Efficient Moving Objects Detection by Lidar for Rain Removal

Yao Wang, Fangfa Fu, Jinjin Shi, Weizhe Xu, and Jinxiang Wang(✉)

Microelectronics Center, Harbin Institute of Technology, Harbin, China
jxwang@hit.edu.cn

Abstract. Rain and snow are often imaged as brighter streaks, which can not only confuse human vision but degrade efficiency of computer vision algorithm. Rain removal is very important technique in these fields such as video-surveillance and automatic driving. Most existing methods rely on optical flow algorithm to detect rain pixel and estimate motion field. However, it is extremely challenging for them to achieve real-time performance. In this paper, a LIDAR based algorithm is proposed, which is capable of achieving rain pixel robustly and efficiently from motion field. The motion objects (vehicles and human) are identified for separation by LIDAR (Sick LMS200) in this paper. Then rain pixels on moving objects are removed by bilateral filter which can preserve edge information instead of causing blurring artifacts around rain streaks. Experimental results show that our method significantly outperforms the previous methods in removing rain pixel and detecting motion objects from motion field.

Keywords: Rain removal · Real-time · LIDAR · Bilateral filter

1 Introduction

The demand for computer vision system in the fields of video-surveillance, military, and automatic driving, etc. is increasing rapidly. Due to the effect of bad weather condition, such as rain, snow, fog and mist, current systems are facing great challenges for the poor performance. According to [1], the detection of rain is a difficult task because both rain and moving objects produce sharp intensity changes in dynamic scene. We should treat these two parts separately instead of causing artifacts around other pixels. Thus methods that can remove visual effects and frequent intensity fluctuations caused by undesirable weather conditions are necessary. In order to remove the rainy effect, separating object motion from rain and removing rain pixels are very crucial. Most previous algorithms have been proposed for this purpose.

Most of the applications simply consider how to remove rain pixels from images, regardless of paying more attention to detecting moving objects efficiently. One of the excellent works in rain removal and separating moving objects from dynamic scenes was studied by Grag and Nayar [2], which shows impressive result but the direction and velocity of rain drops limited its functions.

Another work adopted an alternative approach by Barnum et al. [3, 4] based on the frequency analysis of rain streaks [5, 6]. It is assumed that rain streaks in an entire video sequence have similar shapes and orientations. Therefore they detected the rain

D.-S. Huang et al. (Eds.): ICIC 2016, Part III, LNAI 9773, pp. 697–706, 2016.
DOI: 10.1007/978-3-319-42297-8_64

drops by selecting repeatedly occurring frequency components through the dynamic scenes. However their method always ignored rain drops from moving objects and regard moving objects as the region of rain streaks.

Chen and Chau [7] proposed a method based on optical flow to detect and separate moving objects. Their method was suitable to few moving objects in the dynamic scene. As moving objects increase, optical flow algorithm results in low performance.

To overcome these drawbacks, this paper proposes a method to detect moving objects by LIDAR without using optical flow or physical properties of rain. First we employ LIDAR to scan moving objects according to different speed and map LIDAR coordinates to image plane. Then we set rain streaks as two parts: motion objects and static background. At last we treat two parts respectively by bilateral filter and Gaussian filter instead of causing blurring.

The rest of this paper is organized as follow. We review some related work in Sect. 2. Then we propose our algorithm: Motion segmentation by LIDAR, Rain detection and Motion Exclusion are described in Sect. 3. Experimental results of detecting and removing rain pixels are presented in Sect. 4 with some conclusions follow in Sect. 5.

