HAL
open science

Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing

Fawzi Nashashibi, Raoul de Charette, Alexandre Lia

To cite this version:

Fawzi Nashashibi, Raoul de Charette, Alexandre Lia. Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing. International Conference on Control, Automation, Robotics and Vision : ICARV'2010, Dec 2010, Singapour, Singapore. pp.P0655. hal-00558591

HAL Id: hal-00558591

https://minesparis-psl.hal.science/hal-00558591

Submitted on 23 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing

Fawzi Nashashibi, Raoul de Charette, Alexandre Lia
Laboratory of *Robotics Centre*
Mines ParisTech
Paris, France
fawzi.nashashibi@ensmp.fr ; raoul.de_charette@ensmp.fr

***Abstract*—In a scene, rain produces a complex set of visual effects. Obviously, such effects may infer failures in outdoor vision-based systems which could have important side-effects in terms of security applications. For the sake of these applications, rain detection would be useful to adjust their reliability. In this paper, we introduce the problem (almost unprecedented) of unfocused raindrops. Then, we present a first approach to detect these unfocused raindrops on a transparent screen using a spatio-temporal approach to achieve detection in real-time.**

We successfully tested our algorithm for Intelligent Transport System (ITS) using an on-board camera and thus, detected the raindrops on the windscreen. Our algorithm differs from the others in that we do not need the focus to be set on the windscreen. Therefore, it means that our algorithm may run on the same camera sensor as the other vision-based algorithms.

***Keywords*—raindrop, rain, adverse condition, ITS, ADAS**

I. INTRODUCTION

Nowadays, outdoor vision-based detection systems are used for many various applications. Most of the times these systems are designed to work in clear weather condition which is obviously not always the case (i.e. rain, snow, hail, fog, etc.). Indeed, under adverse conditions perception of the scene may be severely damaged. Such scenarios may infer failures in algorithms' process which could have huge consequences when dealing with security applications (surveillance or vision-based application in vehicle for instance). Since it is yet utopian to develop algorithms working under all kind of weather conditions, a first approach could be the detection of adverse conditions using the same vision sensor. By doing so, automatic systems could evaluate the reliability of their own vision-based algorithms in case of adverse conditions.

Among all kinds of adverse conditions, rainy weather is certainly the most frequent. Yet, few researches were conducted on rain detection. The reason is probably that detecting rain (and generally speaking, detecting weather condition) is still a challenging task in computer vision. Due to the high speed of rain falling and the integration time of camera it is usually impossible to distinguish each *raindrop*. Except when using very high speed camera, raindrops appear as *streaks* (or "*ripples*") [1]. However, some detection systems use camera installed behind a transparent screen and therefore it is usually easier to detect drops *on the screen* rather than in the air. People commonly refer to this process as *raindrop detection*. So far, very few researches address this problem and it is usually assumed that focus is set *on* the screen. With the exception of multi focal camera, such constraint prevents the use of the same camera sensor for other detection tasks which most of the time need the focus to be set behind the screen.

In this paper we present a novel method to detect "*unfocused raindrops*" in grayscale video sequences using low level image processing[1]. We applied our algorithm to an on-board vehicle application in order to count raindrops and estimate the amount of rain in the scene. Because we do not need the focus to be set *on* the windscreen, other detection algorithms may run with the same camera sensor and our method can even be used to decrease the reliability of these vision-based processes whenever rain is detected. Outline of the paper is as follows: Section II presents previous works about vision detection in adverse conditions. Section III is dedicated to the problem of unfocused raindrop detection while Section IV describes our detection method. Finally, Section V shows the first experimental results of our algorithm for on-board camera application in urban scene.

II. STATE OF ART

Detection of unfocused (blurred) raindrops is almost unprecedented but several researches were conducted on vision in bad weather starting in the very late 90s. A good general study was written by Narasimhan and Nayar [2]. However, the first studies on vision in bad weather such as [2] or [3], focused almost exclusively on foggy conditions. Indeed, fog is certainly the easiest since due to the very small size of the particles in the air the visual effect is much easier than when dealing with hail, snow, or even rain. Starting from 2004, several researches attempted rain detection in video sequences. To do so, various approaches were proposed depending on the application (stationary, mobile), the type of elements detected (drops, streaks, waves) or the technique that was used (mainly image processing or learning processes).

