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,* 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. First, we explicitly model adherent raindrops using law of physics, and then, detect them based on these models in combination with motion and intensity temporal derivatives of the input video. Second, relying on an analysis that some areas of a raindrop completely occludes the scene, yet the remaining areas occlude only partially, we remove the two types of areas separately. 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 the proposed method.

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

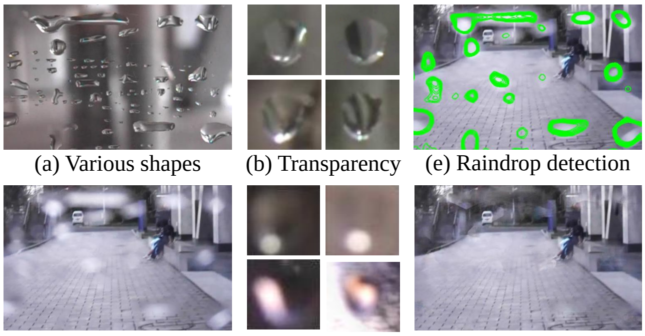
1 Introduction

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, making the performances of many algorithms in the vision systems (such as feature detection, tracking, stereo correspondence, etc.) 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 a few reasons as shown in Fig. 1. Foremost, adherent raindrops have various shapes. Unlike opaque objects, they are transparent, making their appearance and thus intensity values vary depending on the environment. They suffer from out-of-focus blur due to their proximity to the camera. Moreover, most raindrops generate glare.

To address the problems, we analyze the appearance of adherent raindrops from their local spatio-temporal derivatives. First, 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. Second, unlike clear raindrops, blurred raindrops are mixtures of rays originated from the points in the entire scene. Because of this, the intensity temporal derivative of blurred raindrops is significantly smaller than that of non-raindrops. These two clues are the key idea of our detection method. Base on it we propose the pixel basis detection method, which is generally applicable to handle any shape and size of raindrops. Fig. 1.e shows a result of our detection method.

By further analyzing the image formation of raindrops, we found that some area 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, namely, by solving a blend-



(a) Various shapes (b) Transparency (e) Raindrop detection (c) Blurring (d) Glare (f) Raindrop removal

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

- *S. You, K. Ikeuchi and R. Kawakami are with The University of Tokyo, Tokyo, Japan. E-mail: {yousd, ki}@cvl.iis.u-tokyo.ac.jp, rei@nae-lab.org*
- *R. T. Tan is with SIM University, Singapore. E-mail: tanrobby@gmail.com*
- *Y. Mukaigawa is with Nara Insititute of Science and Technology, Japan. E-mail: mukaigawa@is.naist.jp*

ing 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. 1.f shows a result of our removal method.

The contributions of the paper are threefold:

- Adherent raindrops are theoretically modeled and analyzed using the derivative properties with few parameters, enabling the method to be applied to general video cameras, *e.g.*, hand-held and vehicle-mounted cameras.
- A novel real-time pixel-based detection method is introduced.
- A relatively fast adherent raindrop removal method is proposed. It utilizes not only a video completion technique, but also the information behind some blurred areas of raindrops.

The rest of the paper is organized as follows. Sec. 2 discusses the related work on raindrop detection and removal. Sec. 3 explains the modeling of the spatial derivative properties of the raindrop images, followed by temporal derivative properties in Sec. 4. The detailed methodology of the raindrop detection is described in Sec. 5, followed by the detailed methodology of the raindrop removal in Sec. 6. Sec. 7 shows the quantitative experiments and results. Sec. 7 concludes the paper.

2 Related Work

Bad weather enhancement Removing the influence of haze, mist, to some extent fog (e.g., [1], [2], [3], [4]), rain and snow (e.g., [5], [6]) have been well exploited. Dealing with rain, Garg and Nayar first model it [7], and then detect and remove it [8], [6]. Later, Barnum *et al*. [5] propose a method to detect and remove both rain and snow. Later, 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 rather not possible, since the physics and appearance of falling raindrops are significantly different from those of adherent raindrops.

Sensor dust removal Sensor dust removal is also 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 will block the 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. However, in their paper only detection of the dark dust artifact is discussed. Later 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 which both dark and lighten the image slightly. Although adherent raindrops could also be considered as a kind of sensor dust, they cannot be handled by existing sensor dust removal methods, since raindrops could be large and not sufficiently blurred.

Adherent raindrop detection and removal Methods for detecting adherent raindrops caused by light rain have been proposed. Roser *et al*. attempt to model the shape of adherent raindrops by a sphere crown [14], and later, Bezier curves [15]. However, the models are insufficient, since a sphere crown and Bezier curves can cover only a small portion of possible raindrop shapes (Fig. 1.a). Kurihata *et al*. [16] and later Fergus *et al*. [17] approach it through machine learning. However, as shown in Figs. 1.a-d, collecting and aligning training images for all various shapes, environment, illumination and blurring are considerably challenging. Both of their 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. Fergus *et al*. [17] use machine learning and directly place the raindrop template with clear template.

Image/Video inpaiting and completion Video completion has been intensively exploited by computer vision researchers. However, 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 somewhere else in the video. Jia *et al*. [22] exploit video completion by separating static background and 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 consider using cues from our adherent raindrop modeling to help the removal.

3 Clear Raindrop Modeling

In this section, we first consider the camera as pinhole camera so that both raindrop and background are not blurred. Based on the 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 have only few parameters.

