Photo-Realistic Simulation of Road Scene for Data-Driven Methods in Bad Weather

Kunming Li[1,2] Yu Li[3] Shaodi You[1,2] Nick Barnes[1,2]

[1]Data61, CSIRO, Canberra, Australia [2]Australian National University, Canberra, Australia
[3]Advanced Digital Sciences Center, Singapore

Abstract

Modern data-driven computer vision algorithms require a large volume, varied data for validation or evaluation. We utilize computer graphics techniques to generate a large volume foggy image dataset of road scenes with different levels of fog. We compare with other popular synthesized datasets, including data collected both from the virtual world and the real world. In addition, we benchmark recent popular dehazing methods and evaluate their performance on different datasets, which provides us an objectively comparison of their limitations and strengths. To our knowledge, this is the first foggy and hazy dataset with large volume data which can be helpful for computer vision research in the autonomous driving.

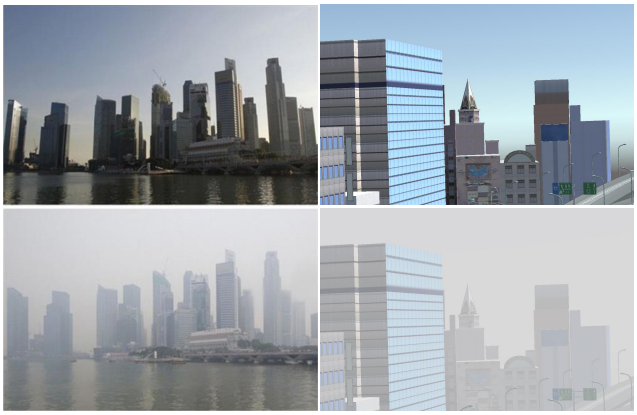

Figure 1. The same scenes in good weather and haze. (Left) real images from [40]; (right) simulated images from our dataset.

1. Introduction

We have witnessed the fast development in autonomous vehicle systems recently [38]. In most autonomous vehicle systems, the visual sensors (*i.e*. cameras) are the key component that is used to sense the circumstance nearby. This visual information is further used to understand the driving conditions and events. To understand the scene captured by the cameras, machine learning approaches are always used to train the system to have the ability for some computer vision tasks like scene parsing [13], object recognition [14] [2], object detection [25] [39] [30], tracking [41], *etc*. Usually, a dataset with large size and large variety is required to train a robust system for the computer vision tasks [28]. For this reason, some datasets are provided in the research community such as **KITTI** [9], **Cityscapes** [5] which are specific for road scenes in city. However, most of the existing datasets, either from real scenes or from rendered scenes, are captured under good weather condition with clear scenes. As pointed by Taral *et al.* [34], one of the causes of vehicle accidents is the reduced visibility caused by bad weather such as foggy and hazy weather. Fog and haze are common bad weather. They are caused by floating particles in the atmosphere which absorb and diffuse the light transmission and subsequently cause the foggy/hazy effect [19]. In the foggy or hazy weather, the visibly is greatly reduced (see Fig. 1 for example) and may impair the performance of the computer vision systems.

The algorithms which are able to recover the visual visibly is called ”dehazing” or ”defogging” algorithms (we will use the term dehazing in the rest of the paper). In the past two decades, there has been a significant progress in improving visibility of foggy or hazy images [19]. Although a large number of dehazing works have been proposed, a quantitative evaluation of the dehazing methods on large dataset of road scene is never done. The challenges are from the fact it is impossible to capture clear / hazy image pair when other conditions (*e.g*. lighting) are exactly the same. Some works render hazy effect using the depth map [6]. However, getting the depth information is not easy. Therefore, [6] only provided 11 images for evaluation. Another

approach is to use computer graphics technique to render the scene, an example is **FRIDA**. However, their rendering is too far from the real scene, making it less convincing for evaluation. For this reason, we generate this dataset. Compared with capturing real atmospheric scenes, using synthetic data costs lower and ensures greater flexibility and variety. We use the physics based rendering technique to ensure the closeness with the real scene. In the following of the paper, we will describe the dataset, and our experiments on the dataset. Our contribution can be summarized as :

Our *first* contribution is to generate large volume, photo-realistic and varied dataset for data-driven computer vision research and autonomous driving systems under bad weather. In the first version of our dataset, it contains approximately 2000 frames, which based on three Japanese cities models. Our dataset provides foggy and hazy images, depth map and masks.