Garg and Nayar, probably the most active researchers in the field of bad weather recognition (as well as Narasimhan),

[1] *For clarity purpose let's remind to the reader that what is called "unfocused raindrops" are the blurred raindrops on a transparent screen.*

proposed in [4] to use dynamics of rain to detect rain streaks in video sequences, as a binary randomly oriented field. Using intensity property of rain streaks (see Section III for further explanations), they detected potential rain pixels with a temporal filter selecting only pixels which change through the time to a brighter value. By applying linear photometric constraint with spatial and temporal correlation, only streaks with the same orientation are validated and form the final segmented rain regions. It is fair to say that this first research is really original. However due to the long temporal integration (30 frames) it is not suitable for mobile camera system. Others algorithms to detect rain streaks in video sequences were also published. References [5] and [6] for instance, use k-means clustering to identify peaks in the intensity histogram of each pixel and therefore identify which pixel have been modified by raindrops. Indeed this method is not suitable in our case because even though [6] improves the method proposed by [5], histogram analysis is done respectively on the entire video and the foregoing parts in video, therefore assuming that camera didn't move. In case of mobile camera or highly dynamic scene this approach can't be used.

With the development of Intelligent Transportation System (ITS), new methods were designed in order to achieve the detection task using a camera installed inside the vehicle. Whereas for most of the other rain detection applications it is possible to assume that the camera will be stationary or moving very slowly, such assumption is obviously impossible for ITS applications. Since vehicles may be running at high speed or in a highly dynamic environment, all methods using video replay as well as long temporal integration are inadequate. Indeed, rain detection for in-vehicle system uses detection of the raindrops *on the windscreen* rather than detection of the raindrops *falling in the air*. It is of course easy to understand that these are two very different problems. Several approaches attempted the use of either learning processes or image processing. For instance in [7], Kurihata et al. proposed to use PCA (Principal Component Analysis) using eigenspace features extracted from raindrop images in the learning stage. At run time template matching is used to determine whether a region is or not a drop on the windscreen. This method shows really good results but only when detection area is restraint to sky which is very constraining for a vehicle application. Conversely, [8] uses image processing to detect edges through segmentation of the Sobel mask whilst [9] proposal is to use a special optical device (camera + near infrared light) to obtain a clear and precise image of raindrops on the windscreen.

So far, methods which have been developed for ITS in the past years [7-9] seem to work quite well but they do require either a special optical device [9] or the focus to be set on the windscreen [7],[8]. However, ITS already have many applications using camera sensor (such as road lanes detection or obstacles detection) and both for the sake of their performance as well as for industrial constraints, the camera is usually installed behind the interior rear-view mirror. Since the camera is very close to the windshield, it is thus impossible to distinguish drops as *focused objects* which make methods described previously not suitable. So far, only Leleve et al. proposed a method for detecting *unfocused* raindrops using a histogram analysis but no further information is provided as they applied a patent [10].

Despite the lack of previous works on unfocused raindrop detection, other algorithms would benefit from such functionality. Indeed, it would mean that no additional devices would be required and that reliability of other detection algorithms could be estimated depending on the fact that the image is modified by rain visual effect.

III. Problem

A. Appearance of unfocused raindrop

The appearance of a raindrop has been already precisely described in [11]. In their paper Nayar and Narasimhan notice that raindrops have complex appearance and are thus challenging to detect with computer vision processes. However, whilst *stationary raindrops* or falling raindrops (producing the so-called "rain streaks") have been extensively described [1],[12], appearance of unfocused raindrops have not yet been described. Nonetheless, when dealing with *unfocused raindrop* visual effects produced are quite different.

As illustrated in Fig. 1, it is somewhat impossible to distinguish precisely unfocused raindrops. Two observations can thus be drawn. It is distinctly visible in this figure that when focus is set on the windscreen raindrops appear as objects well-separated from their background, whereas unfocused raindrops are not separated from their background and seem quite complicate to locate precisely in the image.