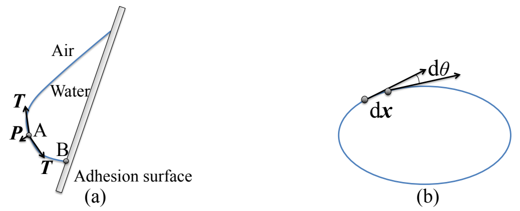


Fig. 2. a. Balance at raindrop surface. A denotes a two-phase point. B denotes a three-phase point. T denotes a surface tensor, and P for pressure. At two-phase point A, surface tensor T and pressure P are balanced. b. Change of tangle angle along raindrop boundary.

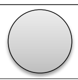
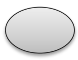
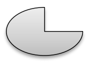
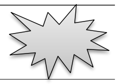
Shape				
Smoothness	2π(6.28)	2π(6.28)	3π(9.42)	54.15
Roundness	1/4π(0.080)	0.075	0.050	0.016

Fig. 3. Smoothness and roundness of some typical shapes.

3.1 Physical Attributes

Fig. 1.a and b are observations of real adherent raindrops. As we can see, adherent raindrops have various shape and size, and their appearance is totally 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 for it depends on the gravity, water-water surface tensor and water-adhering-surface tensor and many other parameters.

Fortunately, it is possible to give an upper bound of the size by using few parameters. As illustrated in Fig. 2.a, to prevent raindrop 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 the three phase point 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., 5mm [27].

Shape While most existing methods assume the shape of raindrops to be circle or ellipse, the real raindrop shape varies in a large range. (Fig. 1). Fortunately, we can still find some regular patterns of raindrop shapes due to the surface tensor. Raindrop boundaries are smooth and raindrops are convex in most cases.

Hence, we quantitatively characterize raindrop shape using two features: shape smoothness and roundness. As illustrated in Fig. 2.b, given a raindrop area on the image plane, denoted as R, we can integrate the change of the tangent angle along the

TABLE 1
Raindrop dynamic of scenes in Fig. 19

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

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 $2D$ coordinate 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 perimeter:

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

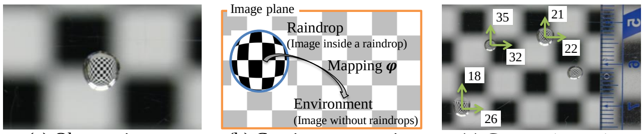
A rounder shape would have a larger roundness value and a perfect circle has the maximum roundness value: $\frac{\pi r^2}{(2\pi r)^2} = \frac{1}{4\pi} = 0.080$. Fig. 3 shows some examples.

Both the smoothness and roundness are invariant to scaling and rotation. Unlike our previous method [28], which used the roundness, in the current method we utilize 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 tensor coefficients, surface 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. We observe the motion of real adherent raindrops in scenes in Fig. 19. For a raindrop, we compare 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. In our research, we only need to assume the raindrops to be static within seconds, and we quantitatively evaluate the tolerance of raindrop dynamics in Section 7.

3.2 Clear Raindrop Imagery

As shown in Fig. 4.a, 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. 4.c are the contraction ratios between the original image and the image inside the raindrops calculated from the black and white patterns. The contraction



(a) Observation (b) Continuous mapping (c) Contraction ratio

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

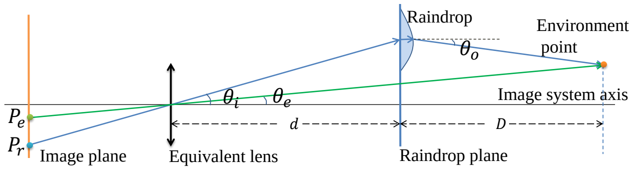


Fig. 5. The refraction model of two points on an image plane (P_e and P_r) that are originated from the same point in the environment. There are two refractions on the light path passing a raindrop. The camera lens cover or protecting shield is assumed to be a thin plane and thus can be neglected.

ratio is 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, so that both the raindrop and environment are not blurred. As illustrated in Fig. 5, there exists point pairs on the image plane which are the images of the same environment point. One of which the light ray directly goes to the image plane, denoted as P_e, and the other goes 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. 4.b, 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. Fortunately, our analysis looks into the spatial derivative properties, and therefore can avoid obtaining $\boldsymbol{\varphi}$ explicitly.

3.3 Spatial Derivative of Clear Raindrop: The Contraction Ratio

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

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

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 distance from a raindrop to camera. The raindrop upper bound has been discussed in Sec. 3.1. The raindrop-camera distance lower bound depends on camera settings. In our observation, normally $d < 200mm$.

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

$$\mathcal{E}_\varphi > 10 \gg 1. \tag{5}$$

The proof is provided in Appendices A and B.

3.4 Detect Raindrops Using Optical Flow

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

$$\begin{aligned}\frac{\|M(P_e)\|}{\|M(P_r)\|} &= \frac{\|P_e(t_2) - P_e(t_1)\|}{\|P_r(t_2) - P_r(t_1)\|}\\ &= \frac{\|\boldsymbol{\varphi}(P_r(t_2)) - \boldsymbol{\varphi}(P_r(t_1))\|}{\|P_r(t_2) - P_r(t_1)\|} \geq \mathcal{E}_\varphi.\end{aligned} \tag{6}$$

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

4 Blurred Raindrop Modeling

Unlike raindrop imagery with a pin-hole camera, for a normal lens camera, when the camera focuses on the environment scene, raindrops will be blurred. Concerning this, we first model the blurred raindrop imagery, and theoretically derive the temporal property of raindrop pixels, namely, a raindrop pixel has less high frequency component. Based on this property, we propose a pixel-wise raindrop detection feature: intensity change.

4.1 Blurred Raindrop Imagery

As illustrated in Fig. 6, the appearance of a pixel on an image plane depends the collection of light, which can be light emitted from a focused 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).