Our *second* contribution is to compare popular foggy and hazy images datasets, which include scenes captured from the virtual world as well as the real world. Fattal [6] used the camera image and the depth map to synthesize the dataset, but it only contains 11 images. **FRIDA** [35] utilized computer graphics techniques to synthesize road images from virtual world. However, the dataset has small volume data but also does not provide non-sky mask to reduce error of evaluation.

Our *third* contribution is a quantitative benchmark of a number of popular dehazing methods. Our experiment shows that recent work [3] and [26] generally perform better than previous methods.

The article is organized as follow. Section 2 reviews related works on using synthetic data for data-driven computer vision research and autonomous driving systems under bad weather. Section 3 describes the methods to build photo-realistic simulation, introducing approaches to constructing virtual world and dynamics of rendering images. In Section 4, we report our experiments on comparison for popular foggy and hazy dataset. Moreover, we benchmark recent popular single image dehazing methods. We conclude in Section 5.

2. Related Work

Several works have investigated the use of synthetic data to solve data-driven computer vision problems and the research of automatic driving systems. For example, Satkin *et al.* [29] utilized the rendered 3D model to create a rich understanding of the scene. Pepik *et al.* [24] also utilized the 3D synthetic data to tackle computer vision problems such as object detection. Initially, 3D simulation has been used in the research of computer vision to model object like human shape [10]. However, as Vaudrey *et al.* [37] suggested, the ground truth which was produced by synthetic data is easy to estimate and these synthetic scenes lack photorealism, which creates the difference between the virtual world and the real world. However, as Gaidon *et al.* [7] introduced, pre-training computer vision algorithms on virtual data improve the performance of the algorithm. To our knowledge, however, the popular foggy and hazy datasets do not contain enough images for training and validation. The strengths and drawbacks of popular foggy and hazy datasets for the autonomous driving system are discussed in the following.

The progress of computer graphics and advanced generics platforms allows wider use of synthetic data. Stark *et al.* [31] indicated that the multi-view detector model can be built only through the 3D source. Marin *et al.* [20] suggested that the appearance detector model trained by the virtual world can be used in the real world. Hattori *et al.* [11] proposed a related approach to learn a pedestrian detector model without using data from the real world. Taylor *et al.* [36] provided publicly available visual surveillance simulation test bed for system design and evaluation, which is based on commercial game engines.

Only a few works focus on training and evaluating autonomous driving systems and data-driven computer vision research under bad weather. Chen *et al.* [4] evaluated their direct perception model on **TORCS**, it allows users to control the object in the simulator through user interface directly. **TORCS** is a highly portable multi-platform car racing simulation. **TORCS** provides different kinds of cars, tracks, and opponents. The roads and objects in the simulator are all related to the racing car. But as the dataset for training and testing autonomous driving systems, it lacks diverse context and does not contain enough labeled ground truth.

Virtual KITTI [7] provides efficient real-to-virtual world cloning methods with labeled with accurate ground truth for tackling computer vision problems such as object detection and depth. The dataset is photo-realistic synthetic video dataset, aiming to learn and evaluate autonomous driving systems. The Virtual KITTI provides 50 monocular videos, which are based on urban traffic road. However, **Virtual KITTI** provides few images about driving environment under bad weather. In addition, the dataset does not provide non-sky masks. When evaluating algorithms, non-sky masks are usually used to reduce the estimation error.

SYNTHIA [27] aims to tackle semantic segmentation and the related scene understanding problem for automatic driving. It has a large volume of data, which includes European-style town, modern city, highway and green areas. The dataset also contains various dynamic objects and multiple seasons. However, the dataset does not provide non-sky masks and the various level of haze and fog image for evaluating dehazing algorithm.

Fattal's dataset [6] provides ground truth, corresponding images and non-sky masks, of which are captured in the real

