Adherent Raindrop Modeling, Detection and Removal in Video

Shaodi You, *Student Member, IEEE*, Robby T. Tan, *Member, IEEE*, Rei Kawakami, *Member, IEEE*, Yasuhiro Mukaigawa, *Member, IEEE*, and Katsushi Ikeuchi, *Fellow, IEEE*

Abstract—Raindrops adhered to a windscreen or window glass can significantly degrade the visibility of a scene. Modeling, detecting and removing raindrops will, therefore, benefit many computer vision applications, particularly outdoor surveillance systems and intelligent vehicle systems. In this paper, a method that automatically detects and removes adherent raindrops is introduced. The core idea is to exploit the local spatio-temporal derivatives of raindrops. To accomplish the idea, we first model adherent raindrops using law of physics, and detect raindrops based on these models in combination with motion and intensity temporal derivatives of the input video. Having detected the raindrops, we remove them and restore the images based on an analysis that some areas of raindrops completely occludes the scene, and some other areas occlude only partially. For partially occluding areas, we restore them by retrieving as much as possible information of the scene, namely, by solving a blending function on the detected partially occluding areas using the temporal intensity derivative. For completely occluding areas, we recover them by using a video completion technique. Experimental results using various real videos show the effectiveness of our method.

Index Terms—Outdoor vision, rainy scenes, raindrop detection, raindrop removal

1 Introduction

Outdoor vision systems employed for various tasks such as navigation, data collection and surveillance, can be adversely affected by bad weather conditions such as rain, haze and snow. In a rainy day, raindrops inevitably adhered to windscreens, camera lenses, or protecting shields. These adherent raindrops occlude and deform some image areas, causing the performances of many algorithms in the vision systems such as feature detection, tracking, stereo correspondence, etc., to be significantly degraded. This problem occurs particularly for vision systems that use a hand-held camera or a top-mounted vehicle sensor where no wipers can be used.

Identifying adherent raindrops from images can be problematic due to various reasons (different shapes, blur, glare, etc.) as shown in Fig. 1. To address the problems, we analyze the appearance of adherent raindrops from their local spatio-temporal derivatives. A clear, non-blurred adherent raindrop works like a fish-eye lens and significantly contracts the image of a scene. Consequently, the motion inside raindrops is distinctively slower than the motion of non-raindrops. Besides, unlike clear raindrops, blurred raindrops are mixtures of light rays originated from various points in the background scene, causing the intensity temporal derivatives of blurred raindrops to be considerably smaller than those of non-raindrops. These two clues are the key to our detection method. Relying on them we propose a pixel based detection method, which is generally applicable to handle any shape and size of raindrops. Fig. 1e shows a result of our detection method.

Having detected the raindrops and analyzed the image formation of raindrops, we found that some areas of a raindrop completely occlude the scene behind, and the remaining areas occlude only partially. For partially occluding areas, we restore their appearance by retrieving as much as possible information of the scene, namely, by solving a blending function on the detected areas using the intensity change over time. For completely occluding areas, we recover them by using a video completion technique. Fig. 1f shows a result of our raindrop removal method.

Our contributions in this paper includes the introduction of:

- Adherent raindrop models and analysis, which use the derivative properties and involve only few parameters.
- A novel pixel-based detection method based on motion and intensity change.
- A relatively fast adherent raindrop removal method, which exploits a blending function of partially occluded areas.

Note that, like most methods, our method is subject to a few assumptions. We assume raindrops are static during the detection process, which we call quasi-static assumption. In our experiments, we consider raindrops to be quasi-static when their motion is less than 4 pixels/second with raindrop size around 40 pixels (we discuss this issue further in Section 7.1). To detect a newly appearing raindrop, our method requires consecutive video frames to integrate the

- *S. You and R. Kawakami are with The University of Tokyo, Tokyo, Japan. E-mail: youshaodi@gmail.com, rei@nae-lab.org.*
- *R. T. Tan is with the Yale-NUS College and National University of Singapore (NUS), Singapore. E-mail: robby.tan@yale-nus.edu.sg.*
- *Y. Mukaigawa is with the Nara Institute of Science and Technology, Japan. E-mail: mukaigawa@is.naist.jp.*
- *K. Ikeuchi is with the Microsoft Research Asia, China. E-mail: ki@cvl.iis.u-tokyo.ac.jp.*

Manuscript received 10 Nov. 2014; revised 21 Sept. 2015; accepted 5 Oct. 2015. Date of publication 15 Oct. 2015; date of current version 11 Aug. 2016. Recommended for acceptance by J. Jia.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2491937

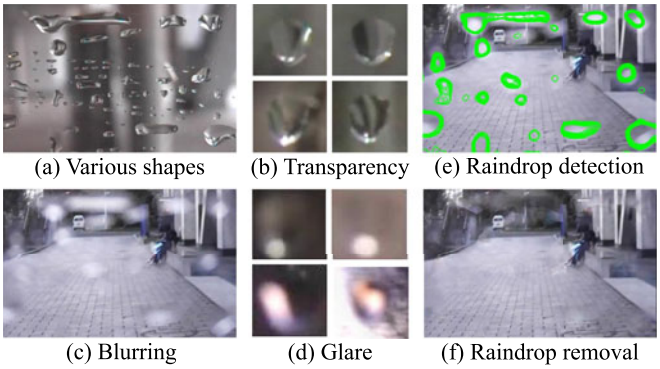


Fig. 1. (a-e) The various appearances of raindrops. (e-f) The detection and removal result by our method.

features. In our experiments, for a 24 fps camera, we use 100 consecutive frames as default, and accuracy will drop when less than 20 frames are used. Also, there is a trade-off between the accuracy and efficiency. While our method achieves optimal accuracy when using both the intensity-change and motion based features, only intensity-change based feature can work in real time.

2 Related Work

Removing the influence of haze, mist, fog (e.g., [1], [2], [3], [4]), rain and snow (e.g., [5], [6]) have been well exploited. Dealing with rain, Garg and Nayar model rain streaks [7], and devise algorithms to detect and remove them [6], [8]. Later, Barnum et al. [5] propose a method to detect and remove both rain and snow. Single-image based methods are proposed by Kang et al. [9] and Chen et al. [10]. Unfortunately, applying these methods to handle adherent raindrops is not possible, since the physics and appearance of falling raindrops are significantly different from those of adherent raindrops.

2.1 Sensor/Lens Dust Removal

Sensor dust removal is to some extent a related topic to raindrop detections. Willson et al. [11] give a detailed analysis on the imagery model with dust adhered to the lens. Dust blocks light reflected from objects and scatter/reflect light coming from the environment. The former is called a dark dust artifact, and the latter a bright dust artifact. Zhou and Lin [12] propose method to detect and remove small dark dust artifacts. Gu et al. [13] extend the solution to sufficiently blurred thin occluders. Although adherent raindrops can be considered as a kind of sensor dust, existing sensor dust removal methods cannot handle adherent raindrops, since raindrops can be large and are not as blurred as dust. Moreover, raindrop appearance significantly more varies than dust appearance.

2.2 Adherent Raindrop Detection and Removal

A few methods for detecting adherent raindrops have been proposed. Roser et al. attempt to model the shape of adherent raindrops by a sphere crown [14], and later, Bezier curves [15]. These models, however, are insufficient, since a sphere crown and Bezier curves can cover only a small portion of raindrop shapes. Kurihata et al. [16] and later Eigen et al. [17] approach the problem through machine learning. However, as shown in Figs. 1a, 1b, 1c, and 1d, collecting the training images for all various shapes, environment, illumination and blur are considerably challenging. Both of the methods are limited to detect small, clear and quasi-round rain spots. Yamashita et al. propose a detection and removal method for videos taken by stereo [18] and pan-tilt [19] cameras. The methods utilize specific constraints from those cameras and are thus inapplicable for a single camera. Hara et al. [20] propose a method to remove glare caused by adherent raindrops by using a specifically designed optical shutter. As for raindrop removal, Roser and Geiger [14] address it using image registration, and Yamashita et al. [18], [19] utilize position and motion constraints from specific cameras.

2.3 Video Completion

Video completion has been intensively exploited by computer vision researchers. Only those methods work with large spatio-temporal missing areas can be used to remove detected adherent raindrops. Wexler et al. [21] propose an exemplar based inpainting method by assuming the missing data reappears elsewhere in the video. Jia et al. [22] exploit video completion by separating static background from moving foreground, and later [23] exploit video completion under cyclic motion. Sapiro and Bertalmio [24] complete the video under constrained camera motion. Shiratori et al. [25] and Liu et al. [26] first calculate the motion of the missing areas, and then complete the video according to the motion. Unfortunately, outdoor environments are too complex to satisfy static background, cyclic motion, constrained camera motion, etc. Therefore, we use cues from our adherent raindrop modeling to help the removal.

3 Clear Raindrop Modeling

Raindrop appearance is highly depending on the camera intrinsic parameters. In this section, we first assume a pinhole camera and non-blurred raindrops, and explore raindrop imagery properties in this condition. Based on our analysis in this section, we model blurred raindrops in the next section. Unlike the previous methods [15], [16], [18], [19], [20], which try to model each raindrop as a unit object, we model raindrops locally from the derivative properties that involve only few parameters.

3.1 Physical Attributes

From Figs. 1a and 1b, we can see that adherent raindrops have various shapes and sizes, and their appearance is dependent on the environment.