2 Related Work

One of the notable works in detecting rain pixels by identifying rain pixels with high intensity changes in the scene, and then recovering rain pixels by taking the average pixel values of neighboring frames. Garg and Nayar [2] based on physical and photometric properties of rain to detect and remove rain from images. However this method fails to separate rain from moving objects and consider all the pixels as rain pixels no matter if there is a moving object in the scene. Therefore this method always causes blur artifacts around moving objects. In order to reduce blurring artifacts around rain streaks, Kim and Lee [5] detected rain streak regions using shape and orientation features, then remove rain by an adaptive nonlocal means filter to remove rain from single image. Liu el at. [6] employed the detecting function based on chromatic properties and the Kalman filter for rain removal. Ding and Chen [8] removed rain and snow via guided L0 smoothing filter from single image based on edge characteristic of rain or snow. These methods fail to performance better in removing rain when facing with dynamic scenes. Chen and Chau [7] put forward a method to remove rain in dynamic scenes. They applied Expectation Maximization (EM) on an optical flow field to separate motion map into two parts: moving objects and static background. Rain pixels are treated respectively by using different filters depending on whether they are belonged to moving objects or static background. Their method performance better in distinguish moving objects because of optical flow. Shen and Xue [9] proposed a fast method by using three successive frames to detect rain based on optical flow. However optical flow is not real-time performance when moving objects increased. To overcome the limitation of optical flow, Tan and Chen [10] used aligning neighboring frames combined with optical flow to detect rain pixels. Varun and Vijayan [11] employed phase congruency features to find rain pixels and applied optical flow to get motion map from dynamic scenes. However their method works poorly when moving objects increase. In general, current methods are not suitable for practical requirements with respect to their detection accuracy and recovery performance.

All the above methods suffer the same problem that they are not applicable to detect moving objects efficiently. To address the limit of the previous approaches, we put forward simple but efficient moving objects detection and rain drops removing method in this paper. That is, we employ LIDAR to scan objects in front of vehicle and distinguish their motion situation by the speed. Then we map the position of moving objects from LIDAR coordinates to image plane. Therefore we set whole rain image as two parts: moving objects and static background. Finally we treat these two parts to remove rain pixels by bilateral filter and Gaussian filter.

3 Proposed Algorithm

The proposed algorithm consists of three parts — for each frame, we apply motion segmentation by LIDAR, detect rain pixels, and then recover each rain pixel.

3.1 Motion Segmentation by LIDAR

The rain and object motion always cause pixel intensity fluctuation of a rainy scene. The fluctuations caused by rain should be removed, and the ones caused by moving object need to be retained. Thus motion field segmentation naturally becomes a fundamental procedure of rain removal algorithm. Previous approaches employed optical flow to estimate moving object. We briefly describe the method here. That method needs two adjacent frames to get motion cue by computing optical flow filed. After thresholding the magnitude of the optical flow field, applying a Gaussian Mixture Model to the resulting binary image in order to get motion objects locations in the target frame. The motion cue is a likelihood function for motion objects based on the optical flow field. However, using optical flow for motion target segmentation has its intrinsic drawbacks, as it can only detect obvious intensity changes and areas that belong to the target without obvious intensity change cannot be effectively recognized, which causes the so called "Aperture Problem" [4]. The accuracy of optical flow always depends on Gaussian Mixture Model (GMM) in order to overcome "Aperture Problem", but its computational complexity is relatively high due to cluster motion field and it is are not capable of real-time applications. We employ LIDAR to detect moving objects efficiently [12] without using other algorithms.

First, we use LIDAR to scan object in front of LIDAR [13], the scanning range of LIDAR is 180° which is capable of scanning all the moving objects. LIDAR is set on the top of our vehicle and the height of LIDAR is 0.89 m, which is the half height of vehicle or the pedestrian. We can get location of object by LIDAR data. By employing distance data from LIDAR, we can get the speed of object which aids us to identify whether the object is static or dramatic. V_P and V_c in (1) are the speed of object and car respectively, d is the distance data between object and LIDAR, t_1 represents time period when LIDAR emit light, t_2 represents time of reflection data form LIDAR. We set D_{th} and D_{th1} in (2) are 1 m/s and 15 m/s respectively. Therefore we can identify whether the object is static or dynamic by value of B_M in (2).

$$V_p = V_C = d/(t_1 - t_2) \tag{1}$$

$$B_M = \begin{cases} 1 & V_p > D_{th} || V_C > D_{th1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

If the LIDAR detects an object (pedestrian or vehicle), it will return $M_1(x_1,y_1,z)$ on the left side and $M_2(x_2,y_2,z)$ on the right side, so the top of the vehicle $M_3(x_1,2y_1,z)$ on the left side and $M_4(x_2,2y_2,z)$ on the right side. The bottom of vehicle $M_5(x_1,y_1\text{-}0.89,z)$ and $M_6(x_2,y_2\text{-}0.89,z)$ in the LIDAR coordinates.