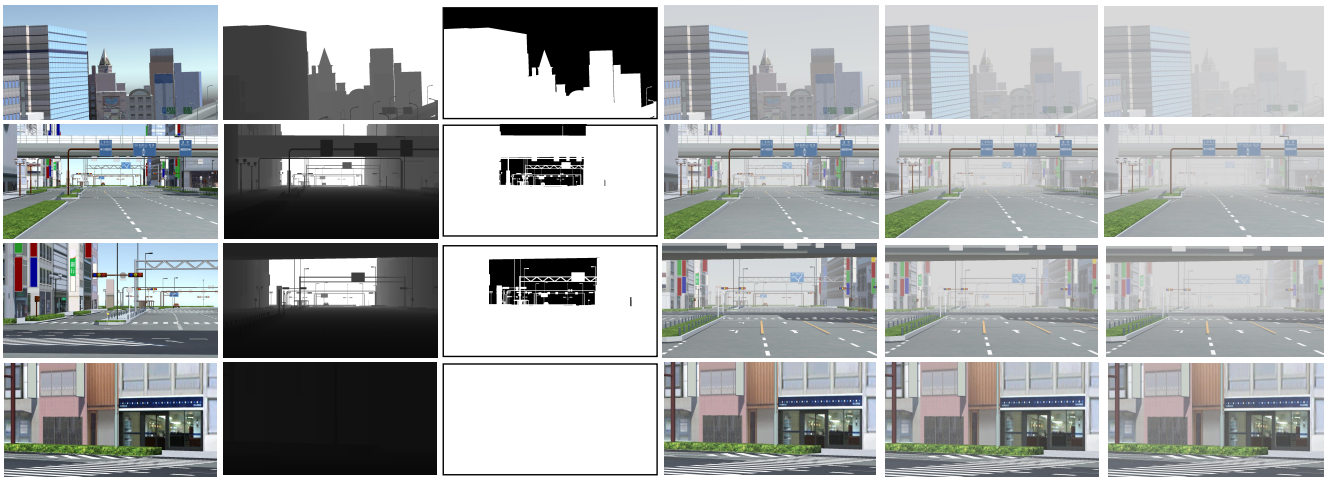

Figure 2. Samples from our dataset.

world. The dataset is mainly used in evaluating dehazing algorithm. However, the dataset only contains 11 synthesized images, which is not large enough for evaluating dehazing algorithms and autonomous driving systems.

FRIDA [35] provides numerical synthetic images, which are used in evaluating the performance of the automatic driving system in a systematic way. The dataset could contribute to the improvement of vision enhancement in the foggy environment. **FRIDA** dataset provides different foggy and hazy images such as uniform fog, heterogeneous fog, cloudy fog, and cloudy heterogeneous fog. But the settings and context are not diverse enough, which only took from similar roads in a city. Also, the dataset only has 420 images.

In this paper, we propose an method to tackle the issues mentioned above, especially for data-driven computer vision research and autonomous driving system under bad weather. The current methods have two mainly limitations: (1) It is costly and time-consuming to produce large volume data. (2) Many datasets do not provide non-sky masks. Because of these limitations, only a few previous works have achieved the full potential of synthesized data.

3. Generating Virtual Driving Environment

Our approach to achieving virtual photo-realistic simulation mainly consists three steps, which are shown as following three sections: (1) Investigation of real-world data, which includes real-world traffic situation, the size of vehicles and driving environment. (2) Generating virtual driving environment based on the investigation. (3) Rendering and collecting the images, which includes rendering depth image and utilizing atmospheric scattering model to form foggy and hazy images.

3.1. Investigating real-world data

First of all, we investigated real-world data about traffics and driving environment, which includes road types, public facilities and traffic rules, the proportion of the size of vehicles, pedestrian and building.

In order to promote the diversity and reality of our dataset, we investigated different traffic situations and public transport facilities, which includes city, residential, various traffic roads, traffic poles, buildings, sky and other objects could be seen in the real world.

3.2. Generating synthetic scenarios

Unity3D [1] is decided to build the simulator. The Unity3D has the diverse and open-source packages for constructing virtual environments. The images can be rendered through programming and computer graphics techniques. Also, compared with the dataset captured from the real-world, the data collected from the virtual environment can be more diverse. It would be less time-consumed and less costly for building simulator and collecting images. Therefore, the simulator is decided to be constructed by Unity3D for the purpose of diverse scenes and budget.

The simulator is based on Japanese cities model from 3D Urban Datamodel [2]. To simulate the real-world situation, the various traffic environments are considered and created. The raw data are taken from various scenarios. Pedestrian are from Unity Asset Store[3].

The system consists of light, camera, vehicles, sky, roads, poles, signs, buildings and pedestrians. The func-

[1] http://www.unity3D.com

[2] http://www.zenrin.co.jp/product/service/3d/asset/

[3] https://www.assetstore.unity3d.com/

tions of each component are realized through programming. The programmed scripts then are attached to the corresponding objects. The light in the simulator creates shadows. The light also adjusts the color of texture in the simulator. The pedestrians have animations, which are the movements of pedestrians in Unity3D [1]. The states of animation of pedestrians include walk, stand and jump. The simulator also contains a number of buildings, poles, signs and other traffic facilities, promoting diversity and reality of the scenes.