Size. Unlike estimating the size of airborne raindrops, which is mentioned in the work of Garg et al. [6], estimating the size of adherent raindrops is not trivial. Since it depends on the gravity, water-water surface tensor, water-adhering-surface tensor and many other parameters.

Fortunately, it is possible to set an upper bound of the size by using few parameters. As illustrated in Fig. 2a, to prevent raindrops from sliding down, both the two-phase point (water-air) and three-phase points (water-air-material), the surface tensor should balance the pressure. This also prevents the water drop from breaking down. Although estimating the balance and upper boundary of

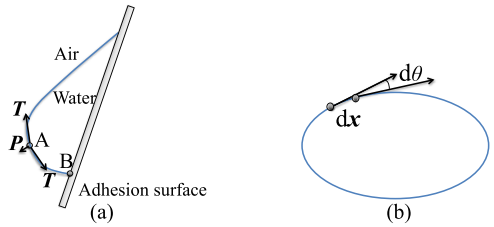


Fig. 2. (a) Balance at a raindrop surface. A denotes a two-phase point. B denotes a three-phase point. **T** denotes a surface tension, and **P** for pressure. At two-phase point A, surface tension **T** and pressure **P** are balanced. Three-phase point is an intersection of water, air and glass, while two-phase point is an intersection between air-water. (b) Change of the angle of tangent along a raindrop boundary.

the three phase points is intractable due to the unknown parameters of the material, estimating the balance and upper bound of two-phase point has been studied by physicists, and can be used to derive an upper bound of raindrop size, i.e., 5 mm [27].

Shape. Although most existing methods assume the shape of raindrops to be circle or ellipse, the real raindrop shape varies in a large range. Despite this, however, we can still find some regular patterns due to the surface tension. Raindrop boundaries are smooth and raindrops are convex in most cases. Hence, we can quantitatively characterize raindrop shape using two features: shape smoothness and roundness. As illustrated in Fig. 2b, given a raindrop area on the image plane, denoted as R, we can integrate the change of the tangent angle along the boundary. The integration is denoted as $\mathcal{S}(R)$:

$$\mathcal{S}(R) = \oint_{\boldsymbol{x}\in\partial R} |\mathrm{d}\theta(\boldsymbol{x})|, \tag{1}$$

where ∂R is the boundary of the raindrop, and $\boldsymbol{x} = (x, y)$ is the the $2D$ coordinates on the image plane. For convex shape, $\mathcal{S}(R) \equiv 2\pi$. For non-convex or zig-zag shape, the smoothness will be greater than 2π. Fig. 3 shows some examples.

Roundness, denoted as $\mathcal{O}(R)$, is the area of the shape divided by the square of its perimeter:

$$\mathcal{O}(R) = \frac{\boldsymbol{x}\in R \mathrm{d}x\mathrm{d}y}{\left(\oint_{\boldsymbol{x}\in\partial R} |\mathrm{d}\boldsymbol{x}|\right)^2}. \tag{2}$$

A rounder shape has a larger roundness value and a perfect circle has the maximum roundness value: $\frac{\pi r^2}{(2\pi r)^2} = \frac{1}{4\pi} = 0.080$. Fig. 3 shows some examples. Both the smoothness and roundness are invariant to scaling and rotation. Unlike our previous method [28], which used the roundness, our current method employs smoothness. This is because the computational complexity of roundness is $O(n^2)$ while smoothness is $O(n)$.

Dynamics. In rainy scenes, some raindrops might slide sporadically. The sliding probability and speed depend on a few attributes, such as, surface tension coefficients, surface

Shape						
Smoothness	2π(6.28)	2π(6.28)	3π(9.42)	11.10	9.41	54.15
Roundness	1/4π(0.080)	0.075	0.050	0.029	0.058	0.016

Fig. 3. Smoothness and roundness of some shapes.

TABLE 1
Raindrop Dynamic of Scenes in Fig. 19

Data	Camera speed	Camera shaking	Max raindrop speed observed
Experiment 1 - 4	5 km/h	yes	0.48 pixel/s
Car-mounted	30 km/h	yes	0.01 pixel/s
Surveillance	0	no	0.40 pixel/s

tilt, wind, raining intensity, raindrop size, etc. An exact modeling of raindrop dynamics is intractable. Fortunately, in light rainy scenes, we find it reasonable to assume most raindrops are quasi-static (in our experiments, we consider raindrops are quasi-static when its motion is less than 4 pixels/second with raindrop size around 40 pixels). We observed the motion of real adherent raindrops in scenes in Fig. 19. Focusing on a raindrop, we compared the current location with the location one minute later and convert it to speed (pixel per second). Table 1 lists the maximum speed observed in each scene. We quantitatively evaluate the tolerance of raindrop dynamics in Section 7.

3.2 Clear Raindrop Imagery

As shown in Fig. 4a, each raindrop is a contracted image of the environment, as if it is taken from a fish-eye-lens camera. The numerical values indicated in Fig. 4c are the scale ratios between the original image and the image inside the raindrops calculated from the black and white patterns. The scale ratios are around 20 to 30, meaning that the motion observed inside the raindrops will be $1/30$ to $1/20$ slower than the other areas in the image.

In this section, we consider the camera as a pin-hole camera, where both the raindrops and environment are not blurred. This assumption is necessary in our analysis to use motion as a feature of raindrop detection. As illustrated in Fig. 5, there are point pairs on the image plane that represent exactly the same environment points. One of them is coming from a light ray that directly travels to the image plane, denoted as P_e, and the other is a light ray traveling through a raindrop, denoted as P_r.

Let us consider the relation between $P_e = (x, y)$ and $P_r = (u, v)$ on the image plane. As shown in Fig. 4b, there is a 2D to 2D mapping $\boldsymbol{\varphi}$ from (u, v) to (x, y):

$$(x, y) = \boldsymbol{\varphi}(u, v) = (\varphi^1(u, v), \varphi^2(u, v)). \tag{3}$$

Considering the refraction model in Fig. 5, to know the function $\boldsymbol{\varphi}$, we need: 1) the position and shape information of the raindrop, 2) the camera inner parameters, and 3) the background depth information. Our analysis, however, looks into the spatial derivative properties and therefore can avoid obtaining $\boldsymbol{\varphi}$ explicitly.

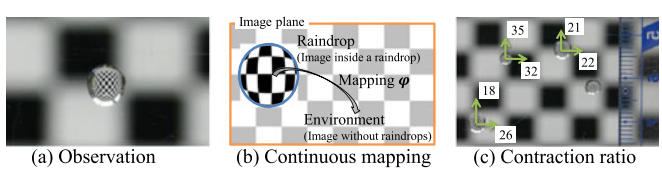


Fig. 4. (a) A raindrop is a contracted image of the environment. (b) On the image plane, there is a smooth mapping $\boldsymbol{\varphi}$ starting from the raindrop into the environment. (c) The contraction ratios from the environment to a raindrop are significant.

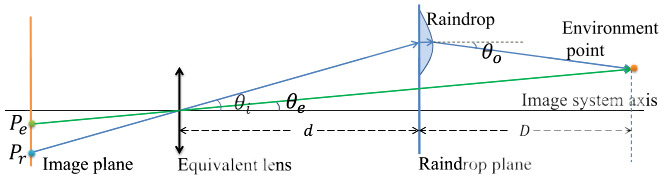


Fig. 5. The refraction model of two points on an image plane (P_e and P_r) that are originated from the same point in the environment. There are two refractions on the light path passing a raindrop. The camera lens cover or protecting shield is assumed to be a thin plane and thus can be neglected.

3.3 Spatial Derivative of Clear Raindrop

The scalar contraction ratio $\mathcal{E}_\varphi$ is the derivative of $\boldsymbol{\varphi}$ with respect to u and v in the direction $(\delta u, \delta v)$:

$$\mathcal{E}_\varphi(u, v, \delta u, \delta v) = \lim_{(\delta u, \delta v)\to 0} \frac{\|\boldsymbol{\varphi}(u+\delta u, v+\delta v) - \boldsymbol{\varphi}(u, v)\|}{\|(u+\delta u, v+\delta v) - (u, v)\|}. \quad (4)$$

Unlike obtaining an explicit expression of $\boldsymbol{\varphi}$, obtaining an upper bound of $\mathcal{E}_\varphi$ needs only the upper bound of raindrop size and the lower bound of the distance between a raindrop and the camera. The raindrop upper bound has been discussed in Section 3.1. The raindrop-to-camera distance lower bound depends on camera settings. In our experiment, we normally found $d < 200$ mm.

Using the imaging model in Fig. 5, for outdoor environment we can prove that, for any (u, v) and any $(\delta u, \delta v)$:

$$\mathcal{E}_\varphi > 10 \gg 1. \quad (5)$$

The proof is provided in Appendices A and B in the supplementary material (which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2491937.

3.4 Detecting Raindrops Using Optic Flow

Raindrops not only contract the images of the environment, but also contract the motion between images. Consider a point pair $P_e(t_1)$ and $P_r(t_1)$, and their correspondence in next frame $P_e(t_2)$, $P_r(t_2)$. We denote the motion between $P_e(t_1)$ and $P_e(t_2)$ as $M(P_e) = P_e(t_2) - P_e(t_1)$ and the motion between $P_r(t_1)$, $P_r(t_2)$ as $M(P_r) = P_r(t_2) - P_r(t_1)$. Based on Eqs. (4) and (5), and using the integral version triangle inequality, we have:

$$\frac{\|M(P_e)\|}{\|M(P_r)\|} = \frac{\|P_e(t_2) - P_e(t_1)\|}{\|P_r(t_2) - P_r(t_1)\|} = \frac{\|\boldsymbol{\varphi}(P_r(t_2)) - \boldsymbol{\varphi}(P_r(t_1))\|}{\|P_r(t_2) - P_r(t_1)\|} \geq \mathcal{E}_\varphi. \quad (6)$$

This means the motion in the raindrop area has been significantly contracted, which gives us an idea to use optic flow as a feature to identify raindrops. Figs. 8a and 8b show examples of optic flow on raindrop and non-raindrop areas. As we can see, the motion intensity in the raindrop area is significantly smaller than that in the non-raindrop area. Based on this, we will develop a detection algorithm in Section 5.

