a , b , and c are the normalized parameters of the vector, E and E_{\max} are employed to control the average and maximum distance apart from the current fitted line, respectively. Smaller values of E and E_{\max} will result in more line vectors to represent the details while the macrostructure becomes lost at the same time. They are user specified constants and we set them be 1.6 and 3 in this paper. The sequential point for each initialized vector on the centerline is gradually added to the vector if the relation (14) holds. The process continues until the added point breaks down the rule in (14) and then a new vector is generated from the end point of the usher vector. Since this operation runs only on centerline points the size of which is usually very small, this algorithm can get desired vector representations efficiently.

Fig. 7 illustrates a typical process of the proposed centerline detector and an example-based analysis of the profile function selection strategy. The input image in Fig. 7(a) contains line regions as well as some disturbances resulted from LWF based small object detector. The result images acquired by evidence map acquisition, centerline generation and incremental line fitting with GPF, EPF

and CPF are presented in Fig. 7(b–d), respectively. We can see from these images that the detector with GPF gets the best result among these profile functions. In addition, due to the relatively small value of k and bandwidth h_c , evidence maps with any profile function can be estimated efficiently by means of accumulated image as described in Section 2. This stimulates us to select GPF as the profile function in the proposed centerline detector. By experience, profile functions like CPF which are too flat or those like delta function which are too sharp are all unsuitable for evidence map generation and there may be a balance between them which can get the best estimation quality. Our experimental results indicate that GPF outperforms EPF and CPF. The analysis of GPF compared to any other profile functions, the task of constituting appropriate criterion to evaluate the quality of estimated evidence map by different profile functions, and the issue whether GPF is the best profile function for evidence map acquisition in centerline detection are left as our future work. However, the proposed centerline detector with GPF can get 4-neighborhood connected centerlines and their vector representations efficiently with satisfactory precision in our experiments.

5. Experimental results

In this section, we first present the experimental results achieved by using synthesized and real images in several applications for small object detector, and compare them with existing

