Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Dynamic Adherent Raindrop Simulator for Automotive Vision Systems

Yazan Hamzeh[1,3], Zaid El Shair[1], Abdallah Chehade[2], and Samir A. Rawashdeh[1]

[1]Department of Electrical and Computer Engineering, University of Michigan-Dearborn: Dearborn, MI 48128, USA
[2]Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn: Dearborn, MI 48128, USA
[3]Product Development, Ford Motor Company: Dearborn, MI 48126, USA

Corresponding Author: yhamzeh@umich.edu

ABSTRACT The automotive domain is highly regulated, with many safety-critical aspects to consider. This means that a great deal of testing is required to validate the performance of automotive systems, under all possible environmental conditions. For vision-based systems, camera images are among the most important input sources of information, and using high-quality images is integral to the system's performance. Rain, as a type of adverse weather condition, degrades the image quality which reflects negatively on the vision-based algorithms. Collecting representative sets of data under different rain conditions is required for system testing and performance evaluation. This usually is both costly and time-consuming. Augmenting the sets of real rained images in system testing is an attractive, feasible alternative. In this paper, we present an adherent rain simulator system, that adds simulated rain to clear image frames captured in real drive cycles. We test the quality of simulated rained images against real rained ones, using common image similarity metrics. We also compare the performance of deep learning-based object detectors, using our simulated rained images vs. real rained images. The results show that object detectors show similar performance using simulated and real rained images. A comparative analysis shows that our model produces more realistic raindrops, compared to a ray-tracing-based raindrop simulator.

INDEX TERMS Adherent Raindrops, Automotive domain, Deep-learning, Image degradation, Object detection, Raindrop simulator, Recall, Similarity metrics

I. INTRODUCTION

Most vision-based systems developed for automotive applications assume optimal visibility conditions. Deviations from these optimal conditions usually result in performance degradations or complete failure of vision-based systems. Reduced lighting level, for example, causes performance degradations in intensity-based vision algorithms, and may cause a total system failure in color-based algorithms that are usually more susceptible to illumination level variations. Raindrops that adhere to the vehicle windshield blocks certain zones of the image and introduce lens effects that cause both spatial and dynamic distortions to the image.

The automotive environment is unpredictable in general. Testing vision-based automotive systems, to verify their robustness against noise factors requires collecting a great deal of data, to cover all possible operational conditions. Collecting representative rained image data is not optimal, since both raindrop sample properties and scene background are uncontrollable. It is not possible to control the size and intensity of real adherent raindrops. This means that many datasets of rained images need to be collected, analyzed, and classified based on adherent raindrop characteristics, before being used for robustness testing and system optimization. Lack of background controllability means that the clear-image ground-truth cannot be established, since it is not possible to repeat the exact drive cycle with and without rain, due to variations in background elements in different drive cycles. De-raining of rainy images presents an option for estimating ground-truth, rain-free, data. This, however, is not an optimal solution either. De-raining algorithms cannot remove all existing raindrops in an image with high accuracy and reliability. They also add distortion, in terms of incorrectly de-raining clear sections of a rained image and adding spatial and intensity distortions to the de-rained image. In this paper, we present a rain simulator system, that adds rain to clear images, collected from real drive cycles. The system is dynamic, meaning that it shows the progressive accumulation of adherent raindrops on a vehicle windshield. The amount of rain and rate of accumulation is controllable, to provide the most flexibility for generating test sets at different rain conditions. Moreover, this paper expands on and follows some of the approaches used in our prior work [1] to assess the effects of adherent Rain on deep learning-based object detectors, and compares it with simulated dynamic adherent rain. As a side contribution, we tested the performance of the object detectors using simulated raindrops that were produced by the state-of-the-art ray-tracing raindrop simulator [2, 3] and showed that the generated rainy scenes were suitable for the validation of the object detectors performance.

As an added contribution, we are releasing a dataset of images with real rain and with generated rain, along with the clear image samples that align with them. The sets of varying rain intensity from light to heavy are included in the database. The generated rain sets are augmented with text files that list the positions, sizes, and orientations of the ellipses that encapsulate the generated raindrops in each image. This can serve as ground truth for researchers developing raindrop detection approaches.

The remainder of this paper is arranged as follows. Section II describes some work related to rain simulation. In section III we describe of data collection method and the test environment and metrics we used to evaluate our system. Our adherent rain simulator design is detailed in section IV. Test results are summarized in section V and a discussion of the results is presented in section VI. Section VII includes our conclusion and discussion of future work.

II. RELATED WORK

Halimeh and Roser [2] developed a geometric-photometric model of raindrops on a windshield, coined Raindrop Intelligent Geometric Scanner and Environment Constructor (RIGSEC). The model makes use of Snell's law [3] to track all points in an assumed raindrop image segment to the originating point in the environment. By comparing associating light beams emitted from the environment points that go through the assumed raindrop to those that go through the windshield glass only, a complete geometric model of the adherent raindrop can be established. Fresnel's reflectivity equations [4] are then used to calculate the intensity of raindrops as seen by the camera, adding photometric enhancement to the raindrop model. The RIGSEC model generates adherent raindrops of different sizes and at different positions on the windshield. This model is cohesive and dynamic but since it is based on ray-tracing of each potential raindrop pixel, it is relatively complex and may be computationally expensive.

Alletto et al. [5] used the ray-tracing approach to generate images with synthetic raindrops and used them to train their deep-learning-based de-raining algorithm. They selected special sequences from the DR(eye)VE dataset [6] with cloudy backgrounds and augment them with the generated raindrops.

Hao et al. [7] also used the ray-tracing approach for synthetic raindrop generation and developed a dataset that they later used for their CNN-based de-raining algorithm. They selected special cloudy images from the Cityscape [8] dataset, and augmented them with their synthetically generated raindrops.

To the best of our knowledge, the only publicly available model for generating simulated raindrops is the one developed by C. Carlin [9], which was used to generate synthetic rain datasets in Alletto et al. [5] rain removal system. We used that model to generate rained images dataset, to compare it with our model output.

Stuppacher and Supan [9] developed a model for realistic waterdrops, that can be used in game development and CGI. The authors observed that a certain mass is required for the raindrop to move and that small drops stay idle or get swept away by moving drops. In addition, moving drops tend to lose mass, which slows down the drop motion. Waterdrop viscosity, speed, mass, and adhesion to the surface all play a role in shaping the water drop. Gravitational vector is introduced to the simulation world space, to govern the direction of raindrop moving speed and acceleration on an inclined surface.

Stuppacher and Supan model produces visually realistic rain simulations, that are suitable for CGI and game development applications.

In previous work, the authors of this paper [10] developed a rain simulation system that adds falling rain to clear image sequences. The system allowed for controlled simulated rain intensity and generated monochromic stereo image pairs. This system, however, simulated falling rain streak, rather than adherent rain on the windshield, the latter form being the more dominant source of distortion in vision-based automotive applications.

III. Data Collection and Experiments Setup

For data collection, we used a dual-lens stereo camera (ELP-960P2CAM-V90-VC) that was attached to the vehicle dashboard, approximately 10 cm away from the windshield. We captured around 15 hours of videos of real drive cycles, under clear and rainy conditions, at 60 frames/second rate and 1280X960 resolution per image frame. We wrote an algorithm in MATLAB scripting language to detect the beginning and end of the wipe events. The frames previous to a wipe event were captured as rained image samples, and the few frames right after that event were considered to represent the clear reference images. Figure 1 shows an example of clear (a) and wet (b) images from the data sets. For the comparison with Carlin's model [9], we cropped a set of 500x500 pixel images from the original images, since this image resolution provided the best result from that system. Used the same dataset to generate rained images from our model as well.