4 Blurred Raindrop Modeling

In contrast to raindrop imagery with a pin-hole camera, for a normal lens camera, when the camera focuses on the environment scene, raindrops will be blurred. To handle this, we model blurred raindrops, and theoretically derive the temporal property of raindrop pixels. Based on this property, we propose a pixel-wise raindrop detection feature: intensity change.

4.1 Blurred Raindrop

As illustrated in Fig. 6, the appearance of a pixel on an image plane depends on the collection of light rays. These rays can come from light emitted directly by an environment point (Fig. 6 A), light refracted from a raindrop (Fig. 6 B), and a mixture of environment light and raindrop light (Fig. 6 C). We denote the light intensity collected by pixel (x, y) as $I(x, y)$. We also denote the light intensity formed by an environment point that intersects with the line of sight as $I_e(x, y)$; and, the light intensity reached (x, y) passing through a raindrop as $I_r(x, y)$. Hence, pixel (x, y) collecting light from both the raindrop and the environment can be described as:

$$I(x, y) = (1 - \alpha) I_e(x, y) + \alpha I_r(x, y), \quad (7)$$

where α is the proportion of the light path covered by a raindrop, as depicted in Figs. 6b and 6b'.

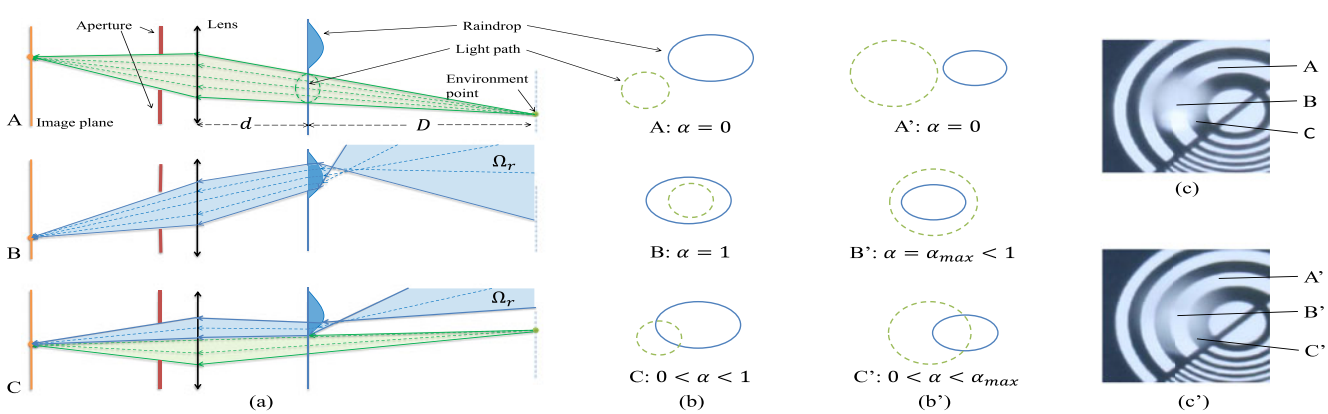


Fig. 6. Rows: The appearance and model of pixels on an image plane collecting light from A: environment, B: raindrop, C: both. Columns: (a) The light path model. Green light: the light coming from environment point; Blue light: the light refracted by a raindrop. (b) Raindrop-plane-cut of the model in (a). Green circle: the area of light collected. Blue circle: the raindrop. α: percentage of light collected from the raindrop. (b') Light path coverage when the raindrop is small. (c) The appearance of the 3 situations in (b). (c') The appearance of the 3 situations in (b').

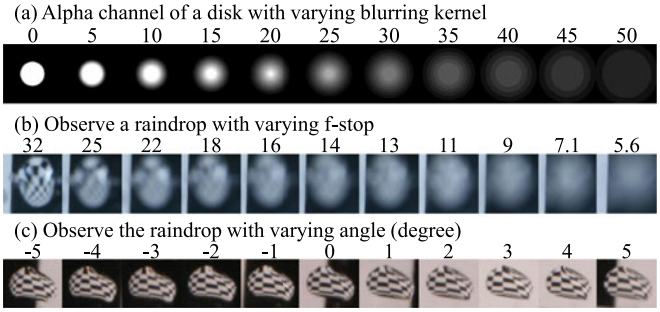


Fig. 7. a. α-channel of a disk with various blur kernels. b. Raindrops with varying f-stop. c. Raindrops with varying angles. Raindrop appearance is highly directional.

Blending coefficient α is determined by the area of the light path and the raindrop. Using the model in Fig. 6, the diameter of the light path on the raindrop plane can be estimated using:

$$\frac{D}{D+d}A = \frac{D}{D+d}\frac{f}{N}, \tag{8}$$

where $\frac{f}{N}$, called the f-stop, is the convention expression for the camera aperture setting.

A more convenient way to express α on the image plane is to use a blur kernel. First, as illustrated in Fig. 7a, α is either 0 or 1. We denote the blending coefficient of clear raindrops as α_c. Hence, α of blurred raindrops can be calculated by convoluting α_c with a disk kernel, where the diameter of the kernel is given by:

$$\ell = \frac{(D-d)}{(D-f)}\frac{f}{d}A, \tag{9}$$

which is proportional of the aperture size A. The derivation of Eq. (9) can be found in the literature of depth from defocus [29]. Consequently, if a raindrop is significantly blurred, the blending coefficient is smaller than 1. In such a case, the raindrop cannot totally occlude the environment. Fig. 6c' shows an example. Fig. 7b shows real blurred raindrops, and Fig. 7c show raindrop appearance, which is highly directional.

4.2 Temporal Derivative of Blurred Raindrop

We avoid estimating the exact appearance of blurred raindrops due to its intractability. Instead, we explore the temporal derivative features. In consecutive frames, we observe that the intensity of blurred pixels (cases B and C) does not change as distinctive as that of environment pixels (case A). To analyze this property, let us look into the intensity temporal derivatives of blurred pixels. Referring to Figs. 6a, case B and C, light collected from a raindrop is actually originated from a large area in the environment. We denote the area as $\Omega_r(x,y)$. At time t, we expand $I_r(x,y)$ in Eq. (7) as:

$$I_r(x,y,t) = \sum_{(z,w)\in\Omega_r(x,y)} W(z,w)I_e(z,w,t), \tag{10}$$

where $W(z,w)$ is the weight coefficient determined by the raindrop geometry. $W(z,w)$ and $\Omega_r(x,y)$ can be considered constant in a short period of time.

If we take the difference of intensity between time t_1 and t_2 in Eq. (10), and consider the triangle inequality, we have:

$$\begin{aligned}&|I_r(x,y,t_1) - I_r(x,y,t_2)|\\ &\le \sum_{(z,w)\in\Omega_r(x,y)} W(z,w)|I_e(z,w,t_1) - I_e(z,w,t_2)|.\end{aligned} \tag{11}$$

Here, by taking into account Eq. (5), we know the area ratio is more than one hundred when the raindrops clearly appear, namely,

$$\mathcal{E}_\varphi^2 > 100 \gg 1. \tag{12}$$

Notice that φ is not conformal, and the proof is provided in the supplementary material (available online). For blurred raindrops, the area ratio further expands. Referring to the model in Fig. 6, in addition to the expanded area caused by a raindrop, the out-of-focus blurring also causes the area to expand. Thus, we can consider $\Omega_r(x,y)$ to be a sufficiently large area. According to the law of large number, we can have:

$$E|I_r(x,y,t_1) - I_r(x,y,t_2)| \ll E|I_e(x,y,t_1) - I_e(x,y,t_2)|, \tag{13}$$

where E denotes the expectation.

Since the temporal derivatives work as a high pass filter, we may also consider Eq. (13) in a frequency domain, where the temporal high frequency component of a raindrop is significantly smaller than those of the environment, described as:

$$\mathcal{I}_r(x,y,\omega) \ll \mathcal{I}_e(x,y,\omega), \omega = \omega_{th}, \omega_{th}+1, \ldots, N \tag{14}$$

where $\mathcal{I}$ is the Fourier transform of sequence $I(x,y,t), t = t_1, t_2, \ldots, N$, and ω_{th} is currently an undetermined threshold for the high frequency.

4.3 Detecting Raindrops Using Intensity Change

By considering Eqs. (13) and (7), the temporal difference for $I(x,y,t)$ will be small when α is large:

$$\begin{aligned}&E|I(x,y,t_1) - I(x,y,t_2)|\\ &= \alpha(x,y)E|I_r(x,y,t_1) - I_r(x,y,t_2)|\\ &\quad + (1-\alpha(x,y))E|I_e(x,y,t_1) - I_e(x,y,t_2)|\\ &\approx (1-\alpha(x,y))E|I_e(x,y,t_1) - I_e(x,y,t_2)|.\end{aligned} \tag{15}$$

Therefore, we can use the temporal intensity change as a feature to detect raindrops. Fig. 9 shows an example.