Let us model the image intensity of blurred pixels using a blending function. We denote the light intensity collected by pixel (x, y) as $I(x, y)$, the light

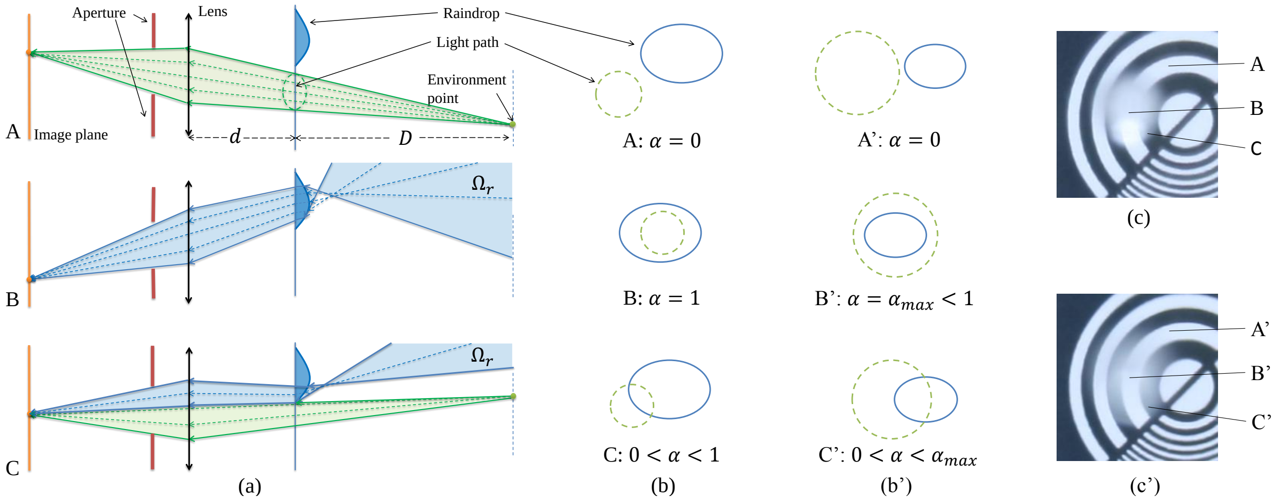


Fig. 6. Rows: The appearance and model of pixels on an image plane collecting light from A: environment, B: raindrop, C: both. Columns: (a) The light path model. Green light: the light coming from environment point; Blue light: the light refracted by a raindrop. (b) Raindrop-plane-cut of the model in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

intensity formed by an environment point that intersects with the line of sight without being through a raindrop as $I_e(x, y)$, and the light intensity reached (x, y) through a raindrop as $I_r(x, y)$. Then, pixel (x, y) collecting light from both the raindrop and the environment can be described as:

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

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

Blending coefficient α is determined by the area of 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 uses a blurring kernel. First, as illustrated in Fig. 7.a, α is either 0 or 1 if the raindrop is clear. We denote the blending coefficient of clear raindrops as α_c. Then, 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. (16) 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 block the environment. Fig. 6.c′ is an example.

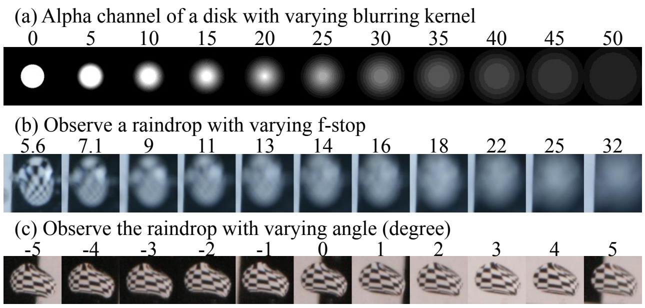


Fig. 7. a. α channel of a disk (with radius as 20 pixel) with varying blurring kernel (radius in pixel) b. Observe a raindrop with varying f-stop. c. Observe a raindrop with varying angle (degree). Raindrop appearance is highly directional..

Fig. 7.b is an observation of real blurred raindrops. In Roser *et al.*[14]'s work, other than α, it directly convolutes a clear raindrop image with the blurring kernel to obtain a blurred raindrop image. We notice that this method is inaccurate, because the convolution with a uniform disk kernel assumes isotropic refraction, which is not true for most raindrops. As shown in Fig. 7.c, raindrop appearance are highly directional.

4.2 Temporal Derivative of Blurred Raindrop

We avoid estimating the exact appearance of blurred raindrops because this estimation is intractable. Alternatively, we explore the temporal derivative features. In consecutive frames, we observed that the intensity of blurred pixels (case B and C) does not change as distinctive as that of environment pixels (case A). To analyze this property more carefully, let us look into the intensity temporal derivatives of blurred pixels. Referring to Figs. 6.a B and C, light collected from raindrop is actually refracted from a large area in the environment. We refer to 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 to be constant in a short time period.

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

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

Here, by considering Eq. (5), we know that the area ratio is more than one hundred, namely,

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

(Notice φ is not conformal, a strict proof is provided in Appendix C), and thus, we can consider $\Omega_r(x,y)$ to be a sufficiently large area. According to the law of large number, we 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 derivative works as a high pass filter, we may also consider Eq. (13) in a frequency domain, where the temporal high frequency component on a raindrop is significantly smaller than those of the environment, described as:

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

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

4.3 Detect Raindrops Using Intensity Change

By taking into account Eq. (13) with Eq. (7), the temporal difference for $I(x,y,t)$ will be small when α is large:

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

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

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

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

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

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

4.4 Discussion on Effects of Glare

As illustrated in Fig. 1.d, a raindrop will refract bright lights from the environment, and generate glare. This phenomenon will not affect the derivative properties described in the previous subsections. The reasons are, first, glare is caused by a light source emitting high intensity light, and the spatial derivative introduced in Sec. 3.1 is independent from light intensity. Second, the appearance of glare in videos 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 does not affect the analysis we have discussed.