LIDAR coordinates map to image plane process includes the following three steps:

(1) We map LIDAR coordinate system onto the camera coordinate system which can be expressed as follows:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + T \tag{3}$$

In the camera coordinate system, the (x_c,y_c,z_c) coordinates identify the point which transform from LIDAR coordinates (x_l,y_l,z_l). R and T are the affine matrix respectively.

(2) Then we transform camera coordinate system into image physical coordinates which can be expressed as below:

$$x = \frac{fx_c}{z_c} y = \frac{fy_c}{z_c} \tag{4}$$

In the image physical plane, the (x,y) coordinates identify the physical position of a pixel. f represents the focal length of the camera. So the relationship between two coordinates could be easily solved by affine transformation which is show as follows:

$$z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{5}$$

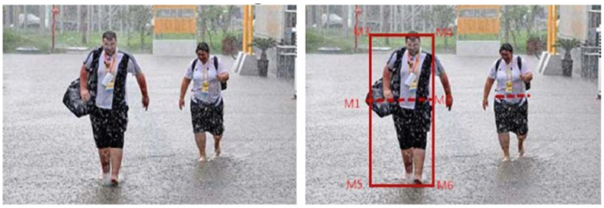
(3) In the image plane, the (u, v) coordinates identify the position of a pixel, the optical center is given by (u_0, v_0). Let $s_x = 1/dx$ and $s_y = 1/dy$ denote the value of the focal length expressed in pixels. From the perspective camera model we know that:

$$\begin{cases} u - u_0 = x/d_x = s_x x \\ v - v_0 = y/d_y = s_y y \end{cases} \tag{6}$$

The relationship between physical cameras coordinates and image coordinates could be expressed as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{7}$$

At last LIDAR coordinates map to image plane, the coordinates in the image plane are M'1, M'2, M'3, M'4, M'5 and M'6 shown in the Fig. 1.



(a) (b)

Fig. 1. LIDAR coordinates map to image plane (a) input image (b) our result

3.2 Rain Detection

First, we get image from successive frame which is shown in Fig. 2 (a). Then we detect moving vehicle by LIDAR which is shown in (b) and employ threshold method to detect rain pixels that grey scale intensity differences between two successive frames are calculated. The grey rain image map is show in (d). The threshold value are set in order to detect all the intensity fluctuations caused by rain, then we set D_{th2} is 5. The binary map is calculated which can be expressed as follows:

$$I_{diff} = \begin{cases} 1 & I_N - I_{N-1} \geq D_{th2} \\ 0 & I_N - I_{N-1} < D_{th2} \end{cases} \tag{8}$$

Then we acquire location of moving car from grey rain map according to LIDAR data in (e) which has similar result with Chen's method by optical flow shown in (c). Finally we get grey rain map without moving objects in (f).

3.3 Motion Exclusion

Rain pixels within the motion object and static background should be treated respectively, therefore we divide rain map into two parts: one is rain pixels in the moving objects and the other is rain pixels in the static background. We set rain pixels in the motion target area is A_m and rain pixels in the static background area is A_b which can be expressed as follows.