4.4 Effects of Glare

As illustrated in Fig. 1d, a raindrop refracts bright rays of light from the environment, and generate glare. This phenomenon does not affect the derivative properties described in the previous sections. Since, while glare emits high intensity light, and the spatial derivatives are independent from light intensity. Moreover, the appearance of glare in video is temporally smooth, i.e., the intensity monotonically increases until it saturates, and then it monotonically decreases until the glare fades out. The temporal derivatives of this smooth change is still small, and thus does not affect the derivative properties.

5 Raindrop Detection

5.1 Feature Extraction

We generate two features for the detection: a motion feature (OF) which is based on the analysis of clear images in Section 3; and the intensity change feature (IC) which is based

(a) Image sequence (b) Inter frame SIFT flow (c) Summation of (b) over 100 frames

Fig. 8. The accumulated optic flow as a feature.

(a) Image sequence (b) Inter frame intensity change (c) Summation of (b) over 100 frames

Fig. 9. The accumulated intensity changes as a feature.

on analysis blurred images in Section 4. We calculate the motion feature using a robust optic flow algorithm, e.g., SIFT-flow [30], which is shown in Fig. 8b, and calculate the intensity change feature using $|I(x, y, t_1) - I(x, y, t_2)|$, which is shown in Fig. 9b.

In the examples, the two features are calculated using only two consecutive frames. In fact, the features will be more informative if they are calculated using data accumulated over more frames. Statistically the more frames used, the more descriptive the features are. Unfortunately, raindrop positions can shift over a certain period of time, making the detection using long frames erroneous. In our observation, with moderate wind, raindrops can be considered static over a few seconds. As default, we calculate over 100 frames which is about 4 seconds for the frame rate of 24 fps. Figs. 8c and 9c show examples of the two accumulated features.

We employ both features to have optimal accuracy. If time is a concern, however, we can use only intensity change.

5.2 Refined Detection

Having calculated the features, we use level sets [31] to identify raindrops. First, a convolution with Gaussian ($\sigma = 2$ pixels by default) is employed to reduce noise. Then, level sets are calculated, as illustrated in Fig. 10. Specifically, for the normalized 2D feature, we calculate the level-sets range from -2 to 2 with the step 0.05.

The following criteria are applied further for determining raindrop areas:

1) Feature threshold. As analyzed previously, raindrop areas should have smaller feature values. Hence, we normalized the accumulated feature with the mean value 0 and variance 1. In our experiment, those pixels with feature values less than -0.7 are considered to be raindrop pixels.
2) Smoothness. As analyzed in Section 3.1, (Eq. (1)), raindrop contours usually have a smoothness value at 2π. Thus, we set the threshold for smoothness as 2.5π.

Note that, unlike [28], we do not utilize the closure explicitly, since it is already represented by the smoothness, which cannot be defined to non-closed lines. We also do not use size, as it varies significantly. Fig. 10 shows the detecton pipeline. For each detection, we accumulate the feature for the past 4 seconds and compute the level sets to detect raindrops. The overall detection algorithm is described in Algorithm. 1.

Algorithm 1. Raindrop Detection

```
Default parameter settings
  Video: 1080 × 720, 24fps
  Feature accumulating period: 4s(96frames)
  Number of detection phases: 2 per second
  Feature threshold:
    −0.7 for intensity change
    −0.4 for optic flow
  Smoothness threshold: 2.5π
while (not video end)
  compute the feature for new frames
  Accumulate the feature in specified period
  if (Detection phase)
    reduce noise of feature, σ = 2 Gaussian filter
    normalize feature to average = 0, variance = 1
    calculate level sets of the feature image.
    for (all contours)
      if (feature < threshold
        & smoothness < threshold)
        This contour circles a raindrop
      end
    end
    Displace result for current detection phase
  end
end
```

5.3 Real Time Detection

The detection method can work in real time if we use only the intensity change as the feature. Although this real time performance is subject to a delay for a few second (4 seconds in our experiments) to detect newly appearing raindrops, since we need to collect the features in a few consecutive frames. We run our program on a 3.1 GHz CPU and Matlab with no parallelization. The video is 1,280 × 720, 24 fps. Accumulating the feature takes 0.0086 s per frame, which is 0.10 s for 12 frames. Gaussian filter takes 0.04 s. The level sets takes 0.22 s. Selecting contours takes 0.06 s. The overall computing time for each detection phase is 0.42 s.

6 Raindrop Removal and Image Restoration

Existing methods try to restore the entire areas of the detected raindrops by considering them as solid occluders [14], [19]. In contrast, we try to restore the raindrop areas

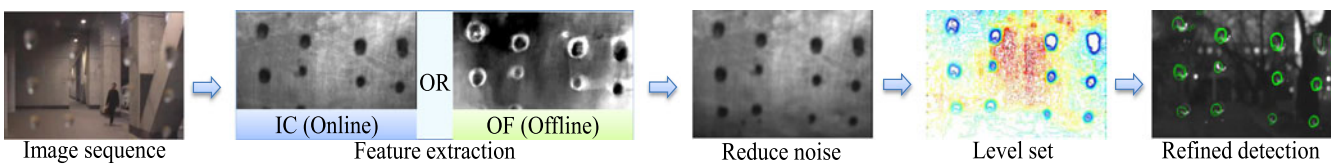


Fig. 10. The detection pipeline. Our method can work in real time if using only the intensity change.

from the available information about the environment whenever possible. Based on Eq. (7), we know that some areas of a raindrop completely occludes the scene behind, however the rest occludes only partially. For partially occluding areas, we restore them by retrieving as much as possible information of the scene, and for completely occluding areas, we recover them by using a video completion technique.

Algorithm 2. Raindrop Removal

if (default)
  $N = 100$, $\omega_{th} = 0.05N$, $\Delta x = \Delta y = \pm 1$pixel
  $th1 = 250$, $th2 = 40$
end
Load N continuous frames
Calculate $\alpha(x, y)$ for each pixel $I(x, y, \cdot)$.
if ($max(I(x, y, \cdot)) > th1$ & $\alpha(x, y) > 0$) {(x, y) is glare}
for (non-glare pixels and $0 < \alpha(x, y) < 0.9$)
  for ($(R; G; B)$ channel separately)
    while ($\exists$ pixel unprocessed)
      Find pixel with smallest α ($I(x, y, \cdot)$)
      Find neighbors of (x, y) in $(x + \Delta x, y + \Delta y)$
      Remove neighbors (intensity difference $> th2$)
      Do DCT: $\mathcal{I}(x, y, \omega) = I(x, y, t)$
      $\mathcal{I}(x, y, \omega_{th} : N) = \frac{1}{1-\alpha(x,y)}\mathcal{I}(x, y, \omega_{th} : N)$
      $\mathcal{I}(x, y, 1 : \omega_{th}) = mean(\mathcal{I}(x + \Delta x, y + \Delta y, 1 : \omega_{th}))$
      Do inverse-DCT
    end
  end
end
Repair the remaining areas using an inpainting method.

6.1 Restoration

A blurred image can be recovered by estimating $I_e(x, y)$ in Eq. (7), in the condition that the blending value is moderate, i.e., $\alpha(x, y) < 1$. To do this, we first have to calculate α in Eq. (7). Note that, based on our previous detection phase, the positions and shapes of raindrops on the image plane are known. Using the out-of-focus blur model in Fig. 6a, the diameter ℓ of the equivalent light path area on the image plane is given by:

$$\ell = \frac{(D - d)}{(D - f)} \frac{f^2}{Od}, \tag{16}$$

where f is the focal length. O is the relative aperture size (also called f-stop) which can be found in the camera setting. D can be assumed to be infinite, and d is estimated empirically (we assumed constant throughout our experiments). The derivation of Eq. (16) can be found in the literature of depth from defocus [29]. Thus, a circle centered at (x, y) with diameter ℓ on the image plane can be drawn, as shown in Figs. 6b and 6b'. The blending coefficient $\alpha(x, y)$ is the proportion of the circle that overlaps with the raindrop.

Having obtained α, we recover I_e from the frequency domain. According to Eq. (14), the high frequency component of raindrop I_r is negligible. Thus, for frequency higher than a threshold ω_{th}, we have:

$$\mathcal{I}_e(x, y, \omega) = \frac{1}{1 - \alpha(x, y)}\mathcal{I}(x, y, \omega), \quad \omega > \omega_{th}, \tag{17}$$

where $\mathcal{I}(x, y, \omega)$ is the Discrete Cosine Fourier Transform (DCT) of $I(x, y, t)$ on N consecutive frames. ω_{th} is set as $0.05N$ as default. As for the low frequency component, we replace it with the mean of its spatial neighborhood (from only the non-raindrop pixels or the already restored pixels):

$$\mathcal{I}_e(x, y, \omega) = \text{mean}(\mathcal{I}(x + \Delta x, y + \Delta y, \omega)), \ \omega \leq \omega_{th}, \tag{18}$$

where $(x + \Delta x, y + \Delta y), \Delta x, \Delta y \leq 1$ pixel are spatial neighborhood of (x, y). When averaging, we exclude neighboring pixels that have intensity differences larger than 40 (in 8-bit RGB value). By combining Eqs. (17) and (18), and performing inverse-DCT, we recover $I_e(x, y, t)$.