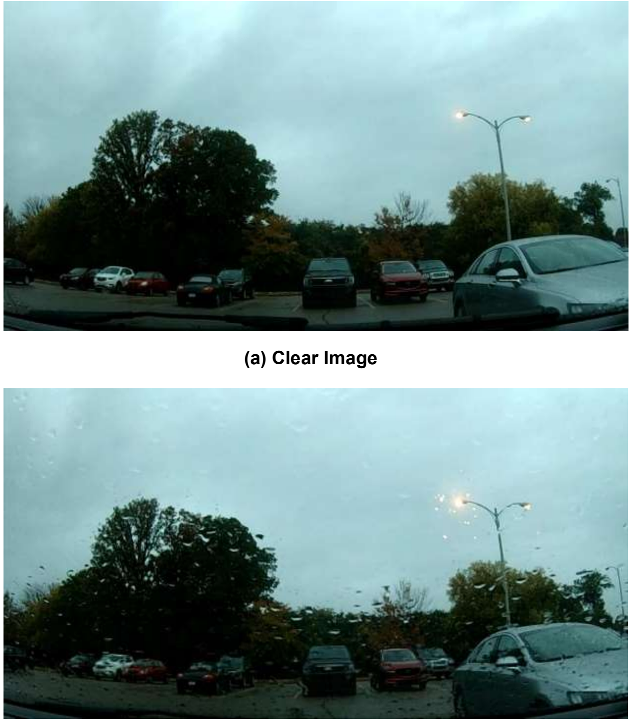
We used two similarity metrics to test the closeness of images with real vs. simulated rain, namely the

Structural SIMilarity (SSIM) index, and the Earth Mover Distance (EMD).

As described by Wang et al. [11], SSIM between two images x and y is given by:

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \quad (1)$$

Where l, c and s are the luminance, contrast, and structure comparator functions, respectively, and $\alpha, \beta,$ and γ are factors that represent the contribution level of each comparator to the overall index. We used the "ssim" function as implemented in MATLAB 2018-b.



(a) Clear Image

(b) Rained Image

FIGURE 1. Example of captured image sets, clear(a) and wet (b).

The EMD "is a measure of distance between two probability distributions over a region D" [12]. In image processing, EMD reflects the cost of moving one image, represented by some feature signature (e.g., intensity histogram), to a reference image, represented with the same signature type. Given two images, Image1 and Image2, one or more features are selected and clustered, to create signatures, [12]

$S = \{(s_1, w_{s1}), (s_2, w_{s2}), \dots, (s_3, w_{sm})\}$, and

$T = \{(t_1, w_{t1}), (t_2, w_{t2}), \dots, (t_m, w_{tm})\}$, for Image1 and Image2, respectively.

s_i, w_{si} for $1 \le i \le m$ and t_j, w_{tj} for $1 \le j \le n$ represent the cluster Ids and weights for the two signatures. A feasible flow $F = [f_{i,j}]$ between the two signatures must satisfy the following constraints:

$$f_{i,j} \ge 0, \forall\ 1 \le i \le m, 1 \le j \le n \quad (2)$$

$$\sum_{i=1}^{m} f_{i,j} \le w_{si}, 1 \le i \le m \quad (3)$$

$$\sum_{j=1}^{n} f_{i,j} \le w_{tj}, 1 \le j \le n \quad (4)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} = \min\left\{\sum_{i=1}^{m} w_{si}, \sum_{j=1}^{n} w_{tj}\right\} \quad (5)$$

Given a ground distance $D = [d_{i,j}]$ between the two clusters, EMD represents the solution that minimizes the work (flow times distance) to move one signature to match the other. Mathematically, this is given by [12]

$$EMD(S,T) = \min \sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}\, d_{i,j}$$

$$= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}\, d_{i,j}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}} \quad (6)$$

In our implementation of EMD, we calculated the histogram of gray-scale image intensity and used it as signatures for the images under comparison. We used U. Yilmaz's [13] implementation of EMD in MATLAB, which was based on the work of Rubner et al. [14].

We tested the quality of the generated simulated rained data sets compared to sets of the real rained images using pre-trained Convolution Neural Networks (CNN) based deep learning models, designed to detect objects in real drive cycles, including YOLOv3 [15], SSD [16], and Faster-RCNN [17].

We developed a MATLAB script to calculate Precision and Recall measures for object detector performance with real and simulated rain input.

IV. ADHERENT RAIN SIMULATOR

We start with the following assumptions while designing our adherent raindrop simulator:

1) Adherent raindrops can take many irregular shapes, but they can be approximated with an ellipse, as a starting point.
2) An Adherent raindrop acts as a lens, adding fish-eye or barrel distortion to the image
3) Adherent raindrops in an image are blurry and lack clear borders that define their shapes.
4) Adherent raindrops are semi-static, in the sense that there is very little observed movement of a raindrop from one frame to the next.

Figure 2 shows the main stages of our raindrop generation process, which are described as follows:

A. SELECT RAINDROP SHAPE, SIZE, AND POSITION

As shown in Figure 3, adherent raindrops can come in different shapes and sizes, and align in any possible orientation. We start with an ellipse to approximate the shape of the adherent raindrop (Figure 4). Subsequent steps distort this ideal elliptical shape, adding more realism to the simulated raindrop shape.

The size, orientation, and position of raindrops in each frame are arbitrarily selected from a calibratable raindrop characteristics table. Table 1 shows the raindrop calibration parameters, with some example ranges.

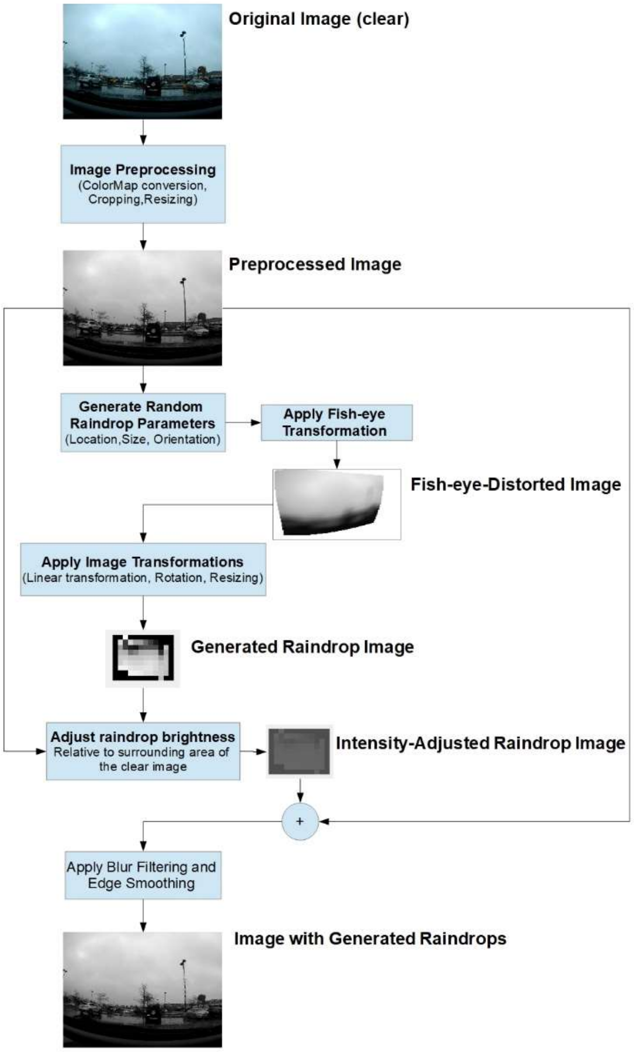

Original Image (clear)
Image Preprocessing
(ColorMap conversion, Cropping,Resizing)
Preprocessed Image
Generate Random Raindrop Parameters
(Location,Size, Orientation)
Apply Fish-eye Transformation
Fish-eye-Distorted Image
Apply Image Transformations
(Linear transformation, Rotation, Resizing)
Generated Raindrop Image
Adjust raindrop brightness
Relative to surrounding area of the clear image
Intensity-Adjusted Raindrop Image
+
Apply Blur Filtering and Edge Smoothing
Image with Generated Raindrops

FIGURE 2. Main stages of raindrop generation include image preprocessing, barrel (fisheye) transformation, raindrop image processing, brightness adjustment, and blurring and edge smoothing.

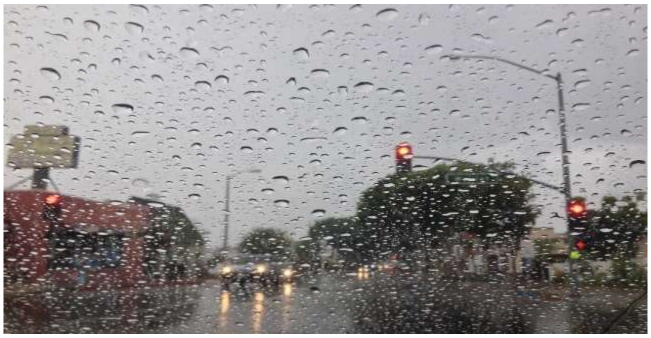


FIGURE 3. Adherent raindrops can come in different shapes, sizes, and orientations. Photo by Good Stock Photos.