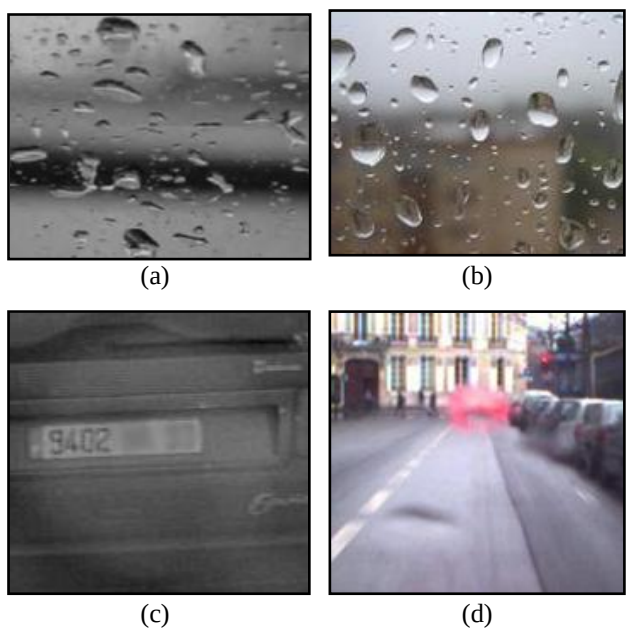


Figure 1. Appearance of focused (a) (b) and unfocused (c) (d) raindrops on a windscreen in grayscale images (a) (c) and color images (b) (d). *Notice that for privacy purpose, registration plate was erased* manually.

Since they hit the windscreen it is no more possible to do a spherical approximation of raindrops' shape as for the raindrops in the air [13], which means that appearance model of a spherical raindrop proposed in [12] is thus not suitable for these raindrops. However, some properties may remain true. As an example, our experiments showed that the brightness of a

raindrop on the windscreen tends to be brighter than the background it occludes. Of course these observations should be proved further with physical experiments.

The specificity of *unfocused raindrop* is such that unlike focused raindrop it isn't possible to see clearly the scene reflection inside the raindrop but part of the reflection may still be seen. Indeed, the scene can be seen *through* the raindrop but the visual effect produced will strongly depend on the position of the camera regarding the windscreen. With a camera distant enough from the windscreen, unfocused raindrops are quite small, quite easily located and look like simple blurred raindrops thus with no net contours. On the other hand, a camera close to the windscreen will show relatively big drops which have even less noticeable contours and are more complicate to locate (even for a human). Let's also add that obviously, in both cases the scene through the unfocused raindrops appears to be blurred.

B. *Unfocused raindrop and Computer Vision*

In terms of computer vision, when a raindrop hit the windscreen (within the visible area, of course) visual effect produced by the apparition of this unfocused raindrop is such that local property of the image changes slightly yet suddenly. Temporal analysis of this local region show us that indeed intensity increased locally while in the same time the scene seen through the drop seems to be blurred which means that sought gradients (and so edges) inside the drop are smoother. Both are of course interesting properties we used for detecting these raindrops.