6.2 Video Completion

Having restored the partially occluding raindrop pixels, there are two types of remaining areas to complete:

- When α is close or equal to 1.0, I_e will be too scarce to be restored, as shown in Eq. (17). Because of this, we do not restore pixels with $\alpha > 0.9$.
- When there is glare, the light component from raindrop will be too strong and therefore saturated.

For those areas, we adopt Wexler et al.'s [21] space-time video completion method. As discussed in the related work, the method [21] only assumes that missing data reappears elsewhere in the video, which is most likely to be satisfied in outdoor scenes. The overall algorithm of our proposed raindrop removal algorithm is shown in Algorithm 2.

7 Experiments and Applications

We conduct quantitative experiments to measure the accuracy and general applicability of our detection and removal method. To show the benefits of our method, we include two real applications of our method on motion estimation and structure from motion.

7.1 Quantitative Analysis on Detection

We evaluate how raindrop size, blur, motion, scene complexity affect the detection using synthetic data, and estimated the optimal parameters. We also conduct the detection on various real scenes and compare the performance with that of the state-of-art methods. We use the precision-recall curve for our evaluation, where precision is defined as the number of the correct detection divided by the number of all the detection, and recall as the number of correct detection divided by the number of the detectable raindrops.

Raindrop size and blur. As discussed in Section 3.2, our detection method is based on the fact that raindrops behave like a fish-eye lens and contract the environment. Obviously, a larger raindrop contracts less than a smaller raindrop does. Hence, raindrop physical size, which is limited by the raindrop tensor, affects the contraction ratio. Moreover, since our input is an image, the distance between the raindrop and the camera lens also affect the contraction ratio.

When raindrops are close to the lens, we need to consider the effect of out-of-focus blurring. Since, the closer to the lens, the more blur the raindrop is, implying lesser visibility. In our experiment, we explore how raindrop size and blur affect the detection accuracy. As illustrated in Fig. 11, we generate synthetic raindrops with fixed positions, but with various

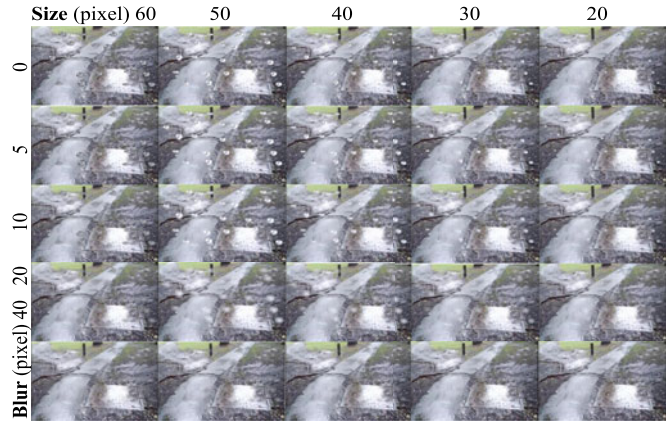


Fig. 11. Synthetic raindrops with various size and blur levels. The image size is 720×480, raindrop size (long axis) varies from 20 to 60 pixels, and the radius of the disk-blur-kernel varies from 0 to 40 pixels.

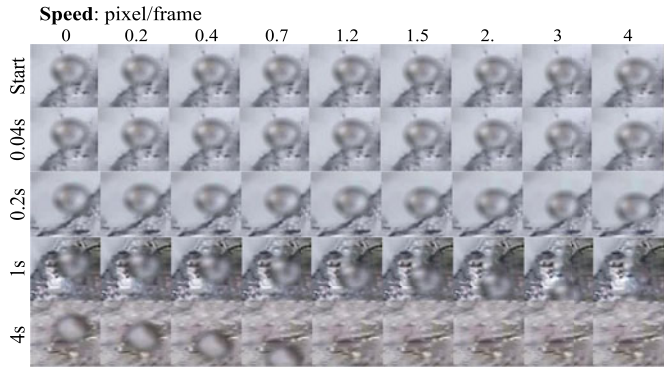


Fig. 13. Appearance of synthetic moving raindrops. The raindrops are around 40 pixels of size and are blurred with a 5 pixel disk kernel. The speed of raindrops varies from 0 to 4 pixels/frame (100 pixels per second).

size and blurring levels. The thresholds of the normalized intensity-change and optic flow feature are set to -0.4 and -0.3, respectively, and the smoothness is set to 2.5π.

The detection precision and recall are evaluated using two methods: pixel-based and number-of-raindrop based methods. For the pixel-based method, the ground truth is the pixels with the raindrop blending coefficient $\alpha > 0.1$. Fig. 12 shows the results. As we can see, for highly visible raindrops, the detection precision and recall rate is not obviously affected by raindrop size. The recall rate is mainly affected by raindrop visibility. When the raindrops are too small and hardly visible, the detection recall rate drops, and when the raindrops are blurred, their visibility decreases and the recall rate goes down accordingly.

When evaluated by the number of pixels, the precision rate is higher on detecting larger raindrops. When evaluated by the number of raindrops, however, the precision rate is about the same for raindrops with any size. As the raindrop visibility decreases, the precision does not drop drastically, which indicates a low false alarm rate of our method.

Raindrop motion and detection latency. As discussed in Section 5, our features are more accurate if they are accumulated overtime. In our experiment, we accumulate the features over 100 frames, which takes around 4 seconds for a video with 24 fps. Hence, we assume the raindrops need to be static within 4 seconds (the quasi-static assumption).

We investigate the tolerance of our method on detecting moving raindrops. As illustrated in Fig. 13, we generate synthetic raindrops with controlled motion speed. The raindrop size is 40 pixels and the raindrops are blurred with a 5 pixel disk kernel. The speed of raindrops varies from 0 to 4 pixels/frame (0 to 100 pixels per second).

Accumulating features will increase the distinction between raindrop and non-raindrop areas. However, when raindrops are moving, this is inapplicable anymore. Hence, we need to know how many frames needed to reliably detect raindrops robustly. An example is illustrated in Fig. 14. Here, the thresholds for the normalized intensity-change and optic-flow features are set to 0.4 and 0.3, respectively. The raindrop parameter is set to 60 pixels to 120 pixels. The smoothness is set to 2.5π. The precision and recall of all data is listed in Fig. 15.

As shown, when raindrops are quasi-static, the detection accuracy is stable. The detection accuracy drops significantly when using less than 20 frames. When using 100 frames and the raindrop moving speed is less than 0.4 pixel per frame (10 pixel per second), the detection accuracy is considerably stable. However, when the speed is increased to more than 0.4 pixel per frame, accumulating less than 100 frames increases the accuracy. In this experiments, the optimal number of accumulated frames is 20. The limit raindrop speed of our method is 4 pixels per frame (100 pixel per second). When raindrops move faster and 4 pixels per frames, our method fails to detect them.

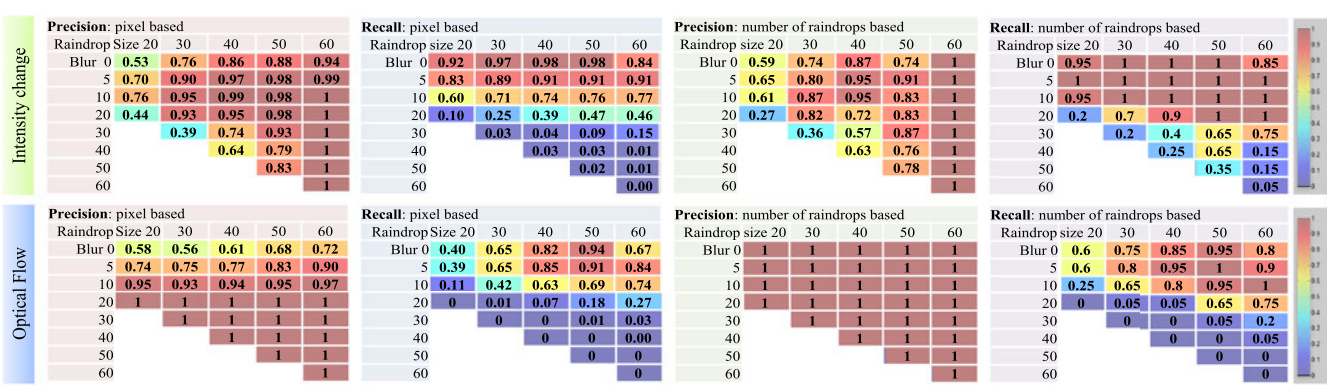


Fig. 12. The precision and recall on detecting raindrops with various size and blur (Fig. 11). The detection threshold is fixed for all of the data. The threshold of the normalized feature is set to 0.4 for the intensity change, and 0.3 for the optic flow. And the smoothness threshold is set to 2.5π.

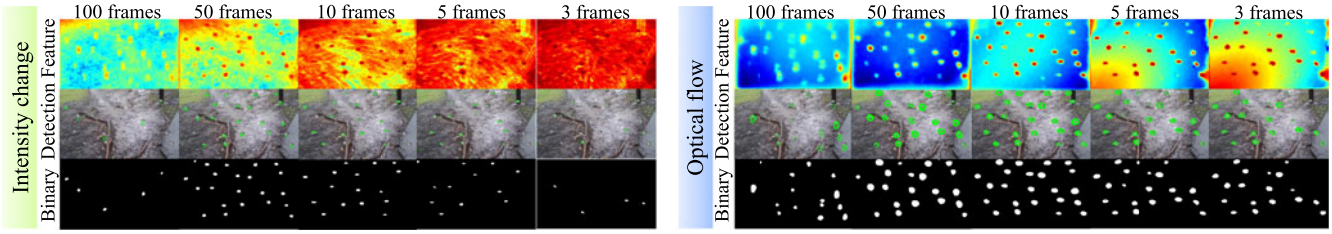


Fig. 14. The influence of number of frames on feature accumulation. Row 1, the accumulated feature. Row 2, the detection result. Row 3, the detection result where the white areas indicate raindrops. The raindrops are 40 pixels of size (long axis) and blurred with a 5 pixel disk kernel, raindrops are moving with a speed 1.2 pixel per frame (30 pixel per second).