Light is one of the essential parts in the simulator, which allows the simulator to be more realistic by defining colors and moods of the 3D environment. It also creates shadows for the objects in the scene. In this simulator, the lights are set as directional light, which illuminates the whole simulator. The directional light covers a large portion of the scene and makes the shadows.

3.3. Rendering Image

To make the images in dataset more realistic, rendering techniques are applied. In Unity3D[1], Shader is the script of Shaderlab, which is a programming language. Shader is used to make shading and produce special effects or do video post-processing. They are small scripts which include the mathematical calculations and algorithms for calculating the color for the single pixel of the object in the scenes. For the simulator, Shader program is used to produce the depth camera. In unity3D [1], the material of an object means the way of its rendering. It includes the texture and tiling information, color tints and more.

3.3.1 Depth Image rendering

The depth camera reflects the distance between objects and camera. The depth camera displays the depth value at each screen coordinate, which means the camera represents the distance between objects and camera through the depth of color. As the object stays further away from the camera, its color becomes lighter. If the object is closer to the camera, its color becomes darker. A shader is created to process the depth texture and display. It mainly consists vertex and fragment shader. The Vertex Shader is the program which runs on each vertex of the object and is used for rasterizing the object on the screen. In the design of the depth camera, the main function of the vertex shader is to sample the texture in the fragment shader. The fragment shader is the program run on each pixel on the screen, calculating the value of the color on each pixel. In the simulator, the depth value is calculated by function *Linear01Depth*, which returns the linear depth to the screen. A material is set to contain the shader. Then, the destination RenderTexture is passed to the material and the material is attached to the camera.

3.3.2 Foggy/hazy image formation

Fog and haze are common phenomena in the real world, which usually are caused by atmospheric particles. The noticeable degradation of foggy and hazy scenes has the effect on object detection. According to the survey provided by [19]. we can apply optical models to build the foggy/hazy scenes and the formation of a hazy image usually is as follow:

$$I(x) = e^{-\beta d(x)} R(x) + L_{\inf}(1 - e^{-\beta d(x)}), \qquad (1)$$

where I is the observed color, R is the scene radiance (clear scene), $L_{\inf}$ is the color of the environmental light, and d is the distance from the scene objects to the camera (*i.e.* depth). In this haze image formation, the observed color is a summation of two components: the direct attenuation $e^{-\beta d(x)} R(x)$ and the airlight $L_{\inf}(1 - e^{-\beta d(x)})$. The direct attenuation describes the decayed scene radiance by scattering and absorb effects of the floating particles in haze. The second component airlight [21] defines the type of scattered environmental light captured in the observers cone of vision. The airlight causes the washout effect in hazy images. The exponential expression is according to Koschmieder's law. It indicates the the scene radiance decays and haze effect increase in a exponential relationship with scene depth d and also determined by the scattering coefficient β. Different scattering coefficient β will result in different levels of haze effect (as we will use in our simulation later). The attenuation factor $e^{-\beta d(x)}$ is often referred to as transmission factor and denoted together as t.

4. Experiments

To evaluate enhancement algorithms, in this section, visibility enhancement methods are benchmarked. Through benchmarking various methods and datasets, we can know the detail of comparison. In addition, popular hazy and foggy datasets are discussed in this section.

We apply recent dehazing methods **Tarel 09** [34], **Ancuti 13** [1], **Tan 08** [32], **He 09** [12], **Meng 13** [22], **Berman 16** [3], **Tang 14** [33] and **Ren 16** [26] on different datasets with ground truth. According to [23], the hazy images should be created from real atmospheric scenes where all possible conditions ranging from light mist to various dense fogy under the different background should be considered. Ideally, the images need be captured under the environment where cloud, sunlight distribution and illumination are fixed. Ideally, It is possible. However, in the real world, it is rarely meet these conditions. In addition, it is challenging in practice to capture the image of distant objects without the effect of particles. Therefore, we use synthesized image to evaluate dehazing methods.

We firstly apply methods on the datasets provided by [6] and [35]. Then, we perform the methods on our dataset.

Table 1. The mean absolute difference of final dehazing results on Fattal's dataset [6]. The three smallest values are highlighted.