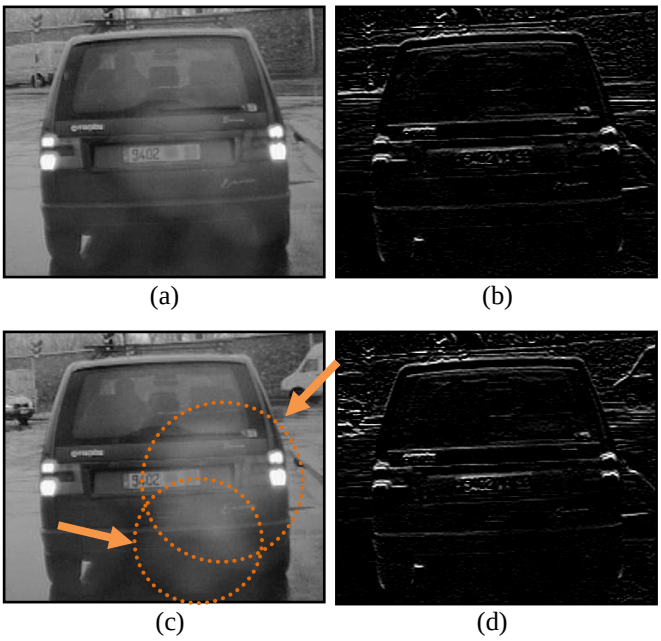


Figure 2. Visual effect produced by apparition of two unfocused raindrops. (a) and (c) are cropped pictures of frame *n-1* and *n*, respectively. Right before raindrops hit the screen (*n-1*) and right after they did (*n*). (c) and (d) are the gradient result of respectively (a) and (c) using Sobel horizontal operator with aperture size set to 3. Let us notice that in the meanwhile the previous raindrop visible in frame *n-1* (over the registration plate), fainted (or merged) and is no more visible in frame *n*. *Reader will notice that new raindrops have been circled in light orange and constrat of gradient enhanced only for readibility purpose.*

As an example, Fig. 2 shows picture from the camera we used in our system, before and after raindrops hit the windscreen. Due to the short distance separating our camera from the windscreen (few centimeters) raindrops produce big marks on the image but are yet hardly noticeable.

C. *Dynamics of unfocused raindrop on windscreen*

Whereas falling raindrops in the air produce traces (rain streaks) due both to their falling velocity and the camera opening duration, raindrops on windscreen often stay stationary. Although, there are several cases producing drops movement. Namely, when driving at high speed the air flow on the car may lead to some drops moving on the windscreen. Likewise, drops which are heavy enough may also slide on the windscreen surface and lead to some drops stream (this is the case with low density rain [14], or when several drops collide thus forming one heavy drop). Conversely, if raindrops are really small, instead of sliding they will slowly dry with time and thus become weaken until they are no more visible.

With our system, experiments showed that the stationary approximation of raindrop (on windscreen and thus in image) is valid during few frames, since raindrops didn't move at all in any of our sequences. Fig. 3 illustrates this property with the apparition of a drop and its evolution through time. Let us also add that wipers were running during our test which means that raindrops maybe didn't have enough time to slide.

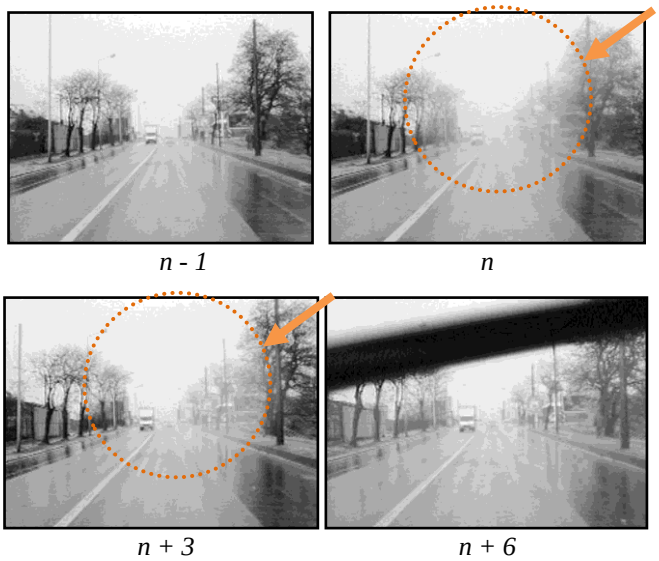


Figure 3. Evolution of an unfocused raindrop through the time stating at *n*. Frame *n+6* shows the wipers currently removing raindrops on windscreen. *Raindrops are circled in light orange and image cropped for readibility.*

IV. METHODOLOGY

A. *Overview*

Using the properties observed and described in the previous section we developed a novel methodology to detect unfocused raindrops at *run-time*. Since we showed that appearance of raindrops is complex and strongly dependent on the background, our algorithm is based both on spatial and temporal analysis. Thus, it could be presented as three stages depicted below in Fig. 4. The whole process is applied in

real-time on the *greyscale* video sequences acquired from an on-board camera in a *dynamic* urban environment.

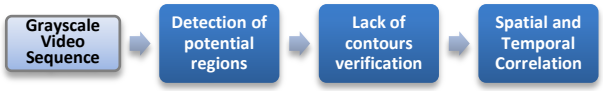


Figure 4. Overview of our method to detect unfocused raindrops.

First stage. Detection of potential regions is achieved by segmenting image into *raindrop* and *non-raindrop* regions using *a priori* knowledge about intensity variation.