Table I
CALIBRATION PARAMETERS FOR GENERATING SIMULATED RAINDROPS FOR EACH IMAGE FRAME

Parameter	Description
DropsPerFrame	Number of raindrops added to a single frame [1-3]
DropPosition	Position of a raindrop (default is the whole image area)
DropRotation	The orientation of a raindrop [80° - 150°]
DropSize	Size of a raindrop defined in terms of major and minor axes of an ellipse in pixels [10-35 x 3-10]

B. APPLYING LENS DISTORTION

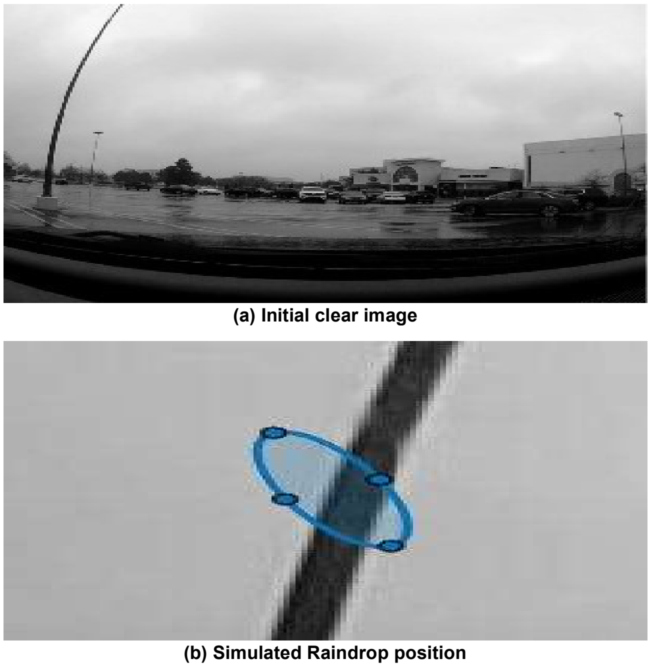
Adherent raindrops on a windshield cause a lens distortion, similar to the fisheye or barrel effect. This distortion can be represented as a nonlinear spatial translation of image points into the raindrop pixels. This translational transformation can be approximated by [18]

$$P_n = P_o + DF * P_o^3 \tag{7}$$

Where P_n is the distorted pixel in the raindrop, P_o is the original (environment) pixel that is influenced by the raindrop distortion, and DF is the distortion factor. We use the MATLAB function "*geometricTransform2d*" to represent this lens distortion effect of a raindrop. Figure 5-a shows the distorted region after applying the lens effect.

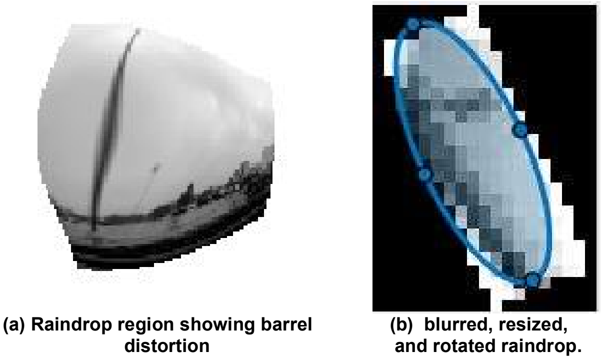
C. BLURRING, RESIZING, ROTATING

Since the vehicle camera used in vision-based applications is usually focused on the environment, any close images, raindrops included, would look blurry [19, 20]. We use the MATLAB function "imfilter" to add the blurring effect to our simulated raindrops. For focus-blurring, we selected the correlation option and set the blur window size to a proper value.


(a) Initial clear image

(b) Simulated Raindrop position

FIGURE 4. Starting with a clear image frame (a), the simulator generates arbitrary values for simulated raindrop location, size, and orientation (b).


(a) Raindrop region showing barrel distortion

(b) blurred, resized, and rotated raindrop.

FIGURE 5. Applying translational transformation on an image produces the barrel effect (a). The distorted region is blurred, resized, and rotated to match desired raindrop characteristics (b).

For motion-blurring, we used the "fspecial" function to create a special filer type, with the 'motion' option, and the X and Y motion-blurring levels set appropriately. This motion type is then used by the imfilter function to add a motion-blurring effect. The parameters for focus and motion blurring were determined experimentally. We then resize and rotate the raindrop image, to approximately match the encapsulating ellipse we have started with. Figure 5-b shows the raindrop region, after blurring and applying the resize and rotation operations.

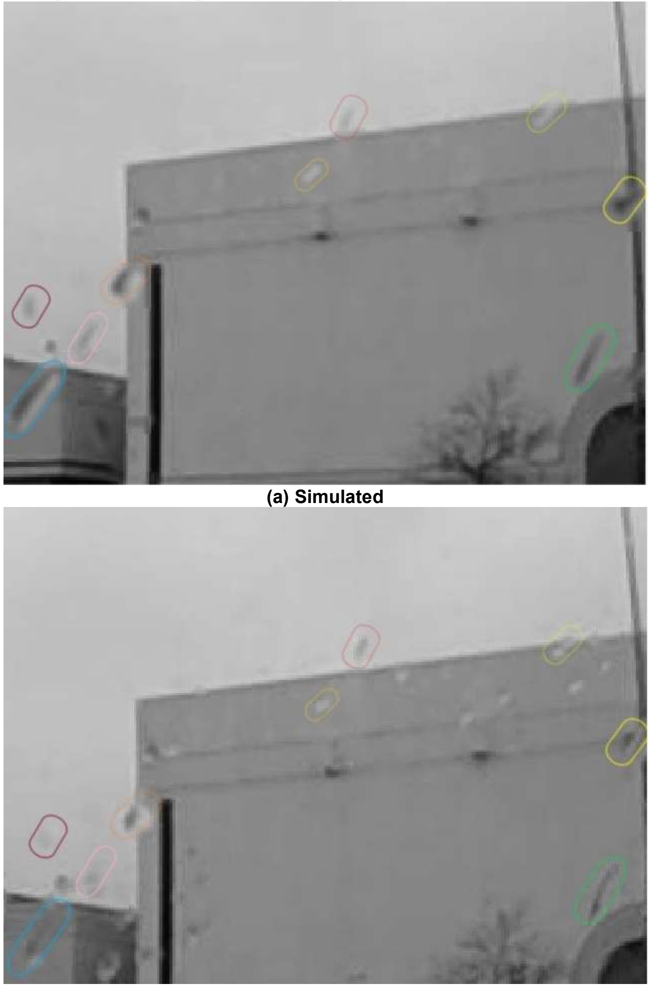
D. ADDING RAINDROP TO IMAGE

Adherent raindrops tend to be slightly brighter than their surrounding background, since they collect light from all areas of the image, due to the lens effect. As stated earlier, raindrops lack strong boundaries that separate them from their background and give them specific shapes. We use intensity adjustment and border dilation and filtering to allow for seamless addition of generated raindrops to the original (clear) image. Figure 6 shows samples of generated raindrops compared to real raindrops in a wet image.

E. CAPTURING ADHERENT RAINDROP DYNAMICS

Raindrops remain adhered to the windshield surface so long as the forces exerted surface tension and gravitational pull are balanced. You et al. [19] found that the observed raindrop speed was around 0.01 pixel/s, as seen by a camera mounted on a vehicle moving at a speed of 30 km/h. They also observed that the motion seen inside a raindrop was 20 to 30 times slower than that seen in other areas of the image. In our raindrop simulator, the raindrop dynamic behavior is implemented as follows:

1) No movement is applied to raindrops from one frame to the next, a reasonable approximation to the quasi-static movement observed by You et al. [19] .
2) New Raindrops are added arbitrarily to the raindrops generated on previous image frames.


