Optik 127 (2016) 4794–4798

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Multiscale edge fusion for vehicle detection based on difference of Gaussian

Kenan Mu [a,*], Fei Hui [a], Xiangmo Zhao [b], Christian Prehofer [c]

[a] School of Information Engineering, Chang'an University, Xi'an, China
[b] Chang'an University, Xi'an, China
[c] Fortiss Research Institution, Technical University Munich, Munich, Germany

ARTICLE INFO

Article history:
Received 5 November 2015
Accepted 6 January 2016

Keywords:
Edge detection
Vehicle detection
Multiscale fusion
Difference of Gaussian

ABSTRACT

Edge information help highlight the contour as well as cast shadow of objects. As the low complexity for edge extraction, the edge-based methods are widely used in vehicle detection. Traditional edge-based vehicle detection methods are easily interfered by noise and background, which resulting in inaccurate false detection. In this paper, a vehicle detection method based on multiscale edge fusion is proposed. First, multiscale images are obtained from the decomposition of the DoG pyramid. Second, multiscale edges are extracted by the DoG operator in multiscale images. Third, different scale edge map are fused according to the proposed multiscale edge fusion strategy. Then, an accurately located, low redundant and strongly anti-noise edge map is obtained. Finally, morphological operation and connectivity analysis are applied on the edge fusion map. Experiments with traffic images in different weather conditions verify the practicability of the proposed method. Comparison with related method in detection rate and detection accuracy verifies the superiority of the proposed method.

1. Introduction

Computer vision-based vehicle detection technology is a hot research topic in intelligent traffic image processing and has been widely used in the field of Intelligent Traffic System (ITS), such as Driver Assistance Systems – preceding vehicle detection [1], blind spot vehicle detection [2,3] and rear/lateral vehicle detection [4], and Traffic Information (vehicle flow, vehicle type) Collection System [5].

Vehicle detection based on computer vision can be roughly divided into three classes: the model-based, the neural network training-based and feature-based methods [6–8]. In [9] and [10], the model-based method is used to match the candidate vehicle region with the vehicle model which are pre-established in database. However, the detection result of this method depends greatly on the geometric modeling of all kinds of vehicles, which is difficult to realize. In [11–13], the neural network training-based method is used to identify vehicles. Training the neural network with samples (vehicles) and then using the trained network to identify vehicles, yet this method is often used to validate the detection results of others. The feature-based method locates vehicles by detecting the local feature, such as the symmetric components (wheels, head-lamp, rear-lamp) [14,15], shadow [16], edge [17] and so on, where the extraction of these features are usually relied on the edge detection. The advantage of this method is its suitableness for vehicle detection in rainy and snowy days, even at night by using the vehicle's features which can be identified in most circumstance. However, the method referred in [14], which locating vehicles by detecting the wheels, is easily influenced by the vehicle's pose or occlusion. And the method described in [15], which locating vehicles by detecting the lamps, would also been interfered by streetlights and city lights in night scene, thereby interfering the detection result. In addition, because such feature-based methods above usually obtain the features from the edge detection, the computational complexity is higher than the edge-based detection method. As edge-based vehicle detection method is low in computational complexity and time consumption, it is suitable for the real-time applications (e.g., traffic flow statistics [16], drive assistance). However, false detection will be easily caused by noise and background edges (such as lanes, guardrails, trees) in edge map. Therefore, the key issue for the accuracy improvement of the edge-based method is that how to highlight vehicle edges while suppress interference edge [17,18].

* Corresponding author. Tel.: +0086 15002941210.
E-mail addresses: kenanmu@chd.edu.cn (K. Mu), feihui@chd.edu.cn (F. Hui), xmzhao@chd.edu.cn (X. Zhao), prehofer@fortiss.org (C. Prehofer).

http://dx.doi.org/10.1016/j.ijleo.2016.01.017
0030-4026/

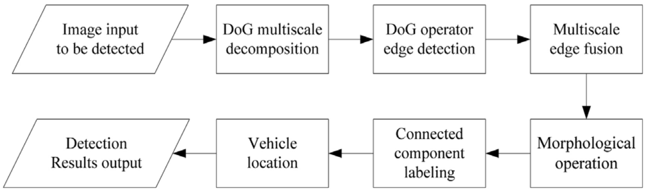

Fig. 1. Flowchart of the proposed vehicle detection method.

Consideration with the problems above, this paper proposes a vehicle detection method with multiscale fusion edge based on Difference of Gaussian (DoG) pyramid. The flowchart of the proposed method is shown in Fig. 1. First, multiscale images are obtained from DoG pyramid the decomposition of DoG pyramid, the detection process is simplified. Then, combining the accurate edge location in small scale image with the clear contour in large scale images, to obtaining the accurate and low redundant edge map. Finally, morphological operation and connectivity analysis are performed to obtain target vehicles.

The remainder of this paper consists of following sections. Section 2 introduces the process of the proposed multiscale edge fusion based on DoG pyramid in detail. Section 3 describes the post-processing to identify target vehicle regions, including morphological operation and connectivity analysis in the fused edge map. In order to verify the effectiveness and superiority of the proposed method, Section 4 provides experiments and validation, shows detection results of the proposed method detecting in the traffic images under different weather conditions, and compares the detection results with the traditional single scale edge-based detection method. Section 5 offers some concluding remarks and discussions of future works.