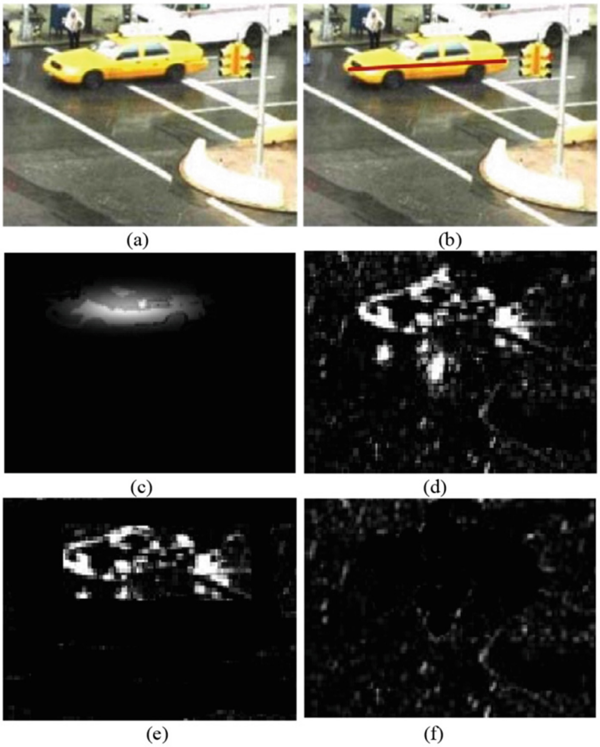


Fig. 2. The rain removal based on LIDAR (a) Our input image (b) Moving car detected by LIDAR and we map LIDAR data to image plan in order to separate moving objects. (c) Chen [6] use optical flow to detect moving objects. (d) Our result of grey images of motion object and rain map. (e) Binary image of moving object. (f) Rain map without moving object. Our results correctly distinguish moving objects from rain map, while the previous method inadequately recognizes whole region of moving object.

$$\begin{aligned} A_m &= \{I(x,y) \,|\, I_{Rain}(x,y) = 1 \& B_M = 1\} \\ A_b &= \{I(x,y) \,|\, I_{Rain}(x,y) = 1 \& B_M = 0\} \end{aligned} \tag{9}$$

Under different situations, we will use different methods to remove rain pixels. Rain pixels of A_m are recovered using bilateral filter on moving objects which can preserve edge information instead of causing blurring artifacts around rain streaks. Then the rain pixels of A_b are covered by using a Gaussian filter temporally on three frame neighbors A_b.

4 Experimental Results

We first show results of our algorithm in Fig. 3. Figure 3(a) shows several successive frames from the video of the yellow car. In Fig. 3(b) LIDAR is used for detecting moving objects, then LIDAR coordinates is mapped to image plane. Figure 3(c) shows our result of binary moving objects map which can help us locate moving objects. And Fig. 3(d) is to detect moving objects from Chen's method by applying optical flow. By comparing (c) and (d) in Fig. 3, our method preserves most parts of car which are larger than the result of Chen, because the good performance of LIDAR in detecting moving objects. Then we compare the second row in Fig. 3, the moving car is in the middle of road with high speed. The optical flow can detect parts of car show in Fig. 3(d) which is similar to our result in Fig. 3(c). In the third row of Fig. 3, the moving car is

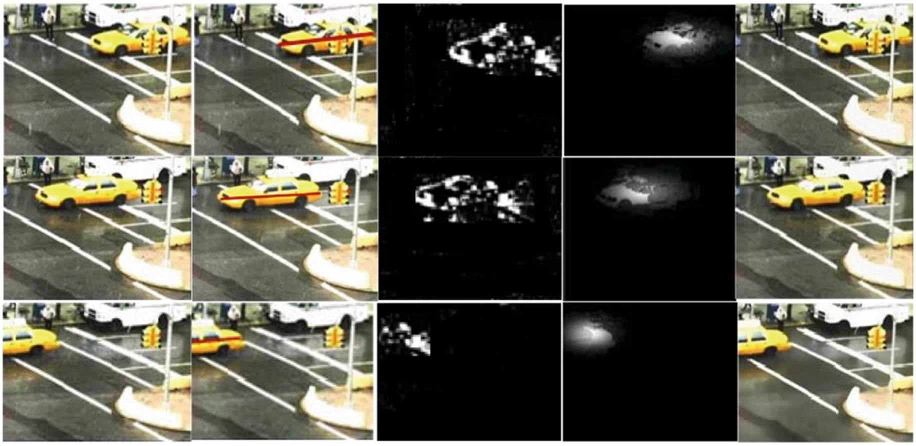


Fig. 3. Rain removal image.(a) input image. (b) moving car is detected by LIDAR. (c) separate moving car from grey rain image. (d) the result of Chen's [6] method to separate moving car. (e) rain remove image of our method. (Color figure online)