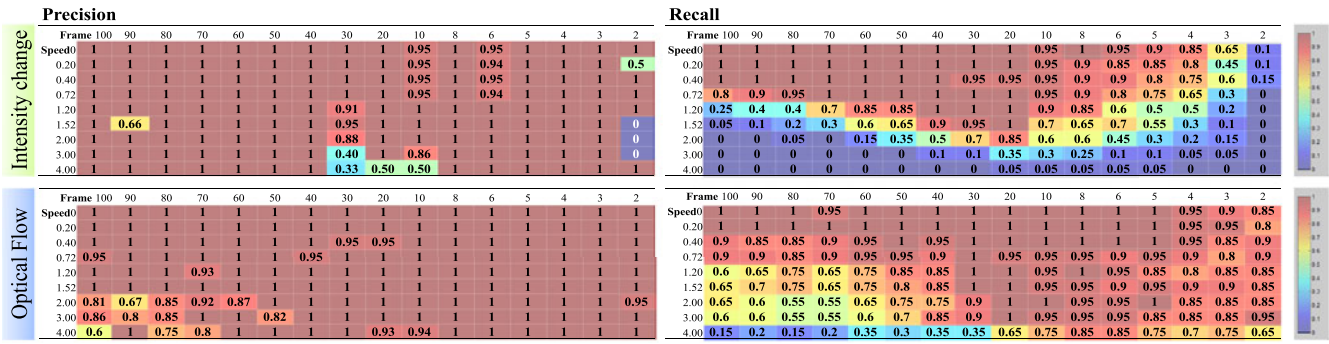


Fig. 15. The precision and recall on detecting raindrops with various raindrop speed and detection latency (Fig. 13). The detection threshold is fixed for all the data. The normalized feature threshold is set to 0.4 for the intensity change, and 0.3 for the optic flow. The raindrop roundness threshold is set to 2.5π.

Textureless Scenes. Our method assumes the environment is sufficiently textured. Hence, in this experiment, we investigate how significant the absence of textures influences the detection accuracy. In this experiment, the threshold for normalized features is set to 0.4 for the intensity change while 0.1 for the optic flow. The smoothness is set to 2.5, and features are accumulated over 100 frames. As illustrated in Fig. 16, we perform Gaussian blur on the scene, with σ varying from 0 to 10, and generate synthetic raindrops with a fixed size (40 pixels) and position.

As illustrated in Fig. 17, when the scene is textureless, the intensity change is affected. The non-raindrop areas change less on a less textured scene. The optic flow, however, is not affected, because optic flow is based on the motion of texture. In addition to that, most of the state of the art optic flow algorithms adopt the coarse-to-fine strategy in estimating the flow. The coarse estimation provides a robust global estimation while the fine estimation provides the accurate and detailed estimation. Thus the texture-less input only affects IC feature. The precision recall is listed in Fig. 18, which shows that when $\sigma > 5$, the accuracy of the intensity change based method drops because the feature on a textureless scene is less distinctive, and the false alarm rate increases.

7.2 Quantitative Comparison on Detection

Real Scenes with Groundtruth. We create a real data by dropping water on a transparent panel as the ground truth and taking videos in the real world. We have a few scenarios for the experiments. Experiment 1 includes the disturbance of the light sources. Experiment 2 emphasizes on the varying shape and size of raindrops. Experiment 3 focuses on significantly blurred raindrops, and experiment 4 includes glare. The input and results are shown in the first four columns in Fig. 19.

We compare our method with Eigen et al.'s [17], Roser et al.'s [14] and Kurihata et al.'s [16] method. Yamashita et al.'s [18], [19] methods require stereo cameras or a pantile camera and are, thus, not included in the comparison. The results are shown in the last two columns of Fig. 19.

We use the precision-recall curve to quantitatively analyze the performances. The results for each experiment are shown in Fig. 20. According to the results, both of our proposed methods outperform the existing methods. By

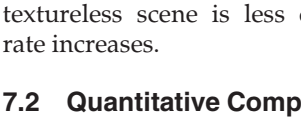


Fig. 16. Gaussian blur on a scene, with σ varying from 0 to 10. The patch size is 120×120 pixels.

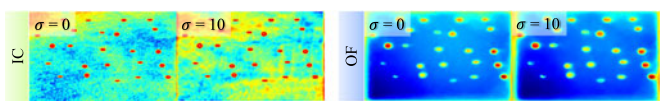


Fig. 17. The accumulated feature using intensity change and optic flow on textured and textureless scenes. 100 frames are used for accumulation.

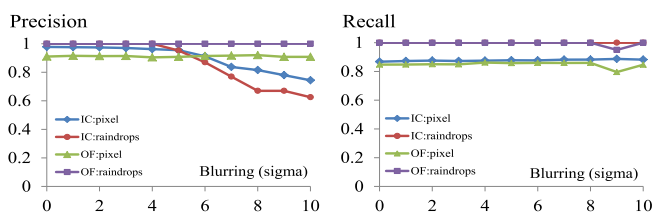


Fig. 18. The precision and recall of raindrop detection on textured and textureless scenes. The threshold for normalized features is set to 0.4 for the intensity change and 0.1 for the optic flow. The raindrop parameter is set to 60 pixels to 160 pixels. The roundness threshold is set to 2.5π. Features are accumulated over 100 frames.

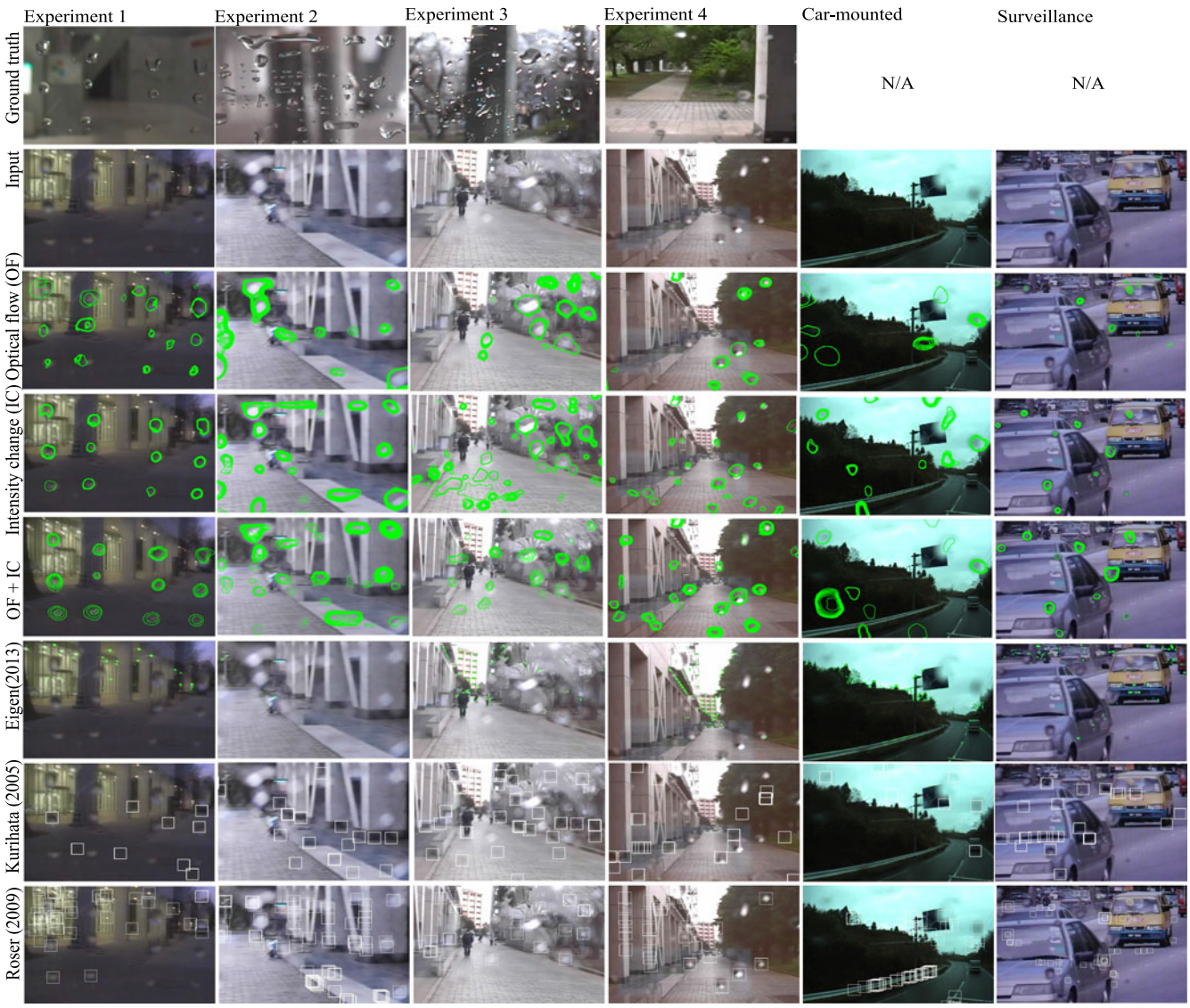


Fig. 19. The detection results using our methods and the existing methods.

combining IC (intensity change) with OF (optical flow), we obtain the best performance to detect all of the raindrops (because of IC) while keeping a low false alarm rate (because of OF). The detection using the intensity change performs best. Unlike the existing methods that only detect the center and size of raindrops, our proposed method can detect raindrops with a large variety of shapes. Our method also achieves high robustness in detecting highly blurred and glared raindrops.