(a) Simulated

(b) Real

FIGURE 6. Generated raindrops (top) are visually similar to real raindrops (bottom), as perceived by a human observer.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3103895, IEEE Access

3) If a new raindrop is generated that intersects with an existing one, the distorted area is generated as a simple addition of the two raindrops. This method allows for approximating complex raindrop shapes as a combination of elliptical shapes.
4) For simplicity, the process of refactoring large raindrops to smaller droplets (see Stuppacher and Supan [9]) is not implemented. This simplification holds reasonably well under light-to-moderate raindrop intensity since the size of raindrops does not grow fast, due to the low probability of arbitrary raindrops intersecting over a short period.
5) Raindrops mask is refreshed (all raindrops regenerated) every 20 to 30 frames (programmable), to account for the dynamic changes of background scene elements and, at the same time, making use of You et al. [19] observation about the slow change of raindrop pixels compared to non-raindrop areas.

V. RESULTS AND ANALYSIS

To validate the quality of our generated raindrops against real ones, we started with a clear/rained image set of the same scene. We individually picked raindrops from the rained image and measured their positions, sizes, and orientations. We then used our simulator to generate raindrops with the same characteristics as the real ones. We used SSIM and EMD metrics to measure the level of similarity of our generated raindrops to their real counterparts. We took each rain raindrop image and compared it to the corresponding simulated raindrop, which was generated by using the same orientation, size, and position of the real raindrop image. Figure 7 shows the similarity measure histogram between real and simulated raindrops, as calculated using EMD and SSIM metrics. Figure 8 shows that the similarity level between a real rained image and an image with generated raindrops increases with the addition of extra simulated raindrops.

In the second level of testing, we generated rained images by adding generated raindrops to clear images. Real and simulated rain image frames are then selected based on the degradation level of each image frame as compared to the clear image frame of the same scene. SSIM and EMD metrics were used as indicators of image degradation, in the sense that worse similarity score of these metrics was taken as a direct indication of increased image degradation caused by raindrops. Only “parking-lot” data sets were used in this series of tests, to eliminate any degradation from the movement of the test vehicle, relative to other objects in the scene. The matched real and simulated rained images are then used as inputs to three deep learning-based object detectors, namely Single Shot Detector (SSD), You Only Look Once version 3 (YOLOv3), and Faster Region-based Convolutional Neural Network (RCNN). Detected objects are evaluated and matched, and detection performance is evaluated in two ways:

A. DETECTION CONFIDENCE LEVEL VERSUS IMAGE DEGRADATION LEVEL

The confidence level that is generated by the object detectors and assigned to each detected object (0-100%), is inspected against image degradation (dissimilarity to the clear image), over all dataset frames. This process is repeated for both real and simulated rained images. Figure 9 shows matched image frame pair of real and simulated rain, with objects detected in each image with different confidence levels. Figure 10 shows plots of confidence levels of one object (Object #2) in the image scene, against SSIM and EMD, used as measures of distortion. There is a clear trend of increased confidence level with decreased degradation (less rain) of rained images. This trend is observed in both real and simulated rained images. The error bars represent the mean (center of the bar) and standard deviation (length of the bar) of sample point segments, each segment containing sample points that have the same range of SSIM or EMD scores.

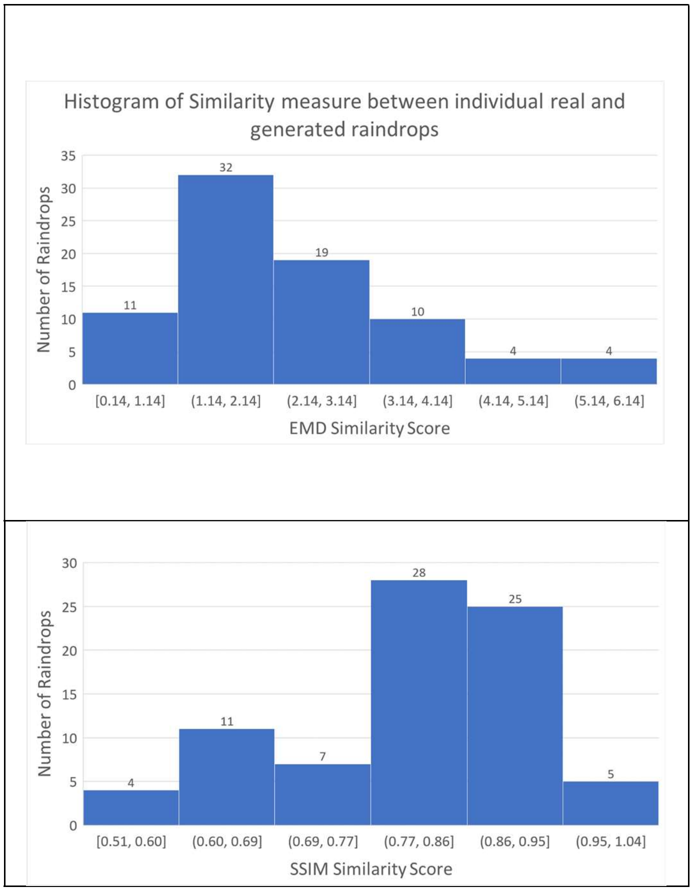
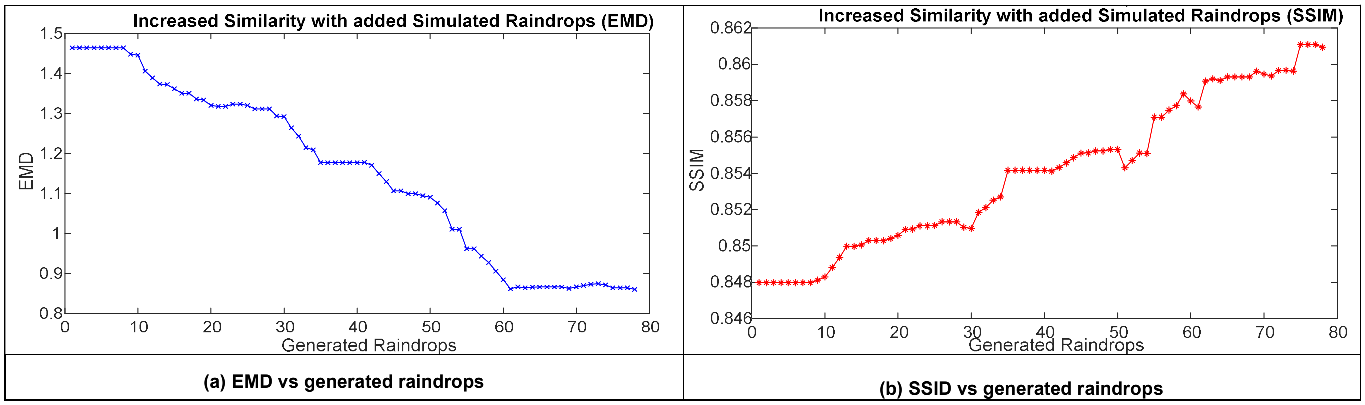


FIGURE 7. The similarity between individual Real and Simulated raindrops is measured using EMD (top) and SSIM (bottom) metrics and the histograms of scores calculated for each metric. The figure shows a strong similarity between real and generated raindrops.