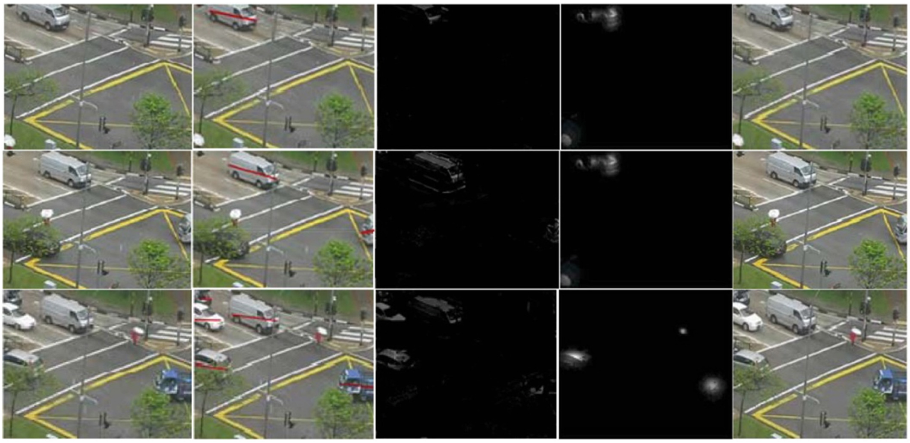


Fig. 4. increasing numbers of moving objects in dynamic scene. (a) input image. (b) moving cars detected by LIDAR. (c) map moving objects to grey image. (d) the method of Chen [6] in separating moving objects from the scene. (e) our proposed method in removing rain.

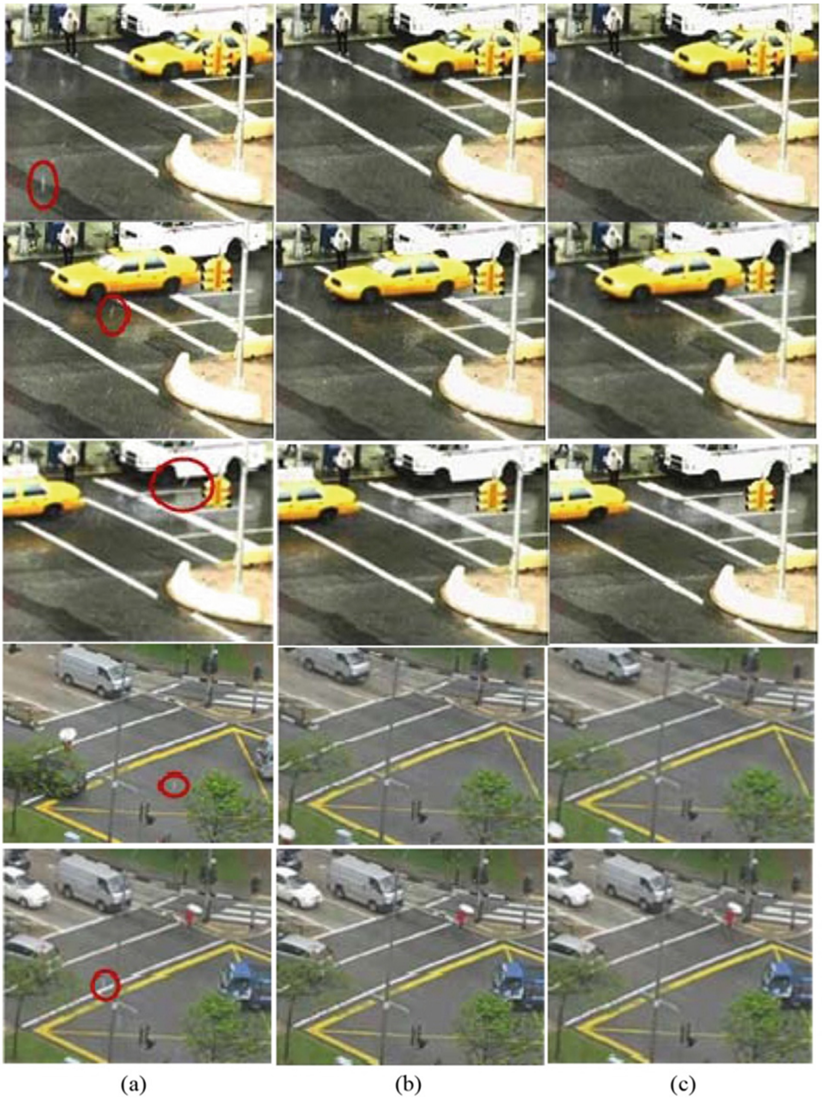


Fig. 5. the result of rain removal (a) input image. (b) and (c) are the results of our method and Chen's [6] method.

disappearing. Our result contains most parts of car while previous method lost some parts of the car.