Real scenes without groundtruth. Fig. 19 shows the results of our detection method in the following 3 situations: 1) A daily use hand held camera, as in experiments 1-4. 2) A vehicle-mounted camera, which is widely used for navigation and data collection. 3) A surveillance camera which is mounted into a fixed location. Our method outperforms the existing methods in the all three situations as shown in the figure.

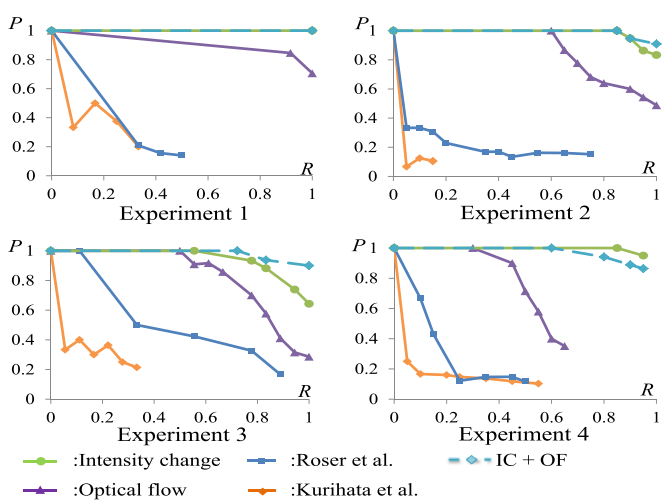


Fig. 20. The precision(R)-recall(R) curves of our methods and the two existing methods. The thresholds of our normalized features are labeled.

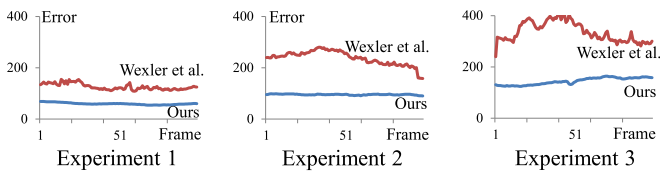


Fig. 21. Average $(R; G; B; dx; dy; dt)$ error of recovering 100 continuous frames of the experiment shown in Fig. 22.

Fig. 22. The raindrop removal results using our methods and the method of Wexler et al. [21].

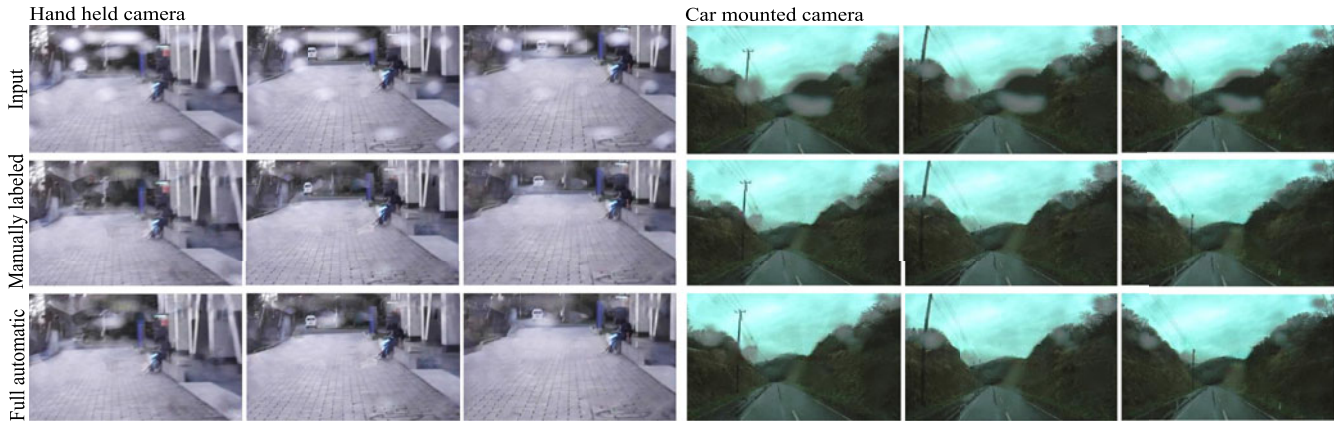


Fig. 23. The raindrop removal using our method. First row: the input sequence. Second row: the removal result with the raindrops manually labeled. Third row: the removal result with the raindrops automatically detected.

7.3 Raindrop Removal

As illustrated in the first two columns of Fig. 22, the synthesized raindrops are generated on a video, and used as an input. Our method is compared with the method proposed by Wexler et al. [21]. In [14], there is insufficient description for the removal algorithm and thus it is not compared here. The results are shown in the last four columns of Fig. 21.

As shown in Fig. 21, for the quantitative evaluation, we run each of them on 100 continuous frames and calculate the average error per pixel for each frame. The same as Wexler et al. [21], the error is calculated on both the 8 bit $(R; G; B)$ value and spatial-temporal gradients $(dx; dy; dt)$. The proposed method benefits from the restoration in all the 3 situations. Using the same computer, our method needs 5 seconds per frame to remove raindrops, and Wexler et al.'s needs 2 minutes.

We show a few results of removing raindrops in videos taken by a handle held camera and a vehicle-mounted camera, as shown in the first and second rows of Fig. 23 where we can see some improvement. To demonstrate the performance of our raindrop removal method, the manually labeled raindrops are also included. The overall automatic raindrop detection and removal results in videos taken by a hand held camera and a car mounted camera are shown in the third row of Fig. 23, where we can see significant improvement. [1]

1. http://www.cvl.iis.u-tokyo.ac.jp/~yousd/CVPR2013/Shaodi_CVPR2013.html

7.4 Applications

To show the benefits of our method, we apply it to two common applications in computer vision: motion field estimation and structure from motion.

Motion estimation. Adherent raindrops occlude the background and their motion is significantly different from the background motion. By removing the raindrops, we show that the motion in the background can be correctly estimated. We demonstrate the improvement on various scenes shown in Fig. 24. SIFT-flow [30] is used for the motion estimation; although, any optic flow algorithm can also be used.

In the first scene of Fig. 24 (columns 1 and 2), we applied our method to a synthetically generated raindrop. As can be seen, the motion field of the raindrop images (row 2) is significantly degraded compared to that of the clear images (row 1). Having removed the raindrop, the motion field becomes more similar to that of the clear images (row 3). In the second scene (columns 3 and 4) of Fig. 24, the images have global motion because of the shaking camera. Although the estimation on the repaired images reflects the global motion, the estimation on raindrop images is also significantly affected. In the last scene (columns 5 and 6), the car-mounted camera is moving forward and the motion on the repaired images correctly reflects the camera motion.

Since we have the clear image of the scene (Fig. 24 column 1), we can quantitatively evaluate the improvement of our restoration by measuring the end point error of optical flow estimation on raindrop area computed from the raindrop video

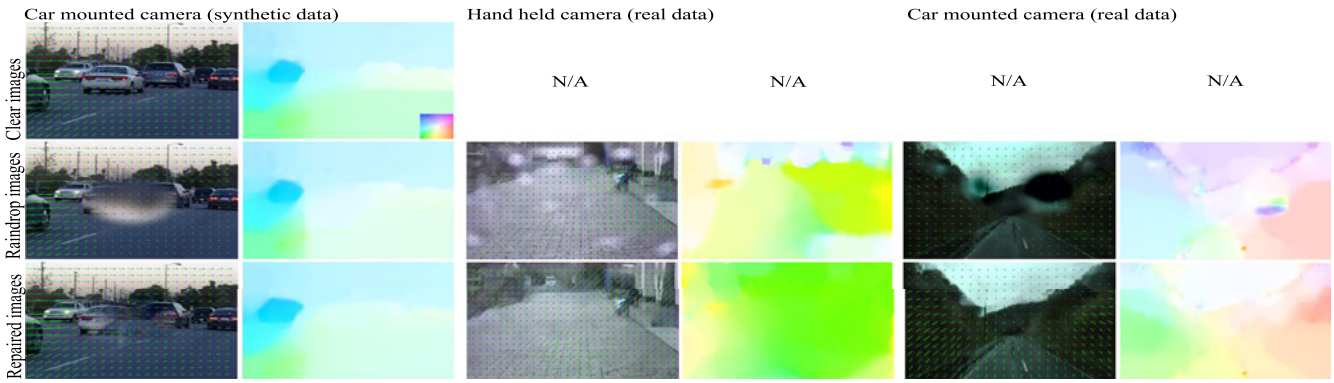


Fig. 24. Motion estimation using a clear video, raindrop video and repaired video.

and restored video. We perform the evaluation using 100 consecutive frames and the result is illustrated in Fig. 25, where the average endpoint error of the restored video is reduced.