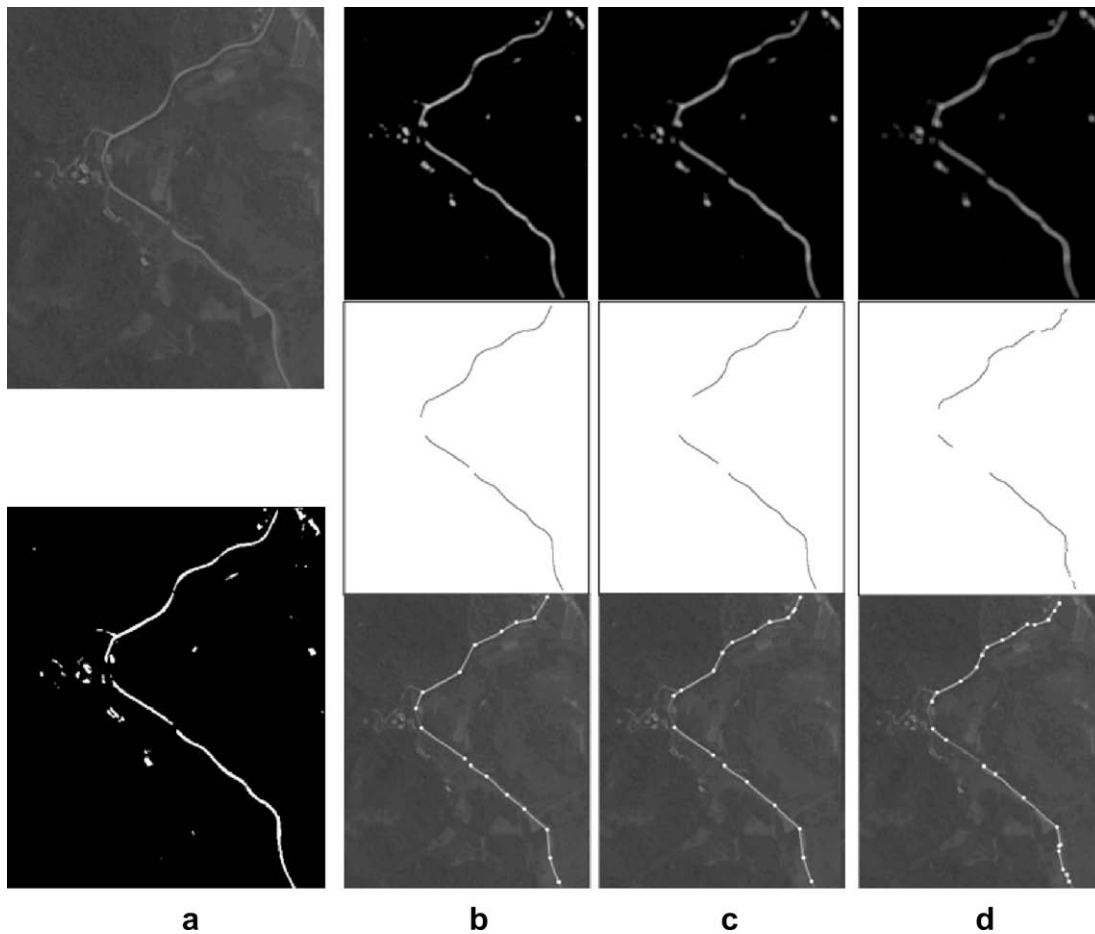


Fig. 7. Illustration and analysis of centerline detection by means of LWF with different profile functions: (a) input binary image generated by LWF based small object detector from the gray image on the top; (b)–(d) results of evidence map by using LWF, centerline by means of non-maximum suppression and seed following, and vector representations of curvilinear structures generated by incremental line fitting with GPF, EPF, and CPF, respectively. Vector representations, which are composed of control points shown as white dots connected by white lines, are imposed upon the original gray image to make evaluations convenient.

algorithms: a recently proposed wide line detector [22] and morphological based operators [21]. Then, several aerial image sequences are employed to test the validity of the proposed linear structure extraction and delineation algorithm. Some representative results are reported below.

5.1. Small object detection

Fig. 8 shows some detection results by using the proposed LWF based small object detector and wide line detector [22], respectively. The wide line detector employs segment test with a circular mask and its parameters are carefully toned to get the best results. The top row of Fig. 8 is a synthesized image including straight lines, curves, points, and large regions with different widths and intensities. It can be seen from the detection results that the performances using these two methods are comparable for detecting curvilinear structures and point features. However, at edges of large regions, the result by using our method is better than that obtained by the wide line detector. The reason is that there is usually a transition of intensities at edges and this makes the edges be dif-

ferent from both objects and background. These edges which are undesired satisfy the segment test criterion of the wide line detector and are therefore detected. Instead, our method solves this problem because the background values acquired by density estimation are similar to the pixel intensities at these edges since different values at both sides of them are counteracted as illustrated in Fig. 6(c).

The second row in Fig. 8 is another example in vision based shot detection. This example is selected because point features and lines are all expected in this application to determine the positions, orientations and circles of shots. We can see that the result of our method is better than that generated utilizing the wide line detector especially inside the inner circle and around the shoulder. The above properties of these methods are some of the main reasons. In addition, our method combines density estimation scheme and circular analysis technique which are both insensitive to noises, and this makes the detection result smoother than that by using the wide line detector.

A road detection example is illustrated in the last row of Fig. 8. Detection of curvilinear structures such as roads and rivers from