Methods	Road 1	Moebius	Reindeer	Road 2	Flower 1	Flower 2	Lawn 1	Lawn 2	Mansion	Church	Couch
Tarel 09	0.142	0.132	0.122	0.167	0.121	0.160	0.145	0.128	0.122	0.125	0.115
Ancuti 13	0.131	0.171	0.113	0.218	0.161	0.141	**0.308**	0.234	0.140	0.143	0.128
Tan 08	0.126	0.105	0.132	0.236	0.174	0.104	**0.300**	0.272	0.153	0.140	0.148
He 09	**0.045**	**0.026**	**0.038**	0.121	**0.061**	**0.046**	0.078	0.080	**0.051**	**0.051**	**0.034**
Meng 13	**0.046**	**0.038**	**0.060**	0.096	**0.065**	**0.048**	0.114	0.106	**0.051**	**0.049**	**0.048**
Berman 16	**0.034**	**0.033**	**0.042**	**0.067**	0.085	**0.049**	0.044	**0.047**	**0.038**	**0.041**	**0.040**
Tang 14	0.107	0.104	0.081	**0.043**	**0.068**	0.130	0.043	**0.026**	0.090	0.094	0.089
Ren 16	0.112	0.090	0.083	0.132	0.116	0.094	**0.279**	0.226	0.130	0.143	0.103
Do nothing	0.140	0.137	0.116	**0.085**	0.101	0.168	0.077	**0.058**	0.124	0.127	0.122

Table 2. The mean signed difference of final dehazing results on Fattal's dataset [6]. The three smallest values are highlighted.

Methods	Road 1	Moebius	Reindeer	Road 2	Flower 1	Flower 2	Lawn 1	Lawn 2	Mansion	Church	Couch
Tarel 09	-0.071	**-0.001**	0.069	0.111	**0.022**	-0.131	0.080	0.078	-0.063	-0.060	**0.000**
Ancuti 13	0.009	0.063	0.028	0.177	0.029	**-0.022**	0.282	0.217	**0.025**	**0.019**	0.054
Tan 08	0.056	0.045	**-0.014**	0.1090	0.028	0.056	0.115	0.110	0.085	0.080	0.065
He 09	**-0.002**	**-0.001**	**0.013**	0.070	**0.023**	**0.014**	0.034	**0.041**	**0.022**	**0.019**	0.009
Meng 13	**0.001**	**0.009**	-0.026	**0.049**	**-0.009**	**0.004**	0.047	0.050	**0.020**	**0.014**	**0.005**
Berman 16	**0.004**	-0.016	**-0.015**	**0.007**	-0.077	0.024	**-0.013**	**-0.005**	-0.009	**-0.003**	**0.001**
Tang 14	-0.102	-0.072	-0.068	**-0.030**	-0.060	-0.126	**-0.025**	-0.017	-0.088	-0.090	-0.071
Ren 16	-0.030	0.051	-0.017	0.101	0.050	-0.020	0.253	0.211	0.057	0.074	0.031
Do nothing	-0.130	-0.092	-0.096	-0.057	-0.087	-0.164	**-0.044**	**-0.036**	-0.118	-0.119	-0.096

The results of the dataset and method are compared and discussed. A number of approaches are used in dehazing, *e.g*. contrast enhancement [15]. In this experiment, eight most representative dehazing methods are compared in the experiments. For methods proposed by [34], [12], [3] and [26], we use the codes from authors. The codes of method [1] and [33] were from Li *et al.* [19]. According to Li *et al.* [19], we evaluate results quantitatively through calculating the mean absolute difference (MAD) and the mean signed difference (MSD), through which we can find whether the method is overestimated or underestimated.

According to Li *et al.* [19], three major steps of dehazing are the estimation of the atmospheric light, estimation of transmission and estimation of final enhancement. Different foggy or hazy images have different airlight estimations. In the experiment, the airlight of some images are provided and some methods which can detect airlight automatically,

4.1. Evaluation on Fattal's dataset

Fattal's dataset [6] provides 11 images from different scenes, ranging from road to building. The dataset also provides their corresponding transmission images, which are used for producing synthetic foggy or hazy images. The dataset has already provided foggy and hazy images which are synthesized by the ground truth and the depth image. One example of the synthesized image is shown in Fig. 4. According to Li *et al.* [19], one of the important steps in dehazing is estimating atmospheric light. As the Fattal's dataset [6] provided, the airlight of the dataset is assumed at [0.5, 0.6, 1]. For methods **Tarel 09** [34], **Ancuti 13** [1] and **Ren 16** [26], we did not use airlight provided by the dataset because these methods can detect the airlight automatically. For rest of the methods, airlight [0.5,0.6,1] is applied. The dataset also provides non-sky masks to reduce error through excluding sky regions. The MSD and MAD of the haze-free image and hazy image were also calculated, providing more detail on the evaluation of the methods. The results can be seen from Table 1 and Table 2.