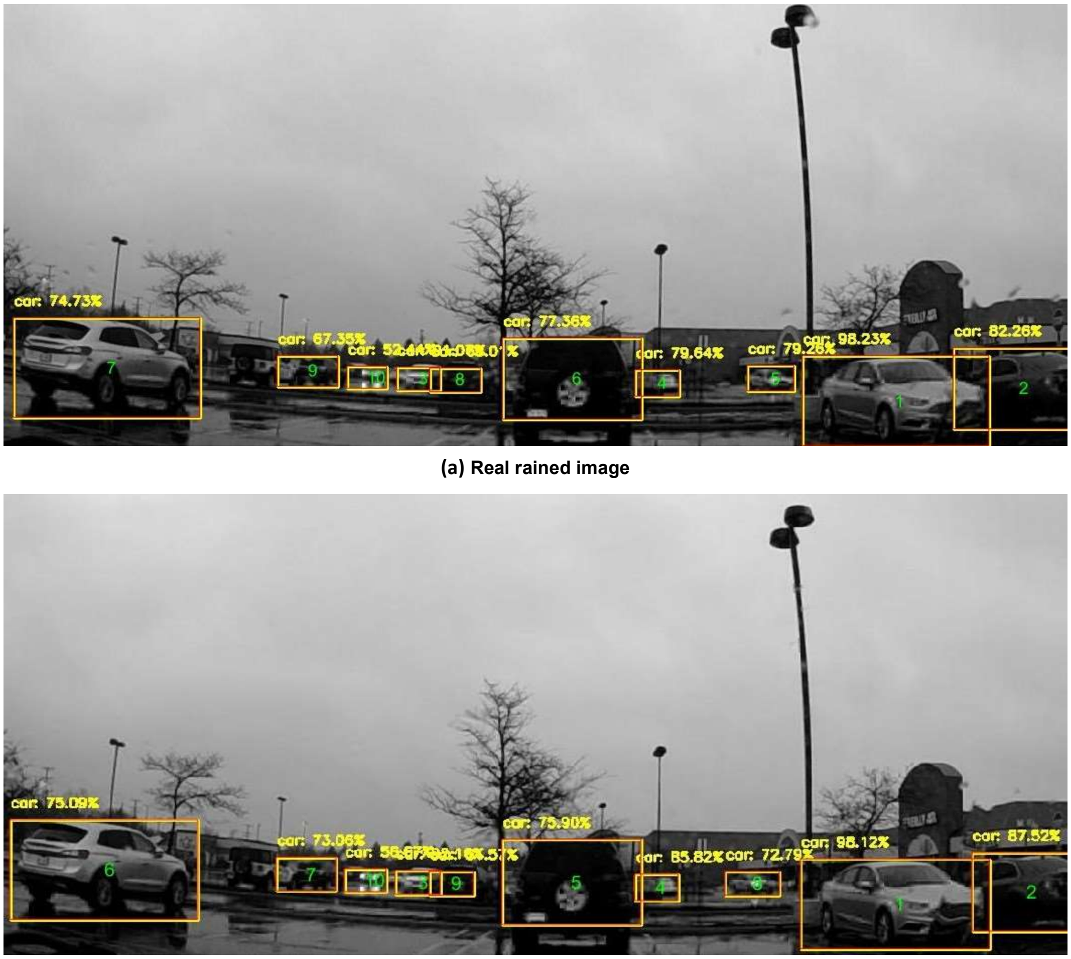
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3103895, IEEE Access



(a) EMD vs generated raindrops

(b) SSID vs generated raindrops

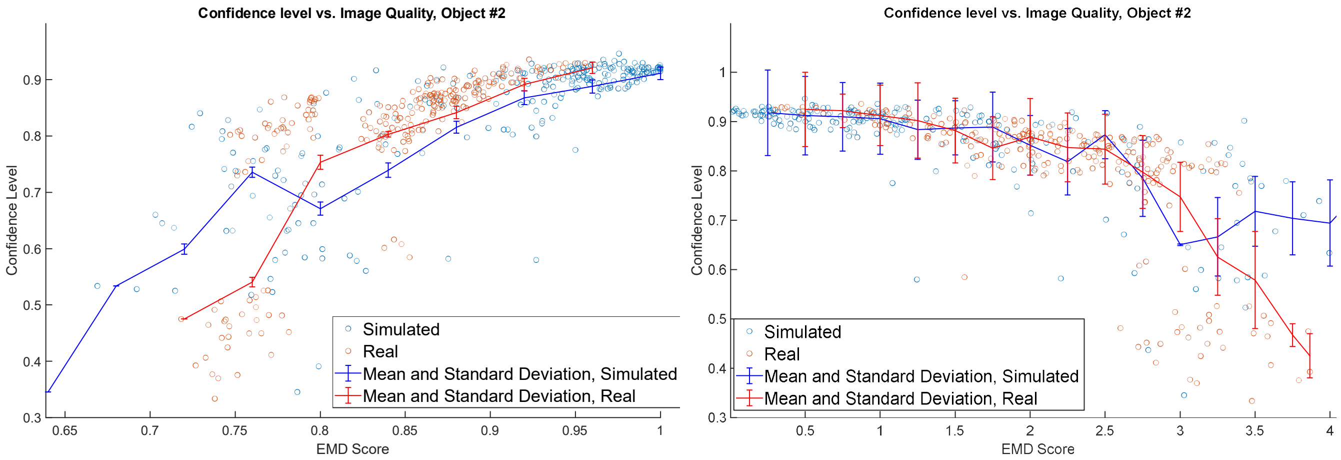
FIGURE 8. Using EMD (left) and SSIM (right) as similarity measures of real rained image and clear image with simulated rain added shows a clear trend towards improving similarity, with the addition of simulated raindrops. Lower EMD scores and higher SSIM scores both mean increased similarity levels between compared images.



(a) Real rained image

(b) Image with simulated rain

FIGURE 9. Objects are detected in real (top) and simulated rained images (bottom), with different confidence levels (using YOLOv3). Bigger objects are detected with higher confidence levels than smaller ones. The detectors order the detected objects according to their detection confidence levels.

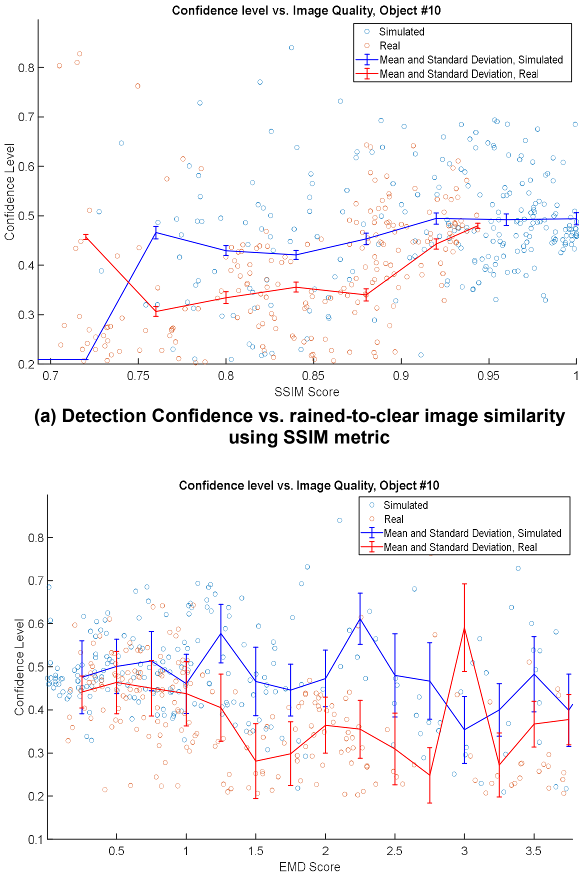


(a) Detection Confidence vs. rained-to-clear image similarity using SSIM metric

(b) Detection Confidence vs. rained-to-clear image similarity using EMD metric

FIGURE 10. Detection Confidence level of Object #2 increases with decreased image degradation in both real and simulated rain images. The mean of sample detection confidence levels (center of error bars) has a strong correlation to image quality.

Figure 11 shows a plot of another object (Object #10) detection confidence levels versus image degradation levels. The trend is still visible on both real and simulated rain images but not as strong as the first object. We calculated the correlation between detection confidence and image degradation for several objects in the real and simulated datasets. The results are shown in Table II. As expected, object 2 showed a strong correlation between its detection score and image quality. The correlation scores for real and simulated rained images for object 2 were also very comparable. Object 10, on the other hand, showed a weaker correlation score, which explains why the trend was observed in Figure 11. The table also shows that object 1 and object 15 show no clear correlation between detection confidence and image degradation level. Further analysis showed that object 1 was the largest object (car) in the image scenes, and its detection confidence remained high under all levels of image degradation. Object 15 was the opposite. It was very small and its detection confidence was low at all levels of image degradation. In both cases, detection confidence levels were not strongly correlated to image degradation, caused mainly by adherent raindrops.