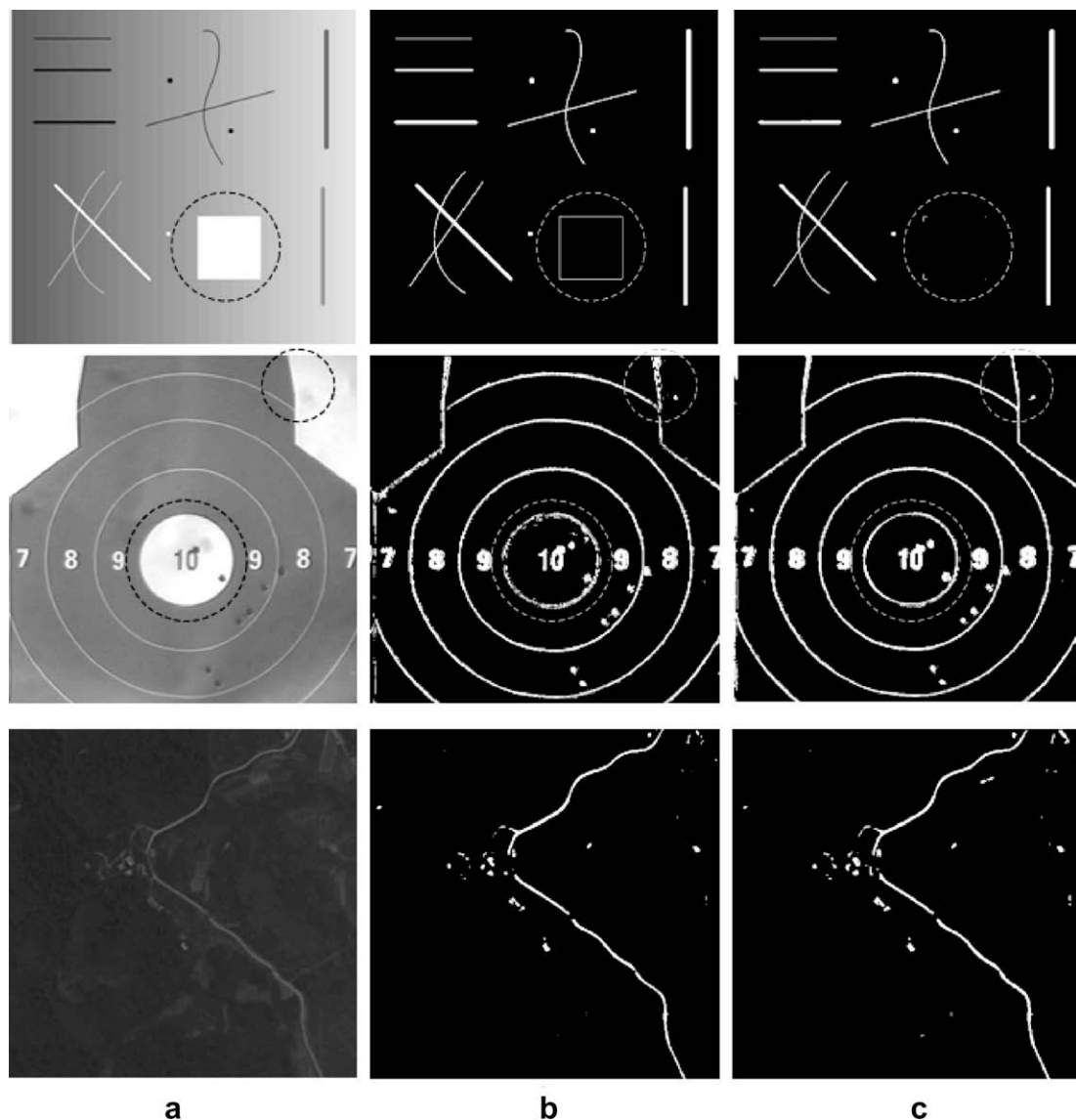


Fig. 8. Experimental results for small object detectors: (a) original images; (b) result images utilizing the wide line detector; (c) result images by means of the proposed small object detector based on LWF. The dotted circles imposed on images indicate the main detection difference areas between these two methods.

Table 2
Computational cost comparison between the proposed method and the wide line detector on a PIV 2.4G processor.

	Synthesized image (500 × 480) ms	Shot image (460 × 460) ms	Road image (480 × 480) ms
The wide line detector	885	992	837
Our method	7.9	8.7	6.8

low resolution aerial images is very important for vision based UAV navigation. It can be seen that the detection result obtained by the wide line detector is comparable to that acquired using the proposed method. The reason is that there are no large objects and no high level noise in the image. The main difference lies in the calculation costs as shown in Table 2. Our method is approximately 100 times faster than the wide line detector while achieving similar or better results, which makes it competent for real-time applications.