In the experiment, we apply the methods mentioned above to the dataset (excluding sky regions). Average MSD of the eight methods are shown as Fig. 3. From the Fig. 3, it can be seen that **Tarel 09** obtains the least error and performs outstandingly among all the methods in terms of MSD, which is followed by **Berman 16** [3], **Meng 13** [22] and **He 09** [12], with obtaining less than 0.03 MSD. However, in terms of the MAD, **Tarel 09** [34] obtains higher error while **He 09** [12], **Meng 13** [22] and **Berman 16** [3] still rank at top place. The reason is that the value of the difference between the ground truth and the dehazing result could be negative and positive. The accumulative difference of pixels could be very small, which leads to the inaccurate result. The final results on *church* are shown in Fig. 4

4.2. Evaluation on FRIDA dataset

FRIDA [35] provides 66 synthetic images using SiVIC software to build a virtual world, which can be used to evaluate the performance of the automatic driving system in a systematic way. Unlike Fattal's dataset [6], the data of FRIDA [35] are from the virtual world. In addition, FRIDA [35] provides different kinds of fog, which includes uniform

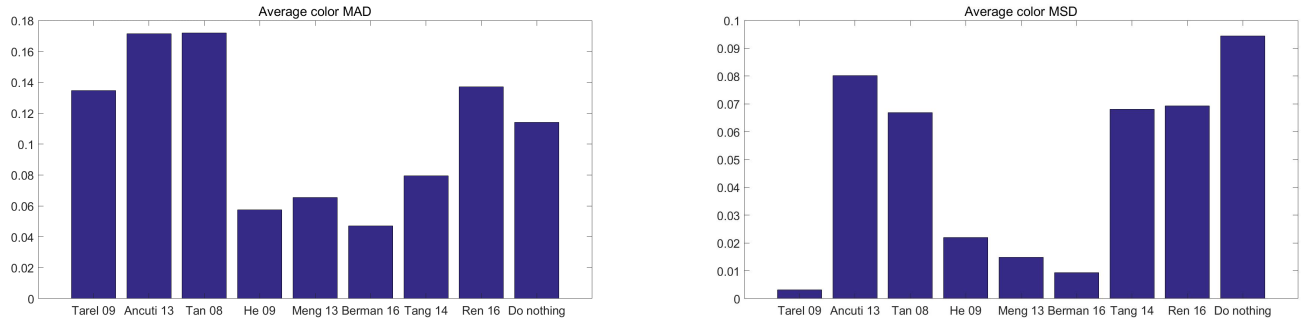

Figure 3. The average performance of different dehazing methods on fattal's [6].

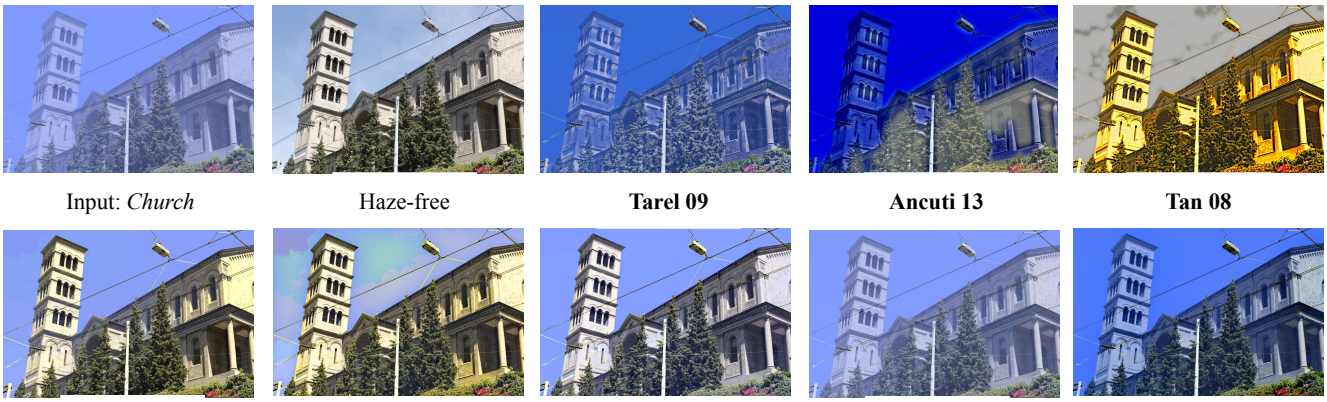

Figure 4. Final haze removal results on the *church* case.

fog, heterogeneous fog, cloudy fog, and cloudy heterogeneous fog. In the experiment, we randomly sample 17 images using uniform fog from the database. Koschmieder's law is applied to produce uniform fog with meteorological visibility distance of 80m in **FRIDA** [35].