5 Raindrop Detection

5.1 Feature extraction

Based on the analysis of motion and the intensity temporal derivative, we generate features for the detection. First, we calculate dense motion, *e.g.*, SIFT-flow [30], as shown in Fig. 8.b. Second, we calculate the intensity temporal change $|I(x,y,t_1) - I(x,y,t_2)|$, as shown in Fig. 9.b.

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

We employ both features to achieve optimal accuracy. If time is a concern, we use only intensity change. Real time detection is described in Sec. 5.3.

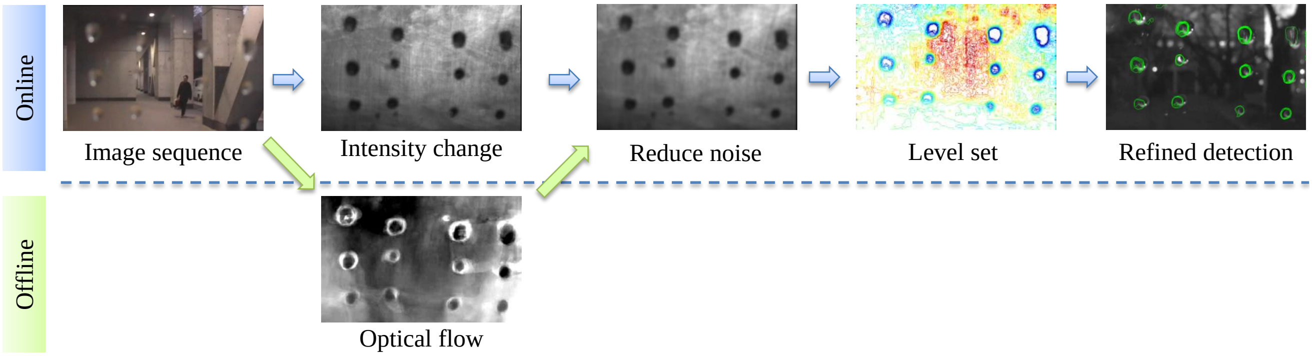


Fig. 10. The detection workflow. Our method can work in real time if using only intensity change.

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.

The following criteria are applied further for determining raindrop areas:

1) Feature threshold. As analyzed previously, raindrop areas should have smaller feature values. Specifically, we normalized the accumulated feature with the mean value at 0 and variance at 1. In our experiment, those pixels with feature values less than -0.7 are considered to be raindrop pixels.
2) Smoothness. As analyzed in Sec. 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 use 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 workflow for one phase. For each detection, we accumulate the feature for the past 4 seconds and compute the level set to detect raindrops. The overall detection algorithm is described in Algorithm. 1.

5.3 Realtime detection

The detection method can work in real time if we use only the intensity change as the feature. We ran our program on a 3.1GHz CPU and Matlab (Windows) with no parallelization. The video was 1280*720, 24fps. We used the profiling tool in Matlab for recording the computational time. Accumulating the feature took 0.0086s per frame, which was 0.10s for 12 frames. Gaussian filter took 0.04s. The level sets took 0.22s. Selecting contours took 0.06s. The overall computing time for each detection phase was 0.42s.