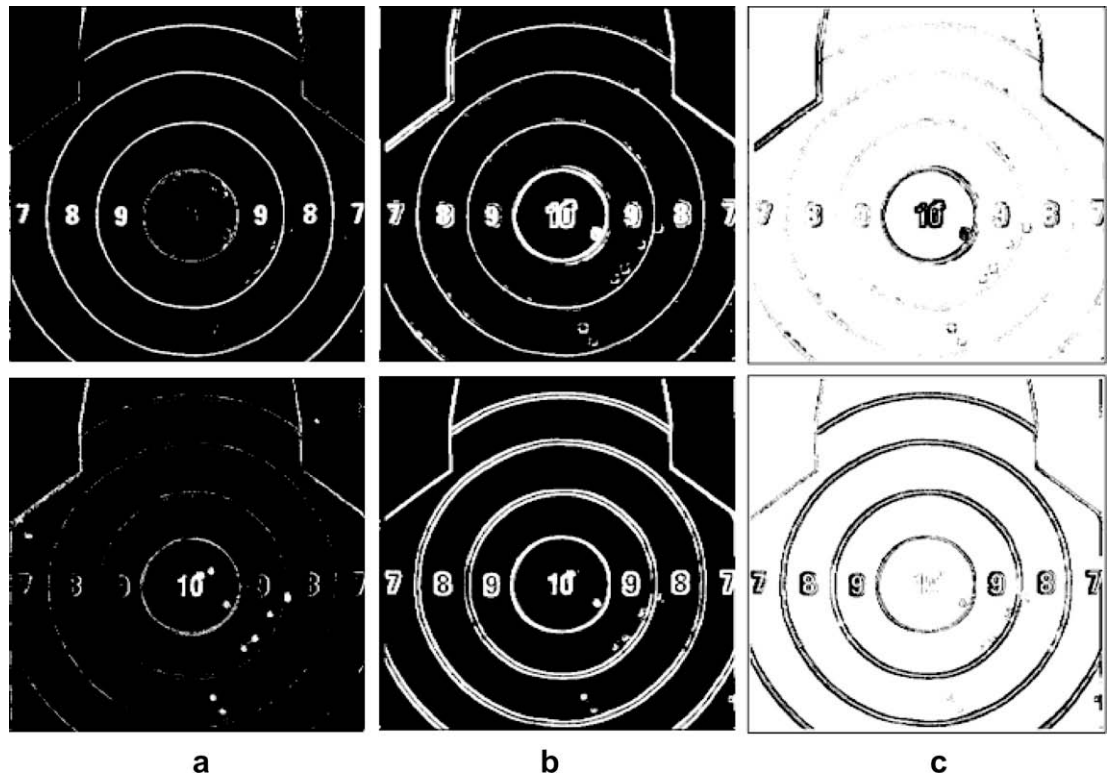


Fig. 9. Experimental result comparison for the shot image in Fig. 8 between our method and morphological based operators: (a) detection results by LWF based bright object detector and dark object detector, respectively; (b) segmentation results by top-hat and bottom-hat operators, respectively; (c) the wrongly extracted regions that LWF based method does not detect while morphological based method does.

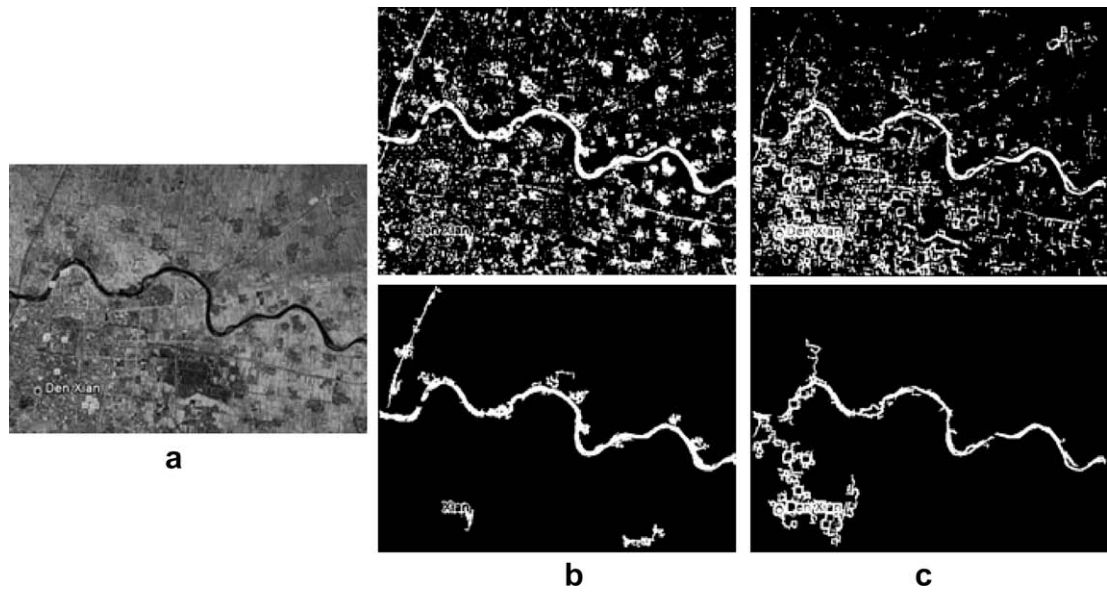


Fig. 10. Experimental result comparison for a river image between our method and bottom-hat operator: (a) original image; (b) results of our method; (c) results of bottom-hat operator. The segmented image and filtered image are displayed at the top and bottom row, respectively.