We applied the same eight dehazing methods to these 17 images. In the experiment, the dataset does not provide airlight. Therefore, we set airlight as [0.85, 0.85, 0.85] for **Tan 08** [32], **He 09** [12] and **Tang 14** [33]. For other methods, we set it as default because that they can detect airlight automatically. The result is shown in Fig. 5. Examples of the result for different dehazing methods is shown in the Fig. 6. It can be seen that **Meng 13** [22], **Berman 16** [3] and **Ren 16** [26] obtain less error in terms of MAD. **He 09** [12] and **Tang 14** [33] obtain more error than other methods. In terms of MSD, **Ancuti 13** [1], **Meng 13** [22] and **Meng 13** [3] rank top place while **Tang 14** [33] and **Tarel 09** [34] obtain more errors. Compare to doing nothing, the average difference us always improved. The estimation could be influenced by the different value of airlight. It may also be influenced by sky region. **FRIDA** [35] does not provide non-sky mask, which could affect the evaluation.

4.3. Evaluation on Our dataset

Our dataset contains more than 2000 images. In this experiment, we sampled 30 frames from our dataset and we applied each method on our dataset. The frames were captured from the view of a driving car in Asian cities. All frames contain large depth variation. It is assumed that space is filled with uniform haze density. Each image has its corresponding non-sky mask, which is used in reducing the error of estimation of evaluation. In addition, each image has three different level of fog and haze, low level, medium level and high level.

We have evaluated the eight methods on our dataset and we quantify the visibility enhancement output by comparing them with ground truth. MAD and MSD are used to measure the closeness of the results to the ground truth pixel by pixel. Fig. 7 shows the performance of each method at the different level of haze and fog in terms of MAD and MSD. As it shows that **Berman 16** [3], **Tang 14** [33] and **Ren 16** [26] perform better than other methods. One example of the results is shown in Fig. 8.

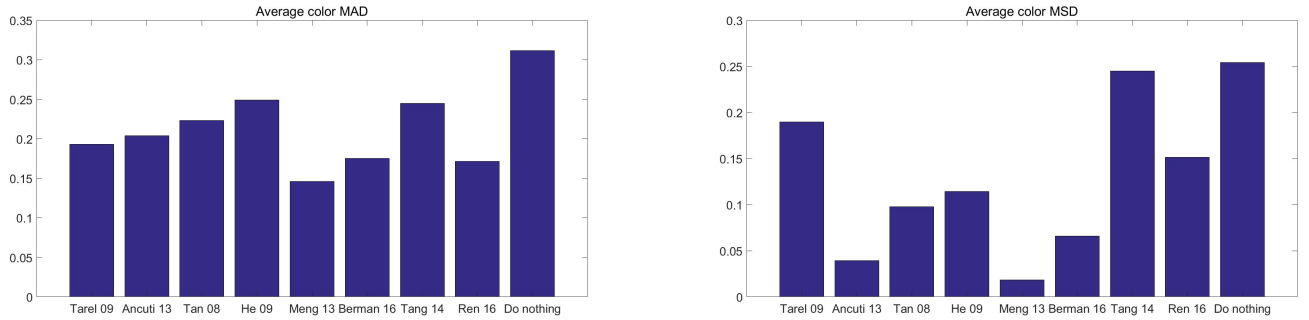

Figure 5. The average performance of different dehazing methods on FRIDA dataset. [35]