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 optical 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
```

6 Raindrop Removal

While the existing methods try to restore the entire areas detected as raindrops by considering them as solid occluders [14], [19] it will be more factual if we can restore the raindrop areas from the source scenes whenever possible. Based on Eq. (7), we know that some area 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)
  $N = 100$, $\omega_{th} = 0.05N$, $\Delta x = \Delta y = \pm 1\text{pixel}$
  $th1 = 250$, $th2 = 40$
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 the detection phase, the position and shape of raindrops on the image plane are known. Using the out-of-focus blur model in Fig. 6.a, 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 by experiments (though, it is not a strict parameter and is 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 in Figs. 6.b and b′. $\alpha(x, y)$ 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:

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

where $(x + \Delta x, y + \Delta y), \Delta x, \Delta y \le 1$ pixel are spatial neighborhood of (x, y). When averaging, we exclude neighboring pixels that have intensity difference 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

After restoring the partially occluding raindrop pixels, there are two types of remaining areas need to be completed:

- When α is close or equal to 1.0, I_e will be too scarce to be restored, as shown in Eq. (17). 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 conducted quantitative experiments to measure the accuracy and general applicability of our proposed detection and removal method. Results in video are included in the supplementary material.

7.1 Quantitative analysis on detection

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

Raindrop Size and Blur As discussed in Sec. 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

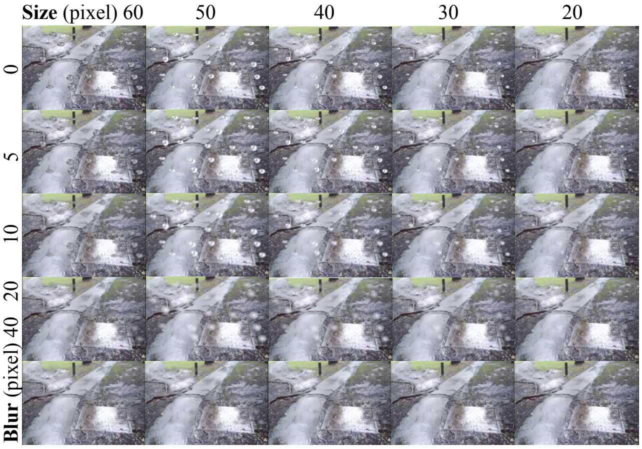


Fig. 11. Synthetic raindrops with various sizes and blurring levels. The image size is 720*480, raindrop size (long axis) varies from 20 to 60 pixels, and the radius of the disk-blurring-kernel varies from 0 to 40 pixels.

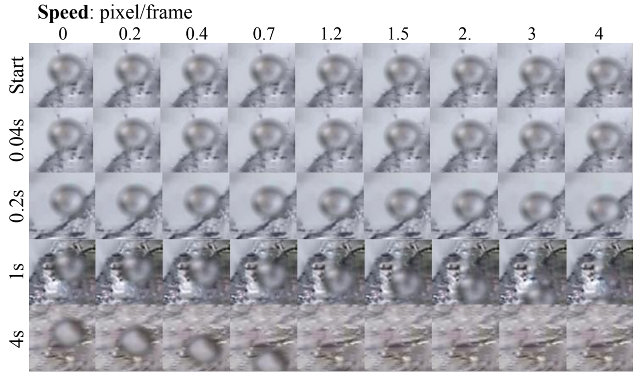


Fig. 13. Appearance of synthetic moving raindrops. The raindrop size were 40 pixels and were blurred with a 5 pixel disk kernel. The speed of raindrops varied from 0 to 4 pixels/frame (100 pixels per second).

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 also 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 explored how raindrop size and blur affect the detection accuracy. As illustrated in Fig. 11, we generated synthetic raindrops with fixed positions, but with various sizes and blurring levels. We fixed the detection thresholds. The thresholds of the normalized intensity-change and optical flow feature were set to -0.4 and -0.3, respectively, and the smoothness was set to 2.5π.

The detection precision and recall were 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 was not obviously affected by raindrop size. The recall rate was mainly affected by raindrop visibility. When the raindrops were too small and hardly visible, the detection recall rate dropped, and when the raindrops were blurred, their visibility decreased and the recall rate went down accordingly.

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

Raindrop Motion and Detection Latency As discussed in Sec. 5, our features are more accurate if they are accumulated overtime. In our experiment, we accumulated the features over 100 frames, which took 4 seconds for a video with a frame rate of 25 fps. Hence, we assumed the raindrops need to be static within 4 seconds.