(a) Detection Confidence vs. rained-to-clear image similarity using SSIM metric

(b) Detection Confidence vs. rained-to-clear image similarity using EMD metric

FIGURE 11. For small objects in the image (e.g., Object #10), the detection confidence level is low, even at low image degradation levels. The correlation between detection confidence and image quality is also weaker than larger and brighter objects in the same image (e.g., Object #2).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3103895, IEEE Access

TABLE II
CORRELATION IS CALCULATED BETWEEN DETECTION CONFIDENCE AND IMAGE QUALITY FOR REAL AND SIMULATED RAINED IMAGES. COMPARABLE CORRELATION SCORES FOR REAL AND SIMULATED RAINED IMAGE OBJECTS. SOME OBJECTS SHOW WEAK TO NO CORRELATIONS.

	Correlation between Confidence and EMD		Correlation between Confidence and SSIM	
Object ID	Real	Simulated	Real	Simulated
1	-0.3007	0.3422	0.2536	-0.4207
2	-0.7440	-0.6641	0.7690	0.7390
3	-0.6552	-0.2923	0.6453	0.6167
4	-0.7019	-0.8262	0.6827	0.8427
5	-0.3589	-0.7343	0.4529	0.7421
6	-0.4145	-0.5397	0.4538	0.5034
10	-0.2617	-0.3070	0.2141	0.3390
15	-0.3912	0.0365	0.2431	-0.3717

Figure 12 shows histograms of correlations between detection confidence and image quality, for fifteen objects in the scenes of the images, calculated for both real and simulated rained image sets.

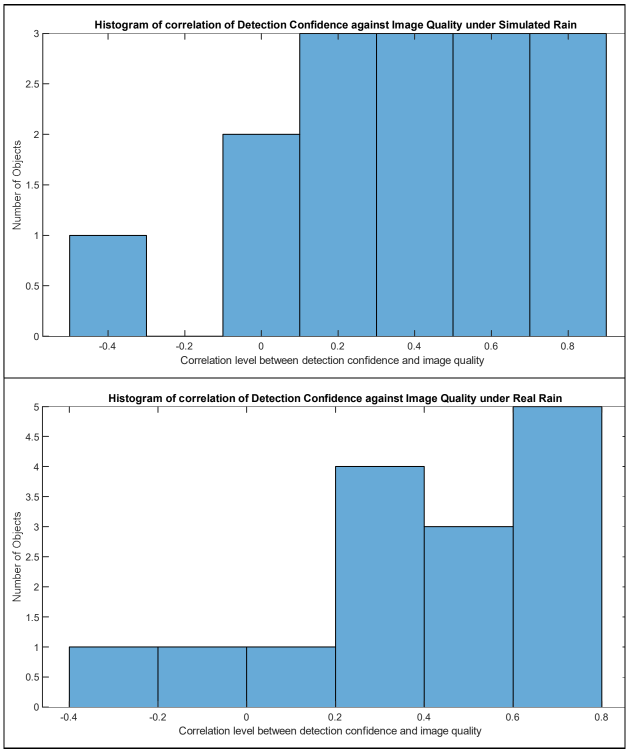


FIGURE 12. Histograms of correlation of detection confidence and image quality for both real (top) and simulated (bottom) rained images show the strongest correlation levels under both real and simulated rain. Only a few objects had weak correlation, and around half the objects showed relatively strong correlation levels (above 0.5).

B. PRECISION AND RECALL METRICS VERSUS IMAGE DEGRADATION LEVEL

The other means of assessing the performance of our raindrop simulator is using precision and recall metrics, instead of just confidence levels, against image degradation levels. Initially, we ran detection algorithms on clear image sets and used them as the ground truth for our precision and recall calculations. A detection is considered true positive (*TP*) if the detected object in the rained image (real or simulated) matched that found in the clear image. A false negative (*FN*) is considered when an object in the clear image is not detected in the rained one. A false positive (*FP*) is when the classifications of the objects detected in clear and rained images do not match (e.g., car vs. boat). Figure 13 shows the plot of recall against image degradation, represented with EMD measure. As can be seen from figure 13, there is a clear trend of decreased recall scores with the increase of image degradation, represented by the EMD similarity metric. The trend is observed in both real and simulated rained image sets.

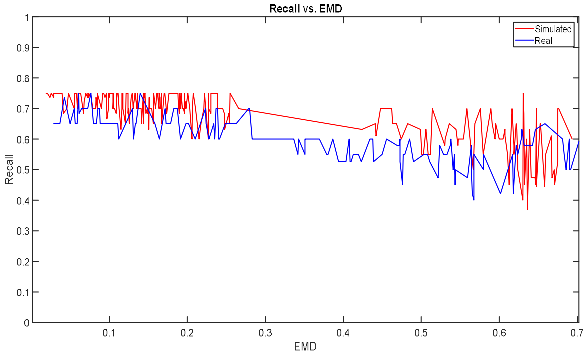


FIGURE 13. Calculating the recall score of detected objects over all captured frames of rained images, with different rain intensities, shows a trend of decreased recall score with increased image quality, represented by the EMD similarity score. As the degradation in image quality increases, objects are detected less often, and recall score correlation to image quality becomes weaker.

Table III shows the correlation value of recall score versus image quality (EMD and SSIM) for both real and simulated rained sets.

TABLE III
CORRELATION OF RECALL SCORE VERSUS IMAGE QUALITY IS STRONG IN BOTH REAL AND SIMULATED RAINED DATASETS. THE CORRELATION VALUES ARE ALSO COMPARABLE FOR REAL AND SIMULATED RAIN

Recall vs. EMD		Recall vs. SSIM	
Real	Simulated	Real	Simulated
-0.7645	-0.8097	0.7253	0.8356

Precision scores calculated on the same datasets did not show a clear dependency on the degradation levels of rained images. Examining the detection results, we found that the dominant failure mode was false negative rather than false positive. This can be explained as follows. The object detectors were trained with full or partial images of common objects that can be found on the street. A raindrop may occlude sections of an object, but the remaining un-occluded section may still be sufficient features to correctly classify the object. Only when the occlusion is significantly large enough, that the detector fails to detect (false negative) the raindrop-occluded object. It is much less likely that the occlusion would leave sections of the object, which would cause the detector to classify it incorrectly (false positive).

Since precision is calculated as $\frac{TP}{TP+FP}$, it is clear why the precision score came as one for most of the samples, and thus was weakly correlated to the degradation level. The recall, however, is calculated as $\frac{TP}{TP+FN}$ so it was more correlated to the degradation level and showed a significant decrease with the increase of the degradation level.

C. Comparative Analysis

To evaluate the performance of our raindrop simulator against the state-of-the-art ray-tracing-based raindrop simulators, we used Carlin's [9] model to generate rainy images from clear 500x500 pixel images, that we selected from our original dataset. A total of 128 images with different rain patterns were used. Figure 14 shows an example of an image with generated raindrops using our model and Carlin's model. Carlin's model generates raindrops with similar shape and orientation, compared to raindrops generated from our model that vary in size, shape, and orientation. Roser et al. [22] modeled raindrops using Bezier curves and showed that the area of a raindrop as seen on a windshield is proportional to its volume and maximum thickness. For real raindrops, the bigger the raindrop volume is, the less transparent the raindrop becomes. Raindrop transparency level in Carlin's model is also higher than that generated by our model, and higher than what is normal for the size of raindrops generated by his model.