Another state of the art small object detector is the morphological top-hat and bottom-hat operators, which are usually used to extract the bright and dark structures in the image. Fig. 9 shows the bright and dark object detection results for the shot image in Fig. 8 by using the LWF based method and the morphological based method. The segmentation threshold for top-hat and bottom-hat operators is set as 30 to get the best results for the shot image. The ideal bright object detection results should mainly include circles and the number “7–9”, while the darker results should consist of the shots and the number “10”. Fig. 9(c) shows the wrongly extracted regions that LWF based method does not detect while morphological based method does. We can see that the detection results of our method are better than that of morphological ones. In addition, the proposed method is approximately 6 times faster than top-hat or bottom-hat operators and thus is more suitable for real-time applications.

A river detection example under clutters is illustrated in Fig. 10. The segmentation threshold for bottom-hat operator is toned to be 70 to get the best results for the river image, while all parameters of our method are with the default settings. A size filter is followed for both methods to remove the noisy regions with less than 400

pixels. This does not affect the real-time performance because it only increases the cost by about 15%. Results for the proposed method and the bottom-hat operator are shown in Fig. 10(b) and (c), respectively. Compared to bottom-hat operator, our method is easy to use and can extract rivers more efficiently and reliably.

5.2. Curvilinear structure extraction and delineation

When the proposed small object detector and centerline detector work in turn, they behave in whole as a novel curvilinear structure extraction and delineation algorithm. The results obtained by using small object detector usually contain curvilinear structures as well as point features and noises. However, point features and noises are then removed implicitly and effectively by utilizing the centerline detector. As a result, the final output after the two detectors only contains vector representations of curvilinear structures, which is very important for UAV navigation based on GIS and vision as has been described in Section 4. We test the algorithm on a number of aerial sequences and report some typical results below.

Fig. 11 shows some road detection and delineation results from an aerial image sequence under camera zoom and focus. The reso-