We investigated the tolerance of our method on detecting raindrops which is not quasi-static. As illustrated in Fig. 13, we generated synthetic raindrops with controlled motion speed. The raindrop size were 40 pixels and were blurred with a 5 pixel disk kernel. The speed of raindrops varied 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 threshold for the normalized intensity change and optical flow features were set to 0.4 and 0.3 respectively. The raindrop parameter was set to 60 pixels to 120 pixels. The smoothness was 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 was stable. The detection accuracy dropped significantly when using less than 10 frames. When using 100 frames and the raindrop moving speed was less than 0.4 pixel per frame (10 pixel per second), the detection accuracy was considerably stable. However, when the speed was increased to more than 0.4 pixel per frame, accumulating less than 100 frames increased the accuracy. In this experiments, the optimal number of accumulated frames was 20. The limit raindrop speed of our method was 4 pixel per frame (100 pixel per second). When raindrops moves faster and 4 pixels per frames, our method failed to detect them. Fortunately, 4 pixels per frames is considerably fast, which is rare in light rainy scenes.

Textureless Scenes Our method assumes the environment is sufficiently textured. Hence, in this exper-

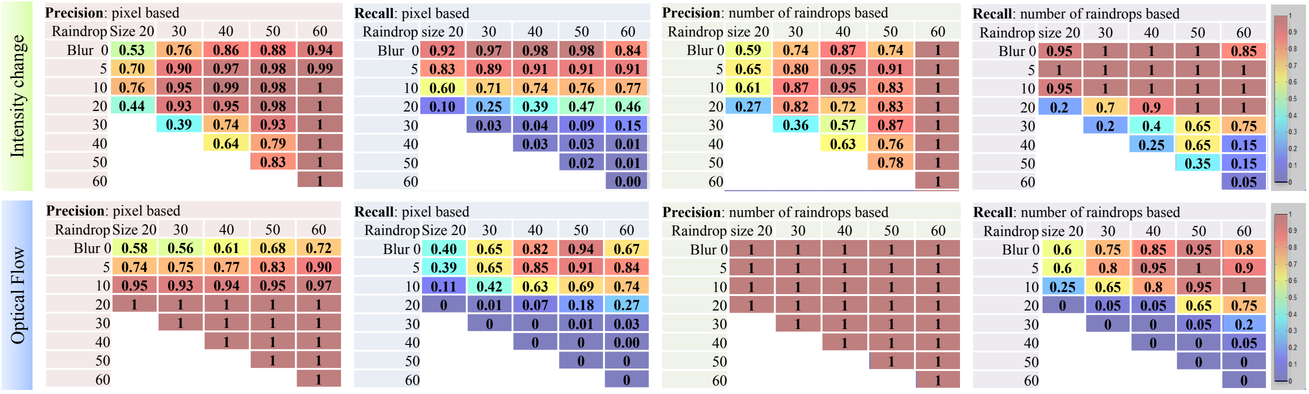


Fig. 12. The precision and recall on detecting raindrops with various size and blur (Fig. 11). The detection threshold was fixed for all of the data. The threshold of the normalized feature was set to 0.4 for the intensity change, and 0.3 for the optical flow. And the smoothness threshold was set to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

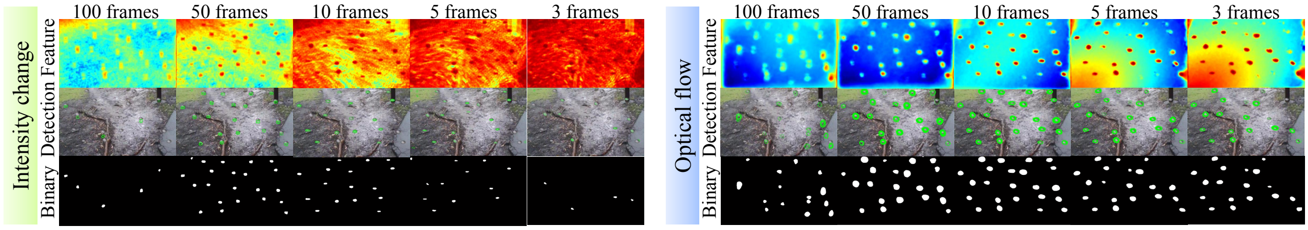


Fig. 14. The influence of number of frames for feature accumulation. Row 1, the accumulated feature. Row 2, the detection result. Row 3, the detection result where the white area indicate raindrop. The raindrop size were 40 pixels (long axis) and blurred with a 5 pixel disk kernel, raindrops were moving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

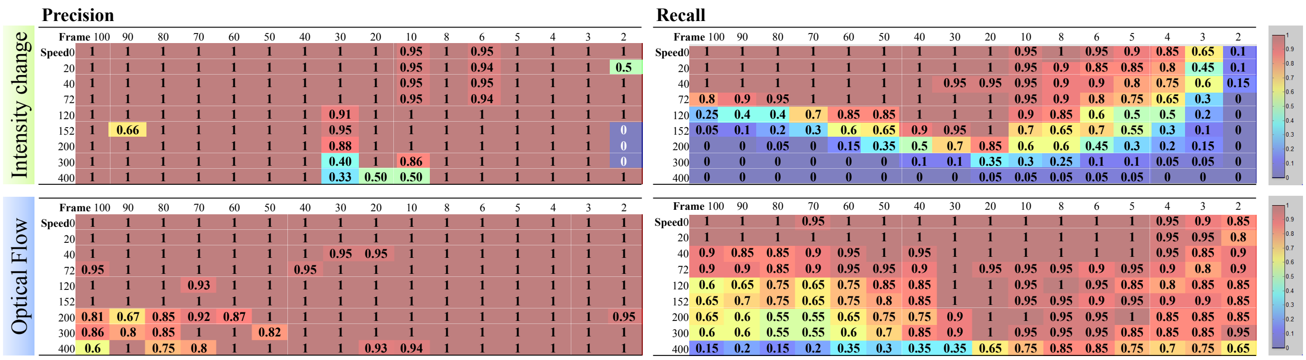


Fig. 15. The precision and recall on detecting raindrops with various raindrop speed and detection latency (Fig. 13). The detection threshold was fixed for all the data. The normalized feature threshold was set to 0.4 for the intensity change, and 0.3 for the optical flow. The raindrop roundness threshold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

iment, we investigated how significant the absence of textures influences the detection accuracy.

In this experiment, the threshold for normalized features was set to 0.4 for the intensity change while 0.1 for the optical flow. The smoothness was set to 2.5, and features were accumulated over 100 frames. As illustrated in Fig. 16, we performed Gaussian blur on the scene, with σ varying from 0 to 10, and generated synthetic raindrops with a fixed size (40 pixels) and position.

As illustrated in Fig. 17, when the scene was textureless, the intensity change was affected. The non-raindrop areas changed less on a less textured scene. The optical flow, however, was not affected. The precision recall is listed in Fig. 18, which shows that when $\sigma > 5$, the accuracy of the intensity change based method dropped because the feature on a textureless scene was less distinctive, and the false alarm rate increased.