Second stage. Potential regions for raindrops are filtered and only those which verify a lack of contours are validated.

Third stage. Spatial and temporal correlation in continuous space-time volume to validate the previously validated regions.

B. Stage 1: Detection of potential regions

As described in Section III, due to the refraction of light, the apparition of a raindrop produces a local variation of intensity in the region concerned. Thus, we use this property to classify pixels of an image into *rain* and *non-rain* pixels. The constraint (1) is therefore applied (*n* being the index of the frame when the raindrop hit the windscreen, and *p1* a threshold parameter set experimentally).

$$\Delta \mathrm{I} = I_n - I_{n-1} \geq p1 \qquad (1)$$

However, since our algorithm runs in highly dynamic environment, intensity variations may also have been produced by objects moving at high speed. Indeed, we observed that road surface markings often verify (1). By studying the local intensity of these "*false regions*", we noticed that intensity often decrease between frame *n* and frame *n+1*; whereas, in case of a raindrop the intensity stay roughly the same on the next frames. Thus, we introduce an additional constraint (2) which verifies that the intensity variation does not result from a fast moving object.

$$\Delta \mathrm{I} = I_n - I_{n+1} \leq p2 \qquad (2)$$

The result of constraints (1) and (2) is a greyscale map where all pixels potentially belonging to a raindrop have non-zero value. Fig. 5 shows that the regions corresponding to new raindrops are segmented correctly though *non-raindrop* regions have been segmented too.

In order to reject some of the *non-raindrop* regions, we extract connected components in the output map using a contours extractor based on 8-connectivity. Then, the latter are filtered out regarding several geometric criteria. One of the criteria being the roundness of the connected component, since we showed in Section III that unfocused raindrops appears to be circular on windscreen. Thus, the ratio between the perimeter and the area should be close to the ratio of a circle. Which means that connected component CC should verify constraint (3).

$$\left| \frac{\mathrm{CC_{Area}}^2}{\mathrm{CC_{Perimeter}}} \right| \leq p3 * 4 * \pi \qquad (3)$$

(a) (b)

Figure 5. Result of the detection of potential regions. (a) being the source sequence used and (b) the result of filtering of the extracted components.

Equation (3) and other constraints about minimum and maximum size are applied to reject some of the extracted connected components which can't obviously be a raindrop.

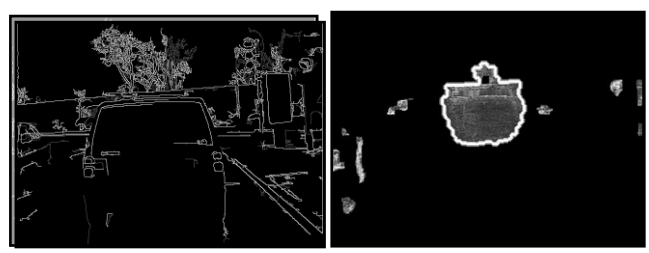
After some experiments parameters were set to the following values: $p1 = 10 \quad p2 = 40 \quad p3 = 2.7$

C. Stage 2: Lack of contours verification

At this stage, we thus know that any validated region verifies intensity variation and geometrical properties of an unfocused raindrop. However, when dealing with dynamic environment these variations and properties may be the result of any type of object motion. For instance, intensity variation due to slow motion object may have been segmented in the first stage since constraint (2) is not suitable to filter such cases.

To reject these regions we use the *a priori* knowledge about unfocused raindrops appearance. In paragraph III.A, we observed that raindrops have no net contours. Thus, connected components which do correspond to a raindrop should also present a lack of contours. To verify this property, we compute the sum of edges detection of frame *n-1* and frame *n*. Then, we subtract the sum of edges detection from the map of connected components.

Using the result of this comparison we validate only connected components which do not share too many edges with the map output by edges detector. Result of this operation is illustrated in Fig. 6.



(a) (b)

Figure 6. *Illustration of the verification using lack of contours property. (a) is the sum of contours in frame* n-1 *and* n. *Gray edges belong to frame* n-1 *while white ones are from frame* n. *(b) As a result of the substraction, only the connected component surounded by white contours will be validated for further stages.*

To achieve edges detection we use the *Canny* detector which has been pointed out as being a good method for detecting edges "*especially for seriously blurred images*" [15]. Thus, we ensure that edges of objects are not missed because of some motion blur.