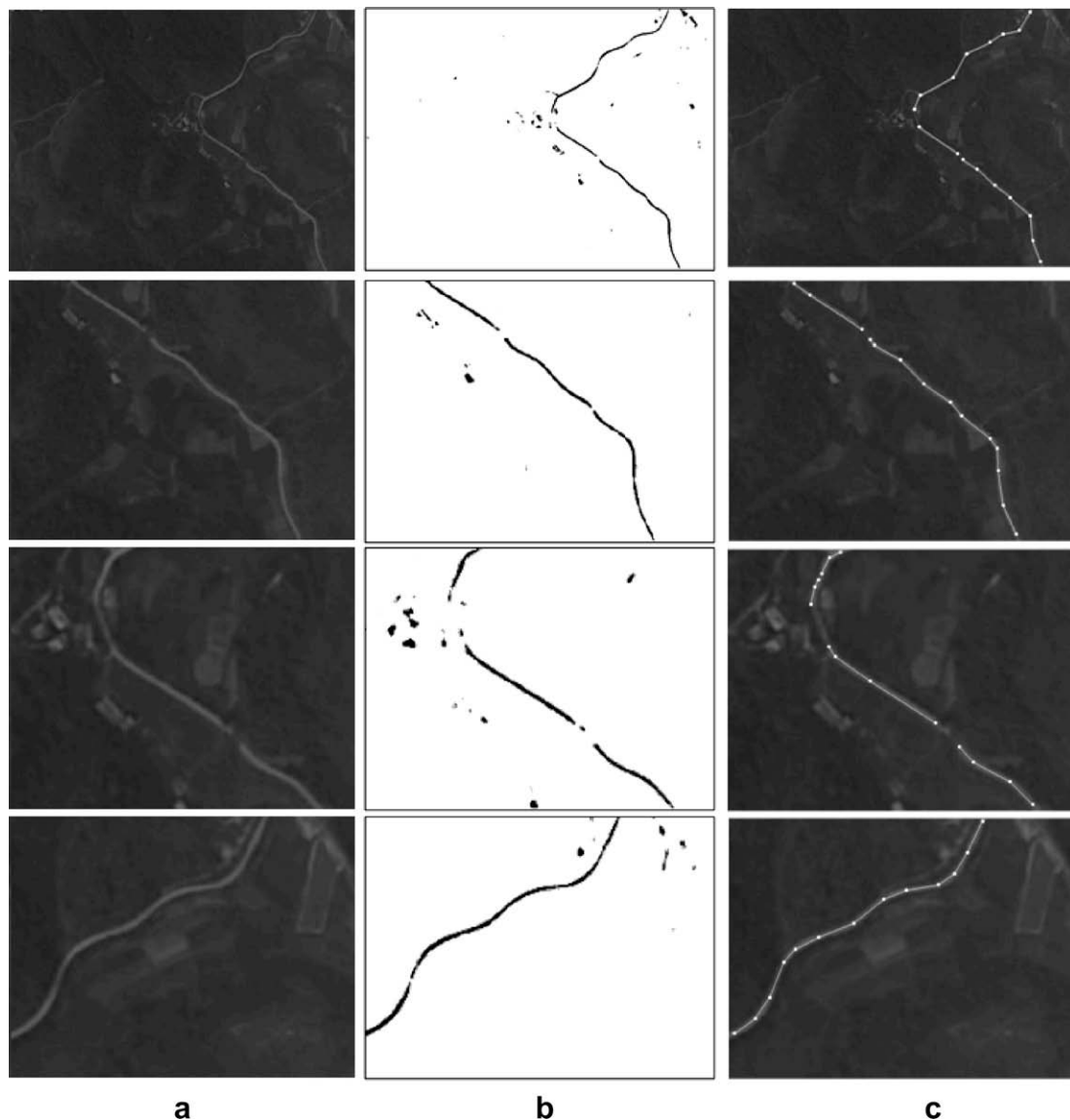


Fig. 11. Line detection and delineation results for road sequence under camera zoom and focus: (a) original images; (b) line region images; (c) vector representations imposed upon the original gray images. This sequence has 550 frames with a resolution of 640×480. Frames 1, 44, 395 and 525 are shown from the top to the bottom in turn.

lution varies approximately from 4 m to 1.6 m per pixel. Road width varies from 4 to 10 pixels in the test sequence while all parameters of the algorithm are fixed with the default values. It can be seen from these images that our method achieves good performance with the default settings of all parameters, which makes it very convenient to use in practices. The reason of this property can be explained from Fig. 4 in part. For thin lines, the default bandwidths h_o are larger than $4h_g$, so a higher quality score may be produced for background estimation though it will be more dependent on image contents and specific applications. In addition, carefully tuning parameters is troublesome and even unfeasible when lines of different widths exist in the same image. Experiments show that though a bit more disturbances do appear in the detection results with the default values, most of these wrongly detected noises are removed during the centerline extraction process.

Fig. 12 displays some road detection and delineation results from an aerial image sequence under clutter scenes. The resolution is about 2 m per pixel. Several factors such as low intensity contrast, various objects, complex scenes and dirty lens exist in the sequence making the task very difficult. It can be seen from the results that our method can cope with these difficulties. The main reason is that the proposed algorithm is insensitive to noise. In addition, the profile

function selection strategy decreases the required bandwidth for small object detection, which reduces the dependences of background estimation quality on surroundings far away.

The algorithm has been tested on a PIV 2.4G processor. It is implemented in C++ on the plate of Visual C++ 6.0 with elementary optimizations. In these conditions, the cost of extraction and delineation of lines from one image sized of 640×480 takes about 18 ms, which indicates that a rate of 50 *fps* can be achieved in practices by using our method.

6. Conclusions

In this paper, we present a novel algorithm to get the regions as well as vector representations of curvilinear structures by using LWF. The proposed algorithm consists of two components: (a) small object detector and (b) centerline detector. Our small object detector utilizes background subtraction strategy by means of LWF with CPF, followed by a noise removal process employing circular analysis. The detection results are then further processed to get vector representations of lines by a novel centerline detector, which uses LWF based evidence map estimation with GPF, seed generation, line following and incremental line fitting subsequently. This algorithm is very fast, because LWF with CPF for