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

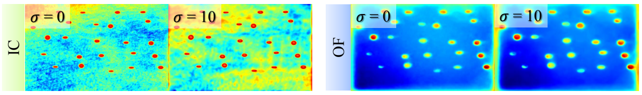


Fig. 17. The accumulated feature using intensity change and optical flow on textured and textureless scenes. 100 frames are used for accumulation.

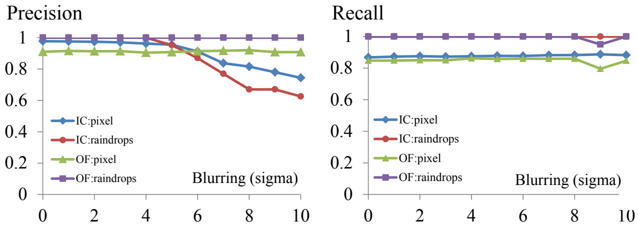


Fig. 18. The precision and recall of raindrop detection on textured and textureless scenes. The threshold for normailzed features was set to 0.4 for the intensity change and 0.1 for the optical flow. The raindrop parameter was set to 60 pixels to 160 pixels. The roundness threshold was set to 2.5π. Features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

7.2 Quantitative comparison with existing methods on detection

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

We compared 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 pan-tile camera and were, thus, not included in the comparison. The results are shown in the last two columns of Fig. 19.

We used 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 method outperformed the existing methods. By combining IC with OF, we get the best performance to sensitively detect all of the raindrops, (because of IC) while keeping a low false alarm rate (because of OF). The detection using the intensity change performed 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 achieved 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 was stuck into a fixed location. Our method outperformed the existing methods in the all three situations as shown in the figure.

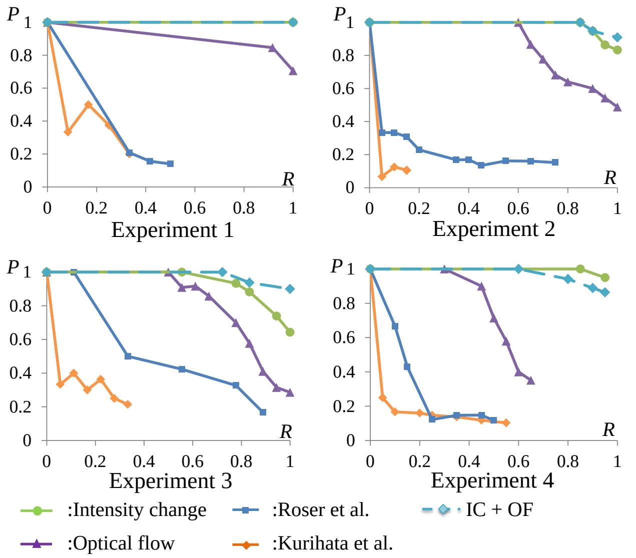


Fig. 20. The precision(R)-recall(R) curves of our methods and the two existing methods. The thresholds of our normalized features are labeled.

7.3 Raindrop Removal

Quantitative Tests on Raindrop Removal As illustrated in the first two columns of Fig. 22, the synthesized raindrops were generated on a video, and used as an input. Our method was compared with the method proposed by Wexler *et al.* [21]. In [14], there is insufficient description for the removal algorithm and thus it was 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 ran each of them on 100 continuous frames and calculated the average error per pixel for each frame. The same as Wexler *et al.* [21], the error was 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 situation. Using the same computer, our method needed 5 seconds per frame to remove raindrops, and Wexler *et al.*'s needed 2 minutes.

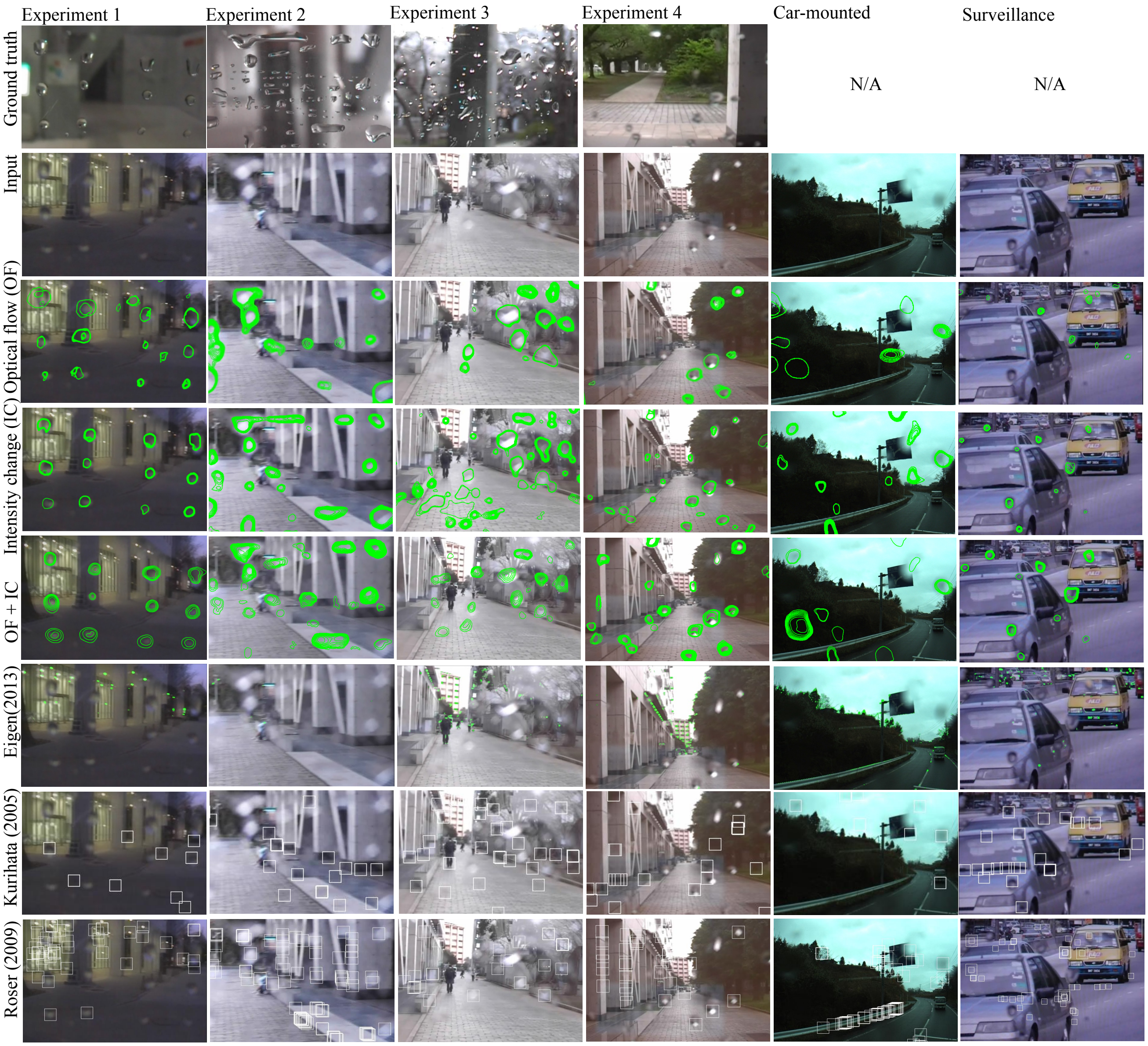


Fig. 19. The detection experiment using our methods and the existing methods.

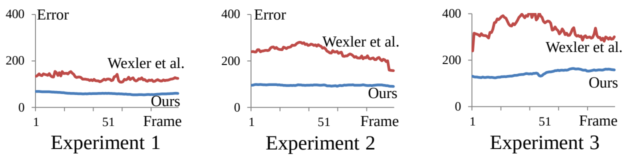


Fig. 21. The average $(R;G;B;dx;dy;dt)$ error on recovering 100 continuous frames of the experiments shown in Fig. 22.

Quantitative evaluation 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 row of Fig. 23 we can see the significant improvement. To demonstrate the performance of our raindrop removal method, the manually labeled raindrops were also included.

Overall Evaluation 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 the significant visibility improvement. [1]

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 key idea on raindrop

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

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

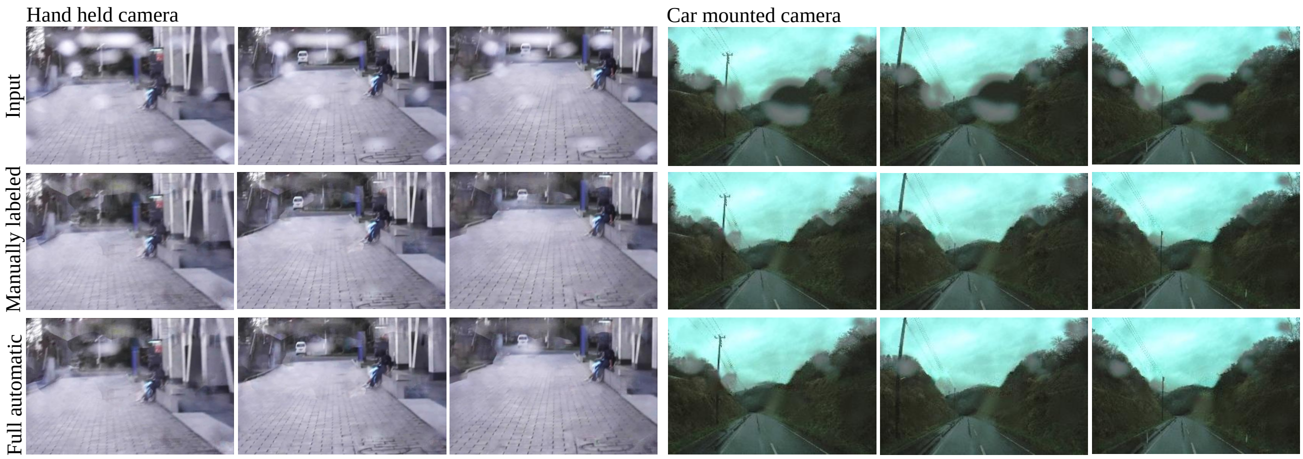


Fig. 23. The raindrop removal using the our method. First row: the input sequence. Second row: the removal result with the raindrops manually labeled. Third row: the removal result with the raindrops automatically detected.

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 suffer from adherent raindrops.