In Fig. 4 moving objects are increasing which pose some difficulties in separating moving objects and static background. Figure 4(a) shows several successive frames from video. Our method employs LIDAR to detect moving objects. According to different speed, we can distinguish moving cars efficiently in Fig. 4(b). Due to the limit of optical flow, neither moving objects with similar speed nor with lower speed can be distinguished. Compared (c) to (d) in Fig. 4, our method detect 1,2 and 5 moving cars respectively in successive frames, however the method of Chen detect 1, 1 and 2 cars. Therefore some cars are ignored by optical flow. In Fig. 4(e) is result of our method to remove rain from image.

Next we compare the performance of two methods in removing rain streaks as shown in Fig. 5. We select five different images and rain streaks are indicated by red circle in Fig. 5(a). In Fig. 5(b) we get similar result as show in (c). Finally we compare the time of our method and previous method, Chen [6] use 300 ms to separate moving objects from motion filed by optical flow and finish rain removal every frames, however our method just costs 100 ms to detect and remove rain pixels. Our rain removal result is better than previous method especially in detecting moving objects no matter if moving objects increases or not.

5 Conclusions

In this paper, we proposed an efficient rain recovery algorithm based on LIDAR. Previous methods proposed a method which shows state of-the-art performance in removing rain in highly dynamic video scenes. However, the method is not suitable for real-time performance when moving objects increased. We put forward an efficient method to detect moving objects for rain removal. We first identified the location of moving objects by LIDAR. Then, we use several models to map LIDAR data to 2D image plan. We separate moving objects from whole images and get rain map without moving objects. Finally we employ edge-preserving bilateral filter to remove rain pixels on moving objects, and use Gaussian filter temporally on three frames to remove rain pixels in the static background.

Experiments show that the proposed method is able to remove rain from scenes taken from dynamic scenes very effectively. The performance of rain removal still has much room for improvement. A future research issue is to extend the proposed algorithm to low resolutions images.

Acknowledgement. This work was supported by a grant from National Natural Science Foundation of China (NSFC, No. 61504032).

References

1. Narasimhan, S.G., Nayar, S.K.: Vision and the atmosphere. Int. J. Comput. Vis. **48**(3), 233–254 (2002)
2. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 1. IEEE (2004)
3. Barnum, P., Kanade, T., Narasimhan, S.: Spatio-temporal frequency analysis for removing rain and snow from videos. In: Proceedings of the First International Workshop on Photometric Analysis for Computer Vision-PACV 2007. INRIA (2007)

4. Barnum, P.C., Narasimhan, S., Kanade, T.: Analysis of rain and snow in frequency space. Int. J. Comput. Vis. **86**(2–3), 256–274 (2010)
5. Kim, J.-H., et al.: Single-image deraining using an adaptive nonlocal means filter. In: 2013 20th IEEE International Conference on Image Processing (ICIP). IEEE (2013)
6. Liu, P., Xu, J., Liu, J., et al.: A rain removal method using chromatic property for image sequence. In: 11th Joint International Conference on Information Sciences. Atlantis Press (2008)
7. Chen, J., Chau, L.P.: A rain pixel recovery algorithm for videos with highly dynamic scenes. IEEE Trans. Image Process. **23**(3), 1097–1104 (2014)
8. Ding, X., Chen, L., Zheng, X., et al.: Single image rain and snow removal via guided L0 smoothing filter. Multimedia Tools Appl. **75**, 2697–2712 (2015)
9. Shen, M., Xue, P.: A fast algorithm for rain detection and removal from videos. In: 2011 IEEE International Conference on Multimedia and Expo (ICME). IEEE (2011)
10. Tan, C.-H., Chen, J., Chau, L.-P.: Dynamic scene rain removal for moving cameras. In: 2014 19th International Conference on Digital Signal Processing (DSP). IEEE (2014)
11. Santhaseelan, V., Asari, V.K.: Utilizing local phase information to remove rain from video. Int. J. Comput. Vis. **112**(1), 71–89 (2015)
12. Arras, K.O., Mozos, ó.M., Burgard, W.: Using boosted features for the detection of people in 2D range data. In: 2007 IEEE International Conference on Robotics and Automation. IEEE (2007)
13. Kurnianggoro, L., Hernandez, D.C., Jo, K.-H.: Camera and laser range finder fusion for real-time car detection. In: 40th Annual Conference of the IEEE Industrial Electronics Society, IECON 2014. IEEE (2014)