2. Multiscale edge fusion based on DoG pyramid

2.1. Difference of Gaussian pyramid decomposition

Lowe's paper has put forward the multiscale image, and achieved the multiscale decomposition by constructing Gaussian pyramid of the image. The first step of the decomposition is up sampling the source image. Then the up-sampled image is done convolution with a series of Gaussian kernels with different scale parameters in the second step. The third step is down sampling the convolution images. Finally, repeating the steps above to generate a series of images with continuous scale. Thus, a Gaussian pyramid is generated [18]. Difference of Gaussian (DoG) pyramid is obtained by differing the two Gaussian images with adjacent scale in the same layer of Gaussian pyramid. Edge map detected from the small scale image is accurate in edge location but redundant in edge detail and sensitive to noise. By contrast, edge map detected from the large scale image is clear in main edge information and strong in noise immunity. So we consider to combine the advantages of the two for a better edge map. However, up and down sampling in Gaussian pyramid decomposition will cause false pixel information and loss of image information, and further decrease the precision of edge localization. To solve this problem, we only allow the first-order Gaussian pyramid decomposition on the original image without up sampling. The decomposition process is introduced in this section.

Fig. 2 shows the decomposition process of the first-order Gaussian and DoG pyramid, where $I(x, y)$ is the source image, $g_i\,(i = 0, 1, \ldots, n-1)$ are a series of Gaussian images with continuous scale, $d_j\,(j = 0, 1, \ldots, n-2)$ are the corresponded DoG images. In particular, $g_0 = I(x, y)$. $G_j\,(j = 0, 1, \ldots, n-2)$ are Gaussian kernels with continuous scale parameters. The decomposition process is described as follows:

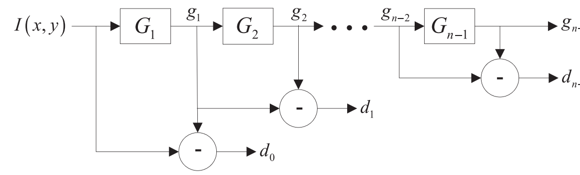

Fig. 2. Decomposition of Gaussian pyramid and DoG pyramid.

Step 1: Initializing $i = 0, j = 0$;

Step 2: Initializing $g_0 = I(x, y)$;

Step 3: Image g_i is done convolution with the Gaussian kernel G_j, and g_{i+1} is obtained as follows,

$$g_{i+1}(x, y) = g_i(x, y) * G_j\left(x, y, \sigma_j\right) \tag{1}$$

$$G_j\left(x, y, \sigma_j\right) = \frac{1}{2\pi\sigma_j^2} e^{-\left(x^2+y^2\right)/2\sigma_j^2} \tag{2}$$

where '*' represents convolution operation, σ_j is the scale parameter;

Step 4: Differing g_i with g_{i+1}, then the DoG image d_j is obtained:

$$\begin{aligned} d_j(x, y) &= g_i(x, y) - g_{i+1}(x, y) \\ &= g_i(x, y) - g_i(x, y) * G_j\left(x, y, \sigma_j\right) \end{aligned} \tag{3}$$

Step 5: $i = i + 1, j = j + 1$, repeating Steps 3 and 4. When $i > n-1$ and $j > n-2$ the decomposition ends.

At this point, n Gaussian images and $n-1$ DoG images are obtained. Fig. 3(b) shows DoG images with continuous scale, which are got from the difference of the first-order Gaussian pyramid when $n = 5$.

2.2. Edge detection

DoG operator is one of the edge detection operators with second-order difference. As the second-order derivative of DoG function cross the zero point, DoG operator regards the Zero Crossing Points (ZCP) as edge points [19,20]. The mathematical expression for the DoG operator is defined as follows,

$$\begin{aligned} DoG &= G_{\sigma_1} - G_{\sigma_2} \\ &= \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1} e^{-\left(x^2+y^2\right)/2\sigma_1^2} - \frac{1}{\sigma_2} e^{-\left(x^2+y^2\right)/2\sigma_2^2}\right] \end{aligned} \tag{4}$$

where G_{σ_1} and G_{σ_2} are the Gaussian kernels with standard deviation σ_1 and σ_2, respectively.

The process of DoG edge detection consists of the following steps.

Step 1: Initializing the standard deviation of the Gaussian kernel;

Step 2: Initializing the DoG mask;

Step 3: The image is done convolution with the DoG mask, and the convolution image e is obtained;

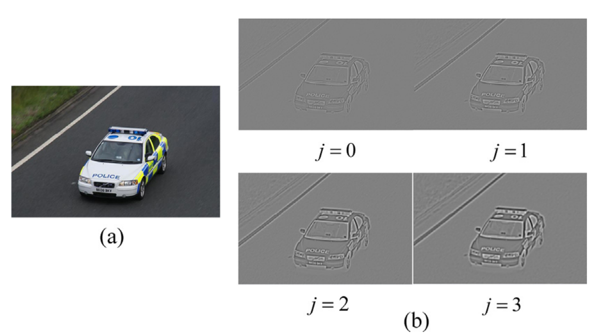

Fig. 3. Result of first-order DoG decomposition. (a) Source image. (b) Multiscale DoG images.