Structure from motion (SfM). Adherent raindrops move along with the camera adversely affect the camera parameter estimation. As a result, they also negatively affect the accuracy of the dense depth estimation. Hence, we expect that with raindrops being removed, the robustness of the camera parameter and depth estimation associated with the structure from motion technique can be improved. As illustrated in Fig. 26, we perform the structure from motion method by Snavely et al. [32]. We use a clear video, a video with adherent raindrop and a repaired video as inputs. Samples of those videos are shown in the second row of Fig. 22 . As can be seen, the repaired video provides better results than that of the raindrop video on both the camera parameter estimation and dense depth estimation.

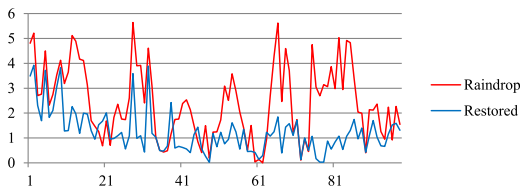


Fig. 25. Average end-point error of optical flow on raindrop video and restored video.

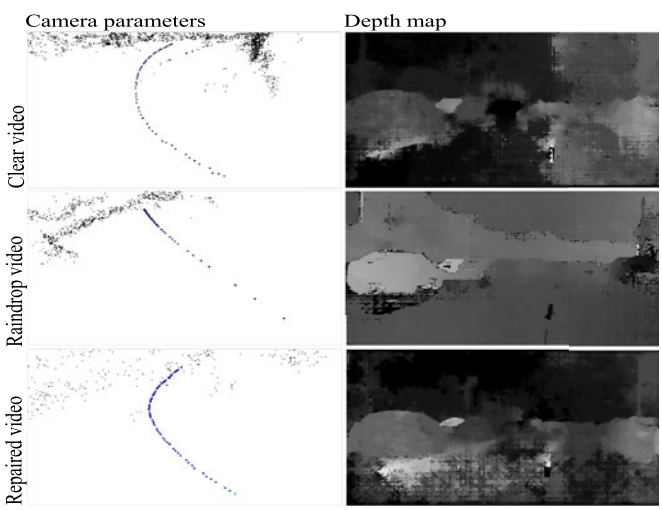


Fig. 26. Structure from motion using a clear video, raindrop video and repaired video. The input view are shown in the second row of Fig. 22.

8 Conclusion

We have introduced a novel method to detect and remove adherent raindrops in video. The key idea of detecting raindrops is based on our theoretical findings that the motion of raindrop pixels is slower than that of non-raindrop pixels, and the temporal change of intensity of raindrop pixels is smaller than that of non-raindrop pixels. The important idea of our raindrop removal is to solve the blending function with the clues from detection and intensity change in a few consecutive frames, as well as to employ a video completion technique only for those that cannot be restored. To our knowledge, our automatic raindrop detection and removal method is novel and can benefit many applications that possibly suffer from adherent raindrops.

Acknowledgments

This research is granted by: 1. The Japan Society for the Promotion of Science (JSPS) through the "Funding Program for Next Generation World-Leading Researchers (NEXT Program)," initiated by the Council for Science and Technology Policy (CSTP). 2. Next-generation Energies for Tohoku Recovery (NET), MEXT, Japan.

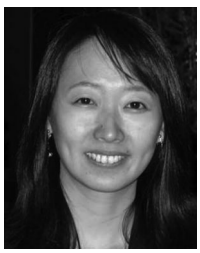
References

[1] R. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[2] R. Fattal, "Single image dehazing," in *Proc. ACM SIGGRAPH*, vol. 27, no. 3, 2008, pp. 72–80.
[3] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1956–1963.
[4] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 617–624.
[5] P. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, no. 2-3, pp. 256–274, 2010.
[6] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3–27, 2007.
[7] K. Garg and S. Nayar, "Photometric model of a rain drop," *Columbia Univ., New York, NY, USA*, Tech. Rep. garg2003photometric, 2003.
[8] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 1, p. I–528.
[9] L. Kang, C. Lin, and Y. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.
[10] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1968–1975.

[11] R. G. Willson, M. Maimone, A. Johnson, and L. Scherr, "An optical model for image artifacts produced by dust particles on lenses," Pasadena, CA, USA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2005.
[12] C. Zhou and S. Lin, "Removal of image artifacts due to sensor dust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[13] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing image artifacts due to dirty camera lenses and thin occluders," *ACM Trans. Graphics (TOG)*, vol. 28, no. 5, p. 144, 2009.
[14] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 570–577.
[15] M. Roser, J. Kurz, and A. Geiger, "Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves," in *Proc. Int. Conf. Comput. Vis. (ACCV)*, 2010, pp. 235–244.
[16] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, "Rainy weather recognition from in-vehicle camera images for driver assistance," in *Proc. IEEE Intell. Vehicles Symp.*, 2005, pp. 205–210.
[17] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 633–640.
[18] A. Yamashita, Y. Tanaka, and T. Kaneko, "Removal of adherent water-drops from images acquired with stereo camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2005, pp. 400–405.
[19] A. Yamashita, I. Fukuchi, and T. Kaneko, "Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2009, pp. 3794–3801.
[20] T. Hara, H. Saito, and T. Kanade, "Removal of glare caused by water droplets," in *Proc. Conf. Vis. Media Prod.*, 2009, pp. 144–151.
[21] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, pp. I-120–I-127.
[22] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video repairing: Inference of foreground and background under severe occlusion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, pp. I-364–I-371.
[23] J. Jia, Y. Tai, T. Wu, and C. Tang, "Video repairing under variable illumination using cyclic motions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 5, pp. 832–839, May 2006.
[24] G. Sapiro and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545–553, Feb. 2007.
[25] T. Shiratori, Y. Matsushita, S. B. Kang, and X. Tang, "Video completion by motion field transfer," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 411–418.
[26] M. Liu, S. Chen, J. Liu, and X. Tang, "Video completion via motion guided spatial-temporal global optimization," in *Proc. ACM 17th Int. Conf. Multimedia (ACMM)*, 2009, pp. 537–540.
[27] E. Villermaux and B. Bossa, "Single-drop fragmentation determines size distribution of raindrops," *Nature Physics*, vol. 5, no. 9, pp. 697–702, 2009.
[28] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi, "Adherent raindrop detection and removal in video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 1035–1042.
[29] M. Subbarao, "Depth recovery from blurred edges," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 1988, pp. 498–503.
[30] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.
[31] O. Stanley and F. Ronald, *Level set methods and dynamic implicit surfaces*. New York, NY, USA: Springer-Verlag, 2002.
[32] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM Trans. Graphics (TOG)*, vol. 25, no. 3, pp. 835–846, 2006.

Shaodi You received the bachelor's degree from Tsinghua University, P. R. China in 2009, the ME and PhD degrees from The University of Tokyo, Japan in 2012 and 2015, respectively. He works as a researcher at NICTA from 2015. His research interests include physics based vision, image/video enhancement, and machine learning. He is a student member of the IEEE.

Robby T. Tan received the PhD degree in computer science from the University of Tokyo. He is an assistant professor at Yale-NUS College and National University of Singapore (NUS). Before coming to Singapore, he was an assistant professor at Utrecht University. His research interests include physics-based computer vision, particularly in bad weather. He is a member of the IEEE.



Rei Kawakami received the BS, MS, and PhD degrees in information science and technology from the University of Tokyo in 2003, 2005, and 2008, respectively. She is an assistant professor at the University of Tokyo, Tokyo, Japan. Her research interests include color constancy, spectral analysis, and physics-based computer vision. She is a member of the IEEE.

Yasuhiro Mukaigawa received the ME and PhD degrees from the University of Tsukuba in 1994 and 1997, respectively. He became a research associate at Okayama University in 1997, an assistant professor at University of Tsukuba in 2003, an associate professor at Osaka University in 2004, and a professor at Nara Institute of Science and Technology (NAIST) in 2014. His current research interests include photometric analysis and computational photography. He is a member of the IEEE.

Katsushi Ikeuchi received the BE degree from Kyoto University in 1973 and the PhD degree from the University of Tokyo in 1978. After working at the Massachusetts Institute of Technology (MIT) Artificial Intelligence (AI) Laboratory for three years, ETL for five years, and Carnegie Mellon University (CMU) Robotics Institute for 10 years, University of Tokyo Institute of Industrial Science for 19 years, he joined Microsoft Research Asia in 2015 as a principal researcher. His research interests include computer vision, robotics, and computer graphics. In these research fields, he has received several awards, including the David Marr Prize in computational vision for the paper "Shape from Interreflection," and "IEEE Robotics and Automation Society." K. S. Fu memorial best transaction paper award for the paper "Toward Automatic Robot Instruction from Perception-Mapping Human Grasps to Manipulator Grasps." In addition, in 1992, his paper "Numerical Shape from Shading and Occluding Boundaries" was selected as one of the most influential papers to have appeared in the *Artificial Intelligence Journal* within the past 10 years. His community service include general chair of IROS 95, ITSC 99, IV 01, ICCV 05, ACCV 07; program chair of CVPR 96, ICCV 03, ICRA 09, ICPR 12, ICCV 15; associate editor of IEEE Trans. RA, IEEE Trans. PAMI; and a distinguished lecturer of the Signal Processing Society in 2000-2002 and Robotics and Automation Society in 2004-2006. He is the editor-in-chief of the *International Journal of Computer Vision*. Through these research and society service, he was awarded a fellow from the IEEE, IEICE, IPSJ and RSJ. He received the Distinguished Researcher Award from IEEE-PAMI, Medal of Honor with Purple Ribbon (Shiju-ho-syo) from Japanese Emperor, and the Okawa award from Okawa foundation.