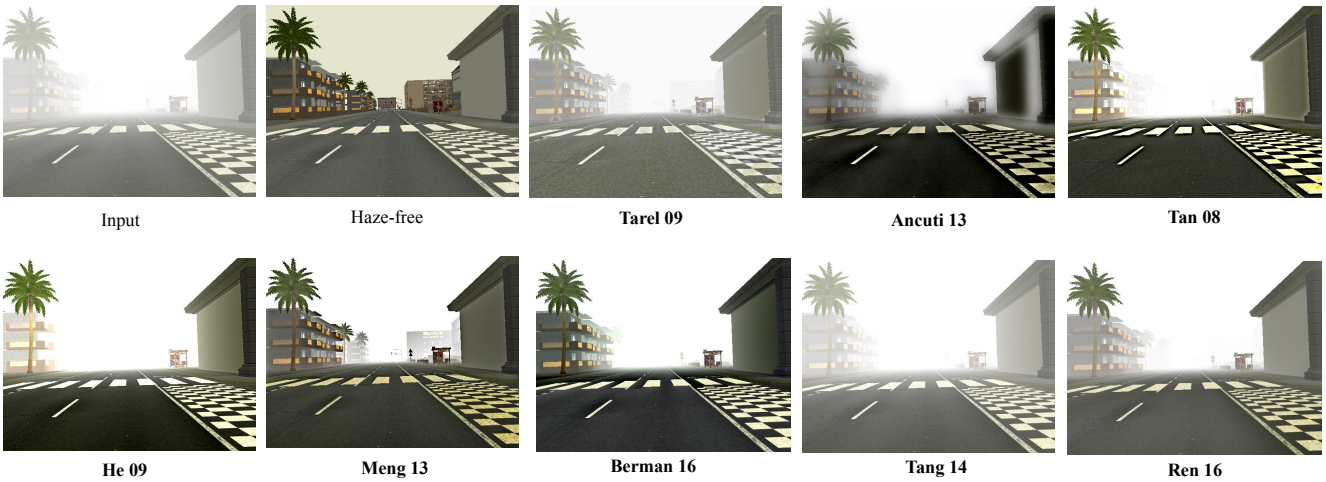

Figure 6. Final haze removal result on *U080-000001* case.

5. Conclusion

In this work, we introduce a new, dynamic, and large volume road scene dataset for data-driven computer vision research in bad weather, which is built through using modern computer graphics technology and image process techniques. The dataset provides road images with the different levels of synthetic fog, depth maps, and masks for sky region. We also discuss the comparison between our dataset and other foggy and hazy datasets. Compared with Fattal's data [6], our dataset has large volume data and different levels of fog for evaluation and validation. Compared with **FRIDA** [35], our dataset is generated using physics-based rendering which tend to be more realistic, and ours has much larger volume data with non-sky mask to evaluate dehazing algorithms comprehensively. In addition, we conducted quantitative benchmark for the most representative single image hazing methods. We found that recent work **Berman 16** [3] and **Ren 16** [26] generally perform better in the dehazing task.

There are a few future directions after this work. First, we are planning to extend the set of synthetic images by using more models of different cities. Second, in the paper we only benchmark the dehazing task. We have plan to label ground truth for other computer vision tasks like semantic segmentation, optical flow, and add to the dataset. Third, we use the uniform haze model in the rendering. As suggested in [8] and [35], the fog model can be improved and produces more complicated foggy or hazy effect by introducing more parameters. Moreover, this work only consider haze condition, other bad weather problems, *e.g.* rain and snow [17] [18] [42] [43] [44], nighttime haze [16] could be included in the extension of our dataset.

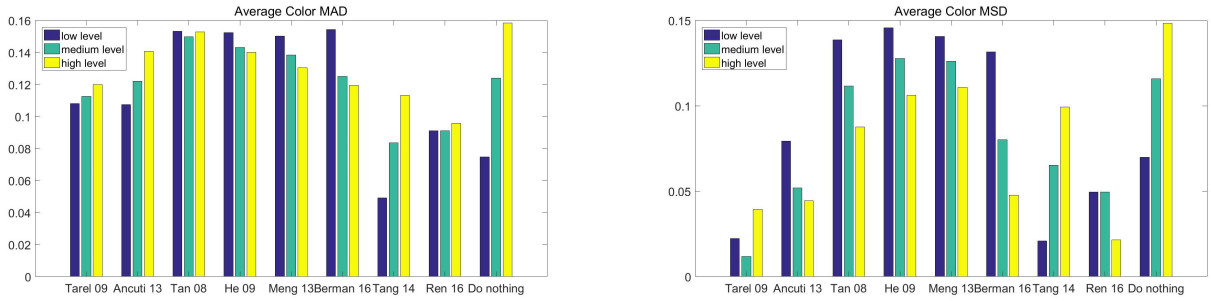

Figure 7. Comparisons of the results of different methods on different haze levels in our dataset in terms of MSD and MAD.