D. Stage 3: Spatial and temporal correlation

As the reader may have noticed, during Stage 1 and 2 we compared the same pixels regions of the input image through time without having any guarantees that the regions we compare correspond to the same part of the scene. Actually, since our system deals with highly dynamic environment (such as urban scene with fast moving objects) we may have been comparing different parts of the scene. Obviously, such comparison is a non-sense and will infer failures in our algorithm. For the sake of the detection reliability we need to ensure that the regions compared through time were really comparable. In case they were not, the result of previous stages should simply be rejected.

In order to overcome this issue, we use alternative approach based on spatio-temporal analysis. That is to say that we compute a visual signature for each validated connected component *CC* in frame *n*. Then, we try to find the closest region *CC'* in frame *n-1* and *n+1* with an equivalent visual signature as the one computed before. Obviously, if CC is a visual effect due to the apparition of a raindrop on the windscreen then comparison of the intensity from CC'_{n-1}, CC_n and CC'_{n+1} should also verifies the constraints described in Stage 1.

Because spatio-temporal correlation is a huge time-consuming process, it is only applied as a final verification stage. Furthermore, several restrictions (such as maximum neighbor distance and 5x5 clusters) are used to make this process fast enough for real-time application.

V. Experimental results

A. Performance

To measure the performance of our algorithm we applied it on video sequences we acquired using an on-board camera mounted behind the interior rear-view mirror of our prototype vehicle. Video sequences are 640x480 greyscale images and were recorded in dynamic urban environment, in Paris.

In addition to the algorithm presented above, we used a basic decision scheme to output rainfall judgement. This decision scheme is based on the number of raindrops detected during the last 150 frames and outputs one of the following states: *no rain*, *light rain*, *medium rain*, or *heavy rain*.

So far, our algorithm was tested on a sequence of 24 minutes, with 7 minutes of rain. In the first 7 minutes our algorithm correctly detects raindrops and thus output *medium rain* and *heavy rain*. Conversely, the rest of the video output the *no rain* state which is in fact correct. Fig. 7 is a set of screenshots of our algorithm output when evaluated on our urban sequence.

B. Discussion

The first results we achieved are quite promising since our algorithm was evaluated in highly dynamic scene. However, we believe that our algorithm may suffer from a lack of flexibility and thus need improvements in order to be fully reliable. As we explained in Section III, unfocused raindrop detection is very challenging and during our experiments we noticed that even humans had serious issues to detect and locate the raindrops in the video sequences.

About the intrinsic performances of our algorithm, we noticed several causes of non-detection of the raindrops. For instance, detecting raindrops over bright background seems almost impossible since the result of intensity variation is barely unnoticeable. Indeed, parameters may be adjusted to decrease the number of "missed" raindrops but doing so may lead to some false positive.

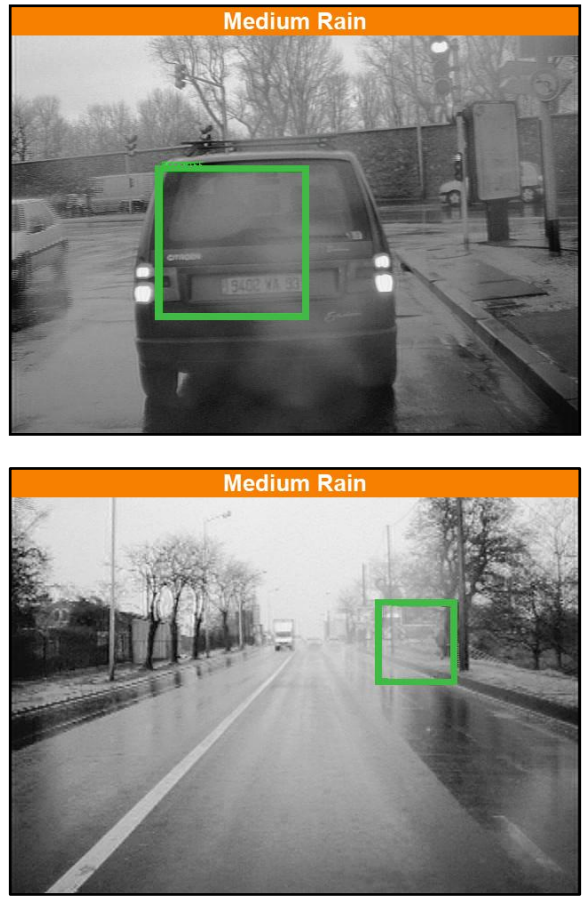


Figure 7. *Screenshots of our algorithm of unfocused raindrop detection in urban environment.*