Acknowledgment

This research is granted by

- 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).
- Next-generation Energies for Tohoku Recovery (NET), MEXT, Japan.

References

[1] R. Tan, “Visibility in bad weather from a single image,” *CVPR*, 2008.

[2] R. Fattal, “Single image dehazing,” *SIGGRAPH*, 2008.

[3] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *CVPR*, 2009.

[4] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, “Efficient image dehazing with boundary constraint and contextual regularization,” *ICCV*, 2013.

[5] P. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” *IJCV*, vol. 86, no. 2-3, pp. 256–274, 2010.

[6] K. Garg and S. K. Nayar, “Vision and rain,” *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.

[7] K. Garg and S. Nayar, “Photometric model of a rain drop,” *Technical Report, Columbia University*, 2003.

[8] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–528, 2004.

[9] L. Kang, C. Lin, and Y. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *TIP*, vol. 21, no. 4, pp. 1742–1755, 2012.

[10] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," *ICCV*, 2013.
[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: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2005.
[12] C. Zhou and S. Lin, "Removal of image artifacts due to sensor dust," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 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 Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 144, 2009.
[14] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," *ICCV Workshops*, 2009.
[15] M. Roser, J. Kurz, and A. Geiger, "Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves," *ACCV*, 2010.
[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," *IEEE Intelligent Vehicles Symposium*, 2005.
[17] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," *ICCV*, 2013.
[18] A. Yamashita, Y. Tanaka, and T. Kaneko, "Removal of adherent water-drops from images acquired with stereo camera," *IROS*, 2005.
[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," *IROS*, 2009.
[20] T. Hara, H. Saito, and T. Kanade, "Removal of glare caused by water droplets," *Conference for Visual Media Production*, 2009.
[21] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," *CVPR*, 2004.
[22] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video repairing: Inference of foreground and background under severe occlusion," *CVPR*, 2004.
[23] J. Jia, Y. Tai, T. Wu, and C. Tang, "Video repairing under variable illumination using cyclic motions," *TPAMI*, vol. 28, no. 5, pp. 832–839, 2006.
[24] G. Sapiro and M. Bertalmio, "Video inpainting under constrained camera motion," *TIP*, vol. 16, no. 2, pp. 545–553, 2007.
[25] T. Shiratori, Y. Matsushita, S. B. Kang, and X. Tang, "Video completion by motion field transfer," *CVPR*, 2006.
[26] M. Liu, S. Chen, J. Liu, and X. Tang, "Video completion via motion guided spatial-temporal global optimization," *ACMM*, 2009.
[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," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
[29] M. Subbarao, "Depth recovery from blurred edges," *CVPR*, 1988.
[30] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *TPAMI*, vol. 33, no. 5, pp. 978–994, 2006.
[31] O. Stanley and F. Ronald, "Level set methods and dynamic implicit surfaces." *Springer-Verlag*, vol. ISBN 0-387-95482-1, 2002.

PLACE PHOTO HERE

Shaodi You Biography text.

PLACE PHOTO HERE

Robby T. Tan Biography text.

PLACE PHOTO HERE

Rei Kawakami Biography text.

PLACE PHOTO HERE

Yasuhiro Makaigawa Biography text.

PLACE PHOTO HERE

Katsushi Ikeuchi Biography text.