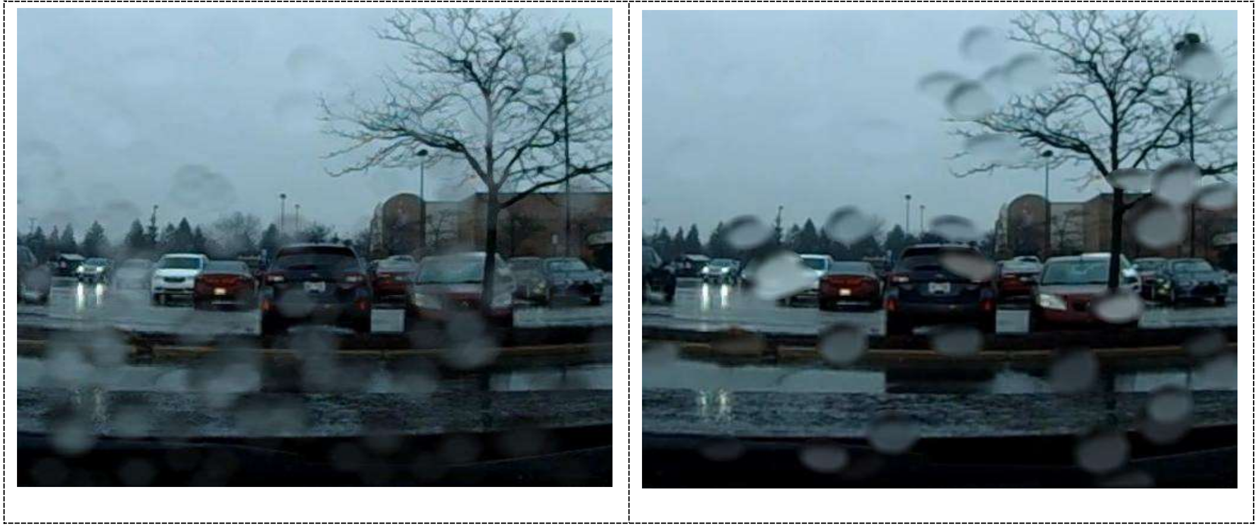


Figure 14: Images with raindrops that were generated by the ray-tracing method (left) and our method (right). Our model generates raindrops with more varieties in size, shape, and orientation compared to the ray-tracing model. The transparency levels in our generated raindrops are closer to that of real drops and are generally lower than that of raindrops generated by the ray-tracing model.

For quantitative comparison, we evaluated the performance of CNN-based object detectors, using rained images generated by both Carlin's model and ours. We used two metrics in our evaluation, detection confidence level, and detection recall score. For the confidence level evaluation, we matched objects in the rained images that were generated by both models to those detected in real rained images. We then calculated the differences in detection confidence for each object detected in real and simulated rain images. Table IV shows a summary of some statistical metrics for the object detection confidence level, using real and simulated rain datasets The two models seem to produce similar results in terms of object detection confidence levels, as indicated by the mean and standard deviation metrics of the results.

TABLE IV
MEAN AND STANDARD DEVIATION OF OBJECT DETECTION CONFIDENCE LEVELS SHOW STATISTICAL SIMILARITY OF RESULTS UNDER REAL AND SIMULATED RAIN DATASETS.

Statistical metrics of detection confidence results	Mean	Standard Deviation
Datasets		
Real rain dataset	0.8029	0.1702
Our generated rain dataset	0.8013	0.1852
Ray-tracing generated dataset	0.8108	0.1857

For Recall score metric, we matched the image objects detected in simulated rained images from the two models, to the ones detected in the clear image (reference) dataset. Recall score is calculated for each image frame and the results are compared to the recall score of detection with real rained images. Table V shows a summary of some statistical metrics for the object detection recall score, using real and simulated rain datasets. The object detection recall scores are closer for our model to those with real raindrops than the scores calculated for the ray-tracing model.

TABLE V
MEAN AND STANDARD DEVIATION OF OBJECT DETECTION RECALL SCORES SHOW STATISTICAL SIMILARITY OF RESULTS UNDER REAL AND SIMULATED RAIN DATASETS.

Statistical metrics of detection Recall results	Mean	Standard Deviation
Datasets		
Real rain dataset	0.6484	0.1956
Our generated rain dataset	0.7601	0.1742
Ray-tracing generated dataset	0.8132	0.0864

VI. DISCUSSION

Our proposed simulator generated a visually convincing adherent raindrop on a vehicle windshield. The model performs best when generating simple raindrops that can be approximated with an ellipse. For more complex raindrop shapes, the model can be programmed to generate several intersecting elliptical raindrops, each approximating one section of the complex raindrop shape. This technique was tested by trying to mimic real raindrops of complex shapes using our simulator. Results showed great improvement of raindrops similarity, compared to using a single ellipse representation of complex shapes. The object detection tests we conducted using three CNN-based deep learning object detectors showed similar behavior using real or simulated rained datasets. This "behavior" can be described as follows:

1) The correlation values between recall score and image quality were very close on all datasets tested and using both YOLOv3 and Faster-RCNN detectors
2) The correlation values between detection confidence levels and image quality were also close on all datasets and the same detectors.
3) Big objects showed Resilience to raindrop-induced image degradation, and that behavior was similar in both real and simulated rained image datasets. Smaller objects in the image were more susceptible to the presence of raindrops and this susceptibility was similarly observed in both real and simulated rained datasets.

Our implementation of the SSD detector did not perform as well as YOLOv3 or Faster-RCNN. It detected much fewer objects and the detection confidence dropped fast with increased raindrop content. Because of that, SSD results were not included in our raindrop simulator performance evaluation.

EMD and SSIM were good metrics for evaluating degradation in image quality at different levels of raindrop content in an image. They, however, are not perfect. Special attention needed to be applied to limit the influence of dynamic background objects, whether being a distant vehicle, moving clouds, or even flickering street lights. We also observed that they do not always agree when representing image similarities, in a sense that increased SSIM score does not always mean a decrease in EMD score, for the same sets of images compared. This meant that these two metrics cannot be used interchangeably for individual image matching. For observing trends that extend over many samples, the metrics show similar behavior and they appropriately track the progression of image degradation, caused by increased raindrop presence.

Comparison of rained images generated by the state-of-the-art ray-tracing-based model showed very close results, both in visual perception or the generated raindrops, and the usability of generated rained images in object detection system validation.

In terms of performance speed, we developed our raindrop simulator using MATLAB 2018b scripting language, with no specific optimizations. We ran it on a PC with an AMD FX-8350 microprocessor, 16 GB of DDR3 RAM, a 500 GB SSD hard drive, and running Windows 10 operating system. It took on average 600 ms to generate each raindrop, using the full (1280 x 650) image as an input. Figure 15 shows samples of our generated raindrop images, alongside the original, clear images, and real rained images with roughly the same level of rain-caused degradation, as our generated ones.

VII. CONCLUSION

We developed a dynamic adherent raindrops simulator that adds generated raindrops to clear images.

The simulator accepts any set of clear images at any size and resolution, and generates simulated rained image sets, with generated raindrops added. The simulator allows for full control of the generated raindrops, in terms of intensity, size, and orientation ranges. The quality of generated raindrops was visually convincing. Using CNN object detectors as a case study, we showed that generated rained images can be successfully used to evaluate real vision-based applications in the automotive domain, such as the detection of objects in the path of the moving vehicle. Our generated rained images showed a strong resemblance to images generated by the state-of-the-art ray-tracing approach, although our approach to raindrop simulation is different. We expect the generated rained images from our model to be useful in different vision-based applications that require lots of data samples for training, testing, and optimizing.

We also published a dataset of images of urban drive scenes, with added simulated raindrops, generated through our raindrop simulator. The images in the dataset were augmented with text files, that identify the size, location, and orientation of each generated raindrop.

For future studies, it is of interest to investigate optical flow calculations that may improve simulated raindrop representation on dynamic scene images, at the expense of added complexity.

References

[1] Y. Hamzeh, Z. El-Shair and S. A. Rawashdeh, "Effect of Adherent Rain on Vision-Based Object Detection Algorithms," SAE International Journal of Advances and Current Practices in Mobility, vol. 2, pp. 3051-3059, 2020.

[2] J. C. Halimeh and M. Roser, "Raindrop Detection on Car Windshields Using Geometric-Photometric Environment Construction and Intensity-Based Correlation," in 2009 IEEE Intelligent Vehicles Symposium (IV), Xi'an, Shaanxi, China, 2009.

[3] MIT OpenCourseWare, "Refraction and Snell's Law," 2011. [Online]. Available: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-007-electromagnetic-energy-from-motors-to-lasers-spring-2011/lecture-notes/MIT6_007S11_lec32.pdf. [Accessed 28 3 2021].

[4] G. Woan, "The Cambridge handbook of physics formulas," 2000. [Online]. Available: https://archive.org/details/cambridgehandboo0000woan/page/154/mode/2up?q=fresnel. [Accessed 28 3 2021].

[5] S. Alletto, C. Carlin, L. Rigazio, Y. Ishii and Tsukizawa, "Adherent Raindrop Removal with Self-Supervised Attention Maps and Spatio-Temporal Generative Adversarial Networks," in IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 2019.