VI. Conclusion

In this paper we proposed an original method for the detection of unfocused raindrop. Our method does not require any specific device or the focus to be set on the windscreen. Important aftermath is that unlike almost all previous proposals for raindrop detection on the windscreen, the method proposed can be used on the same cameras as the one used for other detection tasks. The results we published here are promising but our method still suffers from some lacks. However, this is a first proposal for detecting unfocused raindrops without modifying camera characteristics for this specific application.

In the future, we should improve the detection rate of our method and also quantify the performance of our already existing algorithm. Even though it seems easy, this task is in fact not trivial since even human had issue to detect and locate

raindrops in the sequences. A specific methodology should be found to define some kind of ground truth and thus be able to measure the performance precisely.

ACKNOWLEDGMENT

This research has been conducted within the ICADAC French-German project which objective is to Improve CAmera based Detection under Adverse Conditions. It is funded by French national research agency ANR and German national research agency DFG.

REFERENCES

[1] K. Garg and S.K. Nayar, "Vision and Rain," *International Journal of Computer Vision*, vol. 75, Oct. 2007, pp. 3-27.

[2] S.G. Narasimhan and S.K. Nayar, "Vision and the Atmosphere," *Int. J. Comput. Vision*, vol. 48, 2002, pp. 233-254.

[3] F. Cozman and E. Krotkov, "Depth from scattering," *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 801-806.

[4] K. Garg and S. 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*, 2004, pp. I-528-I-535 Vol.1.

[5] X. Zhang, H. Li, Y. Qi, W.K. Leow, and T.K. Ng, "Rain Removal in Video by Combining Temporal and Chromatic Properties," *Multimedia and Expo, 2006 IEEE International Conference on*, 2006, pp. 461-464.

[6] X. Zhao, P. Liu, J. Liu, and T. Xianglong, "The Application of Histogram on Rain Detection in Video," *Proceedings of the 11th Joint Conference on Information Sciences (JCIS)*, The Harbin Institue of Technology, Shenzhen, China: 2008.

[7] 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," *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, 2005, pp. 205-210.

[8] J. Park, M. Kim, H. Im, K. Lee, and S. Lee, "Development of Vision based Control Smart Windwhield Wiper System for Intelligent Vehicle," *SICE-ICASE, 2006. International Joint Conference*, 2006, pp. 4398-4403.

[9] S. Gormer, A. Kummert, Su-Birm Park, and P. Egbert, "Vision-based rain sensing with an in-vehicle camera," *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 279-284.

[10] J. Leleve, A. Bensrhrair, J. Rebut, and G. Challita, "Method of detecting rain on a windscreen," U.S. Patent EP1860426, November 28, 2007.

[11] S.K. Nayar and S.G. Narasimhan, "Seeing Through Bad Weather," *Robotics Research*, Springer Berlin / Heidelberg, 2005, pp. 335-350.

[12] K. Garg and S.K. Nayar, *Photometric model of a rain drop*, New York, USA: Computer Science Department, Columbia University, 2003.

[13] K.V. Beard and C. Chuang, "A New Model for the Equilibrium Shape of Raindrops," *Journal of the Atmospheric Sciences*, vol. 44, Jun. 1987, pp. 1509-1524.

[14] J. Marshall and W.M. Palmer, "The Distribution of Raindrops with Size," *Journal of the Atmospheric Sciences*, vol. 5, Aug. 1948, pp. 165-166.

[15] S. Wu, W. Lin, S. Xie, Z. Lu, E.P. Ong, and S. Yao, "Blind blur assessment for vision-based applications," *Journal of Visual Communication and Image Representation*, vol. 20, May. 2009, pp. 231-241.