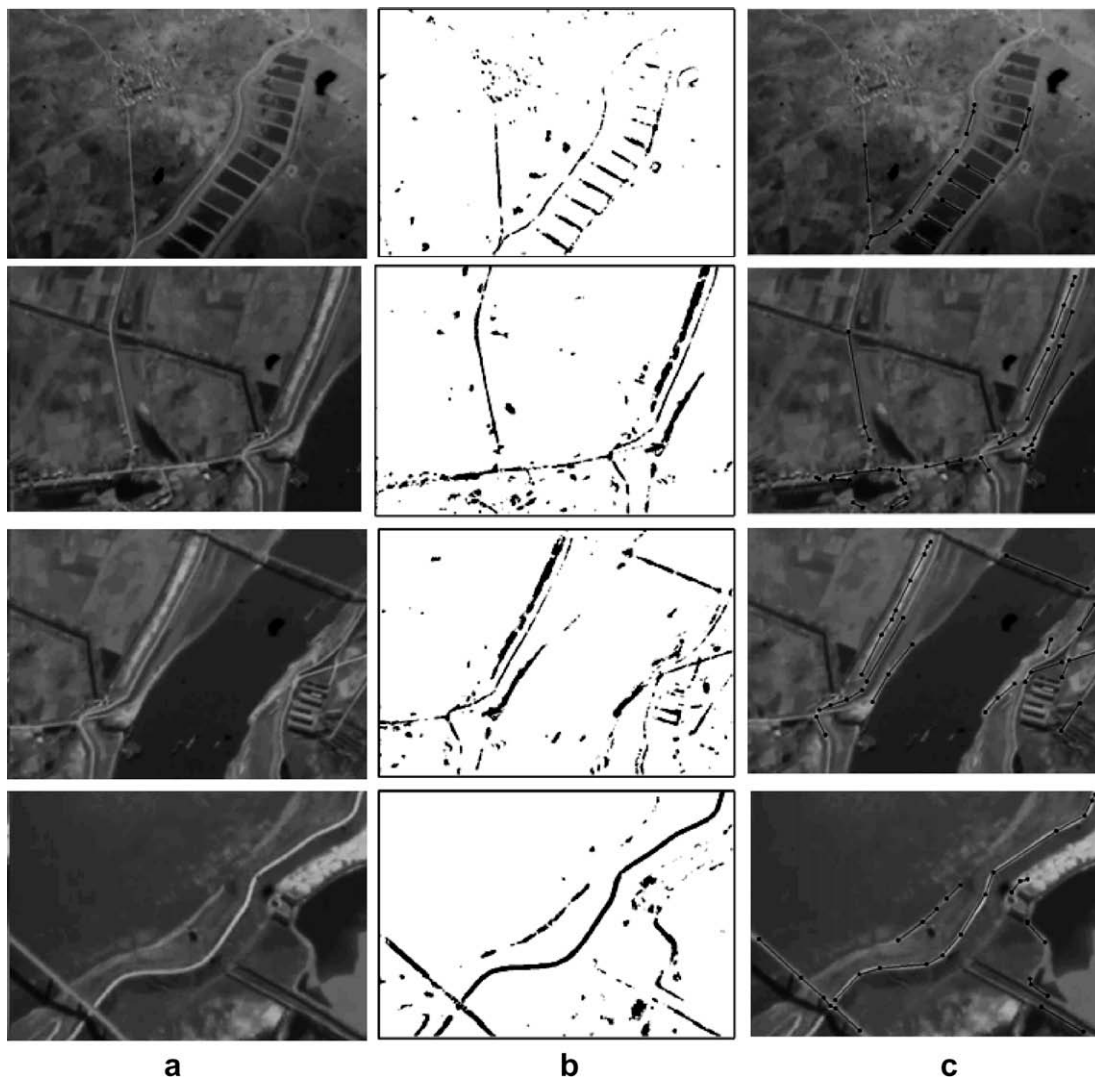


Fig. 12. Line detection and delineation results for road sequence under clusters: (a) original images; (b) line region images; (c) vector representations imposed upon the original gray images. This sequence has 586 frames and the image size is 620×432 . Frames 6, 445, 465 and 499 are shown from the top to the bottom subsequently.

small object detector and LWF with GPF for centerline detector can be efficiently calculated by integral image and accumulated image, respectively. A general calculation strategy for LWF is also discussed to provide guidelines for other related applications. Experimental results on a number of images and videos indicate that the performance and computational speed of our small object detector are superior to those of the wide line detector and morphological based operators and that the proposed line detection and delineation method is very efficient and effective.

Further investigation and improvement could include bringing forward profile function selection criterion for evidence map quality estimation, proving whether GPF is the best profile function to get evidence map for centerline detection, using more experimental images and sequences, and applying LWF to other image processing tasks. Considerations will also be given to studying the matching method to GIS, evaluating the feature detection results and exploring the application of the proposed method to UAV navigation systems and other surveillance systems.

Acknowledgments

This work is supported by China 863 Programming (2007AA04Z233). The authors thank HongDu Group for providing UAV videos. They also thank the anonymous reviewers for their constructive advice and for the revision of the paper.