[6] A. Palazzi, D. Abati, S. Calderara, F. Solera and R. Cucchiara, "Predicting the Driver's Focus of Attention: The DR(eye)VE Project," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1720-1733, 2019.

[7] Z. Hao, S. You, Y. Li, K. Li and F. Lu, "Learning From Synthetic Photorealistic Raindrop for Single Image Raindrop Removal," in IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.

[9] I. Stuppacher and P. Supan, "Rendering of Water Drops in Real-Time," in Central European Seminar on Computer Graphics for Students, 2007.

[10] Y. Hamzeh and S. Rawashdeh, "Framework for simulating and removing rain in stereo-image videos," in IEEE INTERNATIONAL CONFERENCE ON, Oakland, Michigan, 2018.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 13, no. 4, pp. 600 - 612, 4 APRIL 2004.

[12] Wikipedia, "Earth mover's distance," 23 7 2020. [Online]. Available: https://en.wikipedia.org/wiki/Earth_mover's_distance. [Accessed 16 11 2020].

[13] U. Yilmaz, "The Earth Mover's Distance," 12 2 2009. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/22962-the-earth-mover-s-distance. [Accessed 2 11 2020].

[14] Y. RUBNER, C. TOMASI and L. J. GUIBAS, "The Earth Mover's Distance as a Metric for Image Retrieval," International Journal of Computer Vision, vol. 40, no. 2, pp. 99-121, 2000.

[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in European conference on computer vision, Springer, Cham, , 2016.

[17] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015.

[18] MathWorks, "Create a Gallery of Transformed Images," MathWorks Inc., [Online]. Available: https://www.mathworks.com/help/images/creating-a-gallery-of-transformed-images.html. [Accessed 25 3 2021].

[19] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa and K. Ikeuchi, "Adherent Raindrop Modeling, Detection and Removal in Video," IEEE TRANSACTION ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE, vol. 38, no. 9, pp. 1721-1733, 2016.

[20] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3103895, IEEE Access

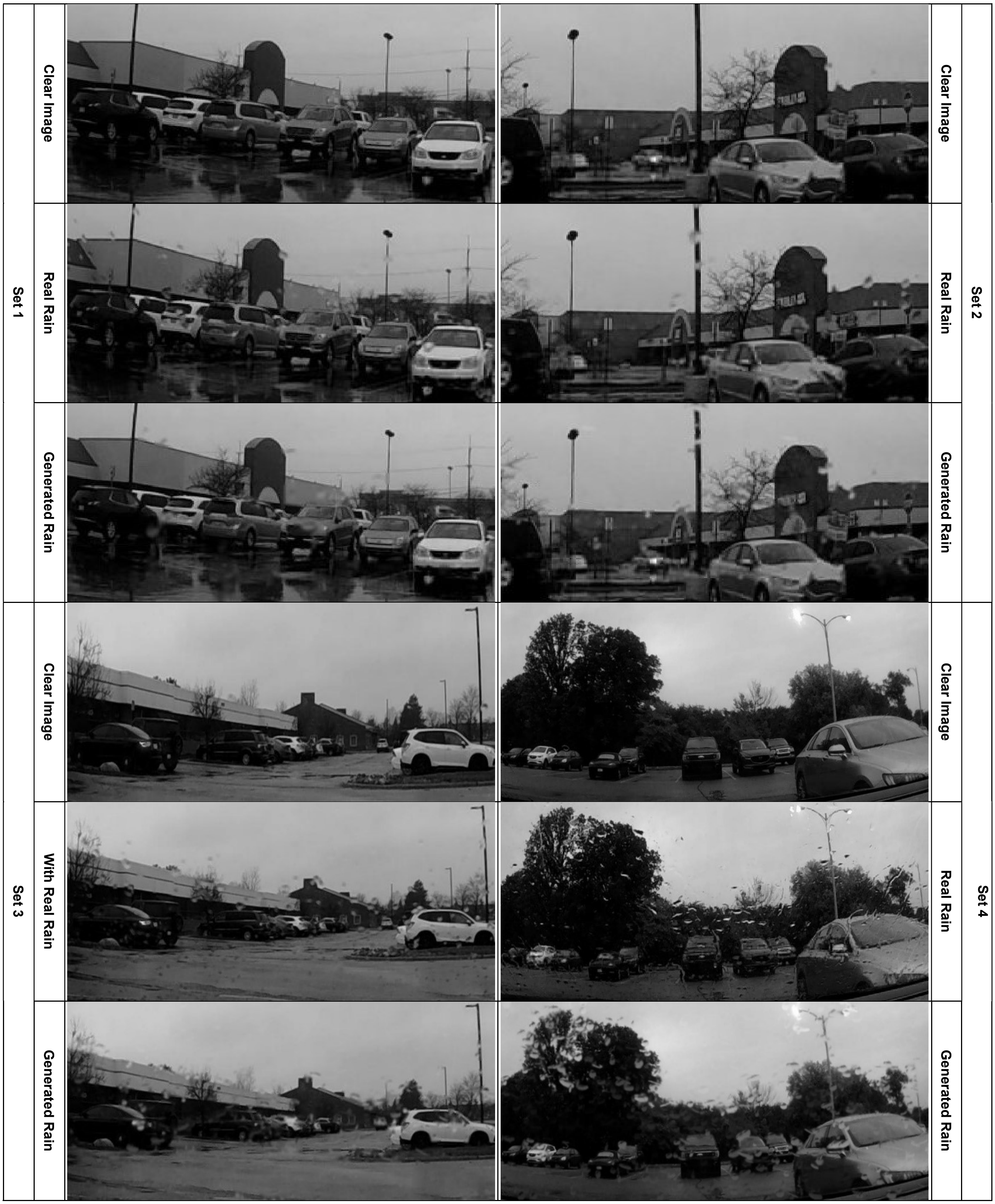


FIGURE 15. Examples of Clear, Real, and randomly generated raindrop images from our dataset. rain intensity ranges from light (set 1) to relatively heavy (set 4). The generated raindrops are perceptually convincing to a human observer.

YAZAN HAMZEH received a B.Sc. degree in electrical and electronics engineering from the United Arab Emirates University, Al-Ain, the United Arab Emirates in 1995. He received an MSc degree in computers and control systems from Wayne State University, Detroit, Michigan in 1998. He is currently a Ph.D. candidate in Electrical, Electronics, and Computer Engineering, at the University of Michigan-Dearborn, working within the domain of Computer Vision. He has been working for the last seven years at Ford Motor Company, Dearborn, Michigan as a systems engineer. His research interest includes vision-based applications in the automotive domain.

ZAID A. EL-SHAIR received the B.Sc. degree in computer engineering from Princess Sumaya University for Technology, Amman, Jordan, in 2018 and the M.S.E. degree in Computer Engineering from the University of Michigan-Dearborn, MI, in 2019. He is currently a Ph.D. candidate in Electrical, Electronics, and Computer Engineering, at the University of Michigan-Dearborn, working within the domain of Computer Vision. Recently, he has been researching and developing methods for robust high-speed object tracking using the very recent, bio-inspired, Event Cameras. Research experience includes Networked-Embedded Systems, Machine Learning, Intelligent systems, and Computer Vision.

ABDALLAH CHEHADE received the B.E., Mechanical Engineering degree from the American University of Beirut, Beirut, Lebanon, M.S. M.S., Mechanical Engineering and M.S., Industrial and Systems Engineering from the University of Wisconsin Madison. He received a Ph.D. degree in Industrial and Systems Engineering and Ph.D. Minor, Computer Sciences and Statistics from the University of Wisconsin Madison. His research activities and interests include system, Informatics & Data Analytics, Reliability Analysis, Prognostics & Degradation Modeling, Stochastic Processes & Decision-making, Statistical Modeling & Bayesian Inference, Data Fusion, and Internet-of-things.

SAMIR A. RAWASHDEH (M'05, SM'20) received the B.S. degree in electrical engineering from the University of Jordan, Amman, Jordan in 2007, and the M.S.E.E. and Ph.D. degrees in 2009 and 2013 in electrical engineering from the University of Kentucky, Lexington, KY. He is currently an Associate Professor in the Electrical and Computer Engineering department at the University of Michigan – Dearborn, which he joined in 2014. His research activities and interests include robot perception, embedded systems, and smart health