References

- [1] N. Merlet, J. Zerubia, New prospects in line detection by dynamic programming, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (2) (1996) 426–431.
- [2] A.W.K. Loh, M.C. Robey, G.A.W. West, Analysis of the interaction between edge and line finding techniques, *Pattern Recognit.* 34 (2001) 1127–1146.
- [3] J.H. Jang, K.S. Hong, Linear band detection based on the Euclidean distance transform and a new line segment extraction method, *Pattern Recognit.* 34 (2001) 1751–1764.
- [4] D.S. Guru, B.H. Shekar, P. Nagabhushan, A simple and robust line detection algorithm based on small eigenvalue analysis, *Pattern Recognit. Lett.* 25 (2001) 1–13.
- [5] T.M. Koller, G. Gerig, G. Szekely, D. Dettwiler, Multiscale detection of curvilinear structures in 2-D and 3-D image data, *Proc. IEEE Int. Conf. Comput. Vis.*, Boston (1995) 864–869.
- [6] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, *Int. J. Comput. Vis.* 30 (2) (1998) 117–175.
- [7] M. Jacob, M. Unser, Design of steerable filters for feature detection using canny-like criteria, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (8) (2004) 1007–1019.
- [8] D. Eberly, R. Gardner, B. Morse, S. Pizer, D. Scharlah, Ridges for image analysis, *J. Math. Image Vis.* 4 (1994) 353–373.
- [9] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2) (2004) 91–110.
- [10] N. Aggarwal, W.C. Karl, Line detection in images through regularized Hough transform, *IEEE Trans. Image Process* 15 (3) (2006) 582–591.
- [11] S.X. Li, H.X. Chang, Novel road detection and tracking algorithm for aerial images, *J. Beijing Univ. Aeronaut. Astronaut.* 33 (4) (2007) 445–449. in Chinese.
- [12] Q.P. Zhang, I. Couloigner, Accurate centerline detection and line width estimation of thick lines using the Radon transform, *IEEE Trans. Image Process* 16 (2) (2007) 310–316.
- [13] N. Bennett, R. Burrige, N. Saito, A method to detect and characterize ellipses using the Hough transform, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (7) (1999) 652–657.
- [14] V. Leavers, Use of the two-dimensional Radon transform to generate a taxonomy of shape for the characterization of abrasive powder particles, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (12) (2000) 1411–1423.
- [15] J. Illingworth, J. Kittler, A survey of the Hough transform, *Comput. Vis. Graph. Image Process* 44 (1988) 87–116.
- [16] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (1981) 381–395.
- [17] D.A. Forsyth, J. Ponce, *Computer Vision—a Modern Approach*, Pearson Education Inc., 2003.
- [18] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognit.* 13 (2) (1981) 111–122.
- [19] P. Bhowmick, B.B. Bhattacharya, Fast polygonal approximation of digital curves using relaxed straightness properties, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (9) (2007) 1590–1602.
- [20] H.D. Lin, Automated visual inspection of ripple defects using wavelet characteristic based multivariate statistical approach, *Image Vis. Comput.* 25 (11) (2007) 1785–1801.
- [21] X.Y. Jin, C.H. Davis, Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks, *Image Vis. Comput.* 25 (9) (2007) 1422–1431.
- [22] L. Liu, D. Zhang, J. You, Detecting wide lines using isotropic nonlinear filtering, *IEEE Trans. Image Process* 16 (6) (2007) 1584–1595.
- [23] S.M. Smith, J.M. Brady, SUSAN—a new approach to low level image processing, *Int. J. Comput. Vis.* 23 (1) (1997) 45–78.
- [24] S.M. Chao, D.M. Tsai, An anisotropic diffusion-based defect detection for low-contrast glass substrates, *Image Vis. Comput.* 26 (2) (2008) 187–200.
- [25] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (5) (2002) 603–619.
- [26] B. Georgescu, I. Shimshoni, P. Meer, Mean shift based clustering in high dimensions: a texture classification example, *Proc. IEEE Int. Conf. Comput. Vis. France* (2003) 456–463.
- [27] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (5) (2003) 564–577.
- [28] P. Viola, M.J. Jones, Robust real-time face detection, *Int. J. Comput. Vis.* 57 (2) (2004) 137–154.
- [29] C.J. Yang, R. Duraiswami, L. Davis, Fast multiple object tracking via a hierarchical particle filter, *Proc. IEEE Int. Conf. Comput. Vis. Beijing* (2005) 212–219.
- [30] T. Ojala, M. Pietikainen, T. Maenpaa, Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7) (2002) 971–987.
- [31] M. Heikkila, M. Pietikainen, A texture-based method for modeling the background and detecting moving objects, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (4) (2006) 657–662.
- [32] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, *Proc. IEEE Int. Conf. Comput. Vis. Beijing* (2005) 1508–1515.
- [33] E. Rosten, T. Drummond, Machine learning for high speed corner detection, *Eur. Conf. Comput. Vis.* (2006) 430–443.
- [34] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (6) (1986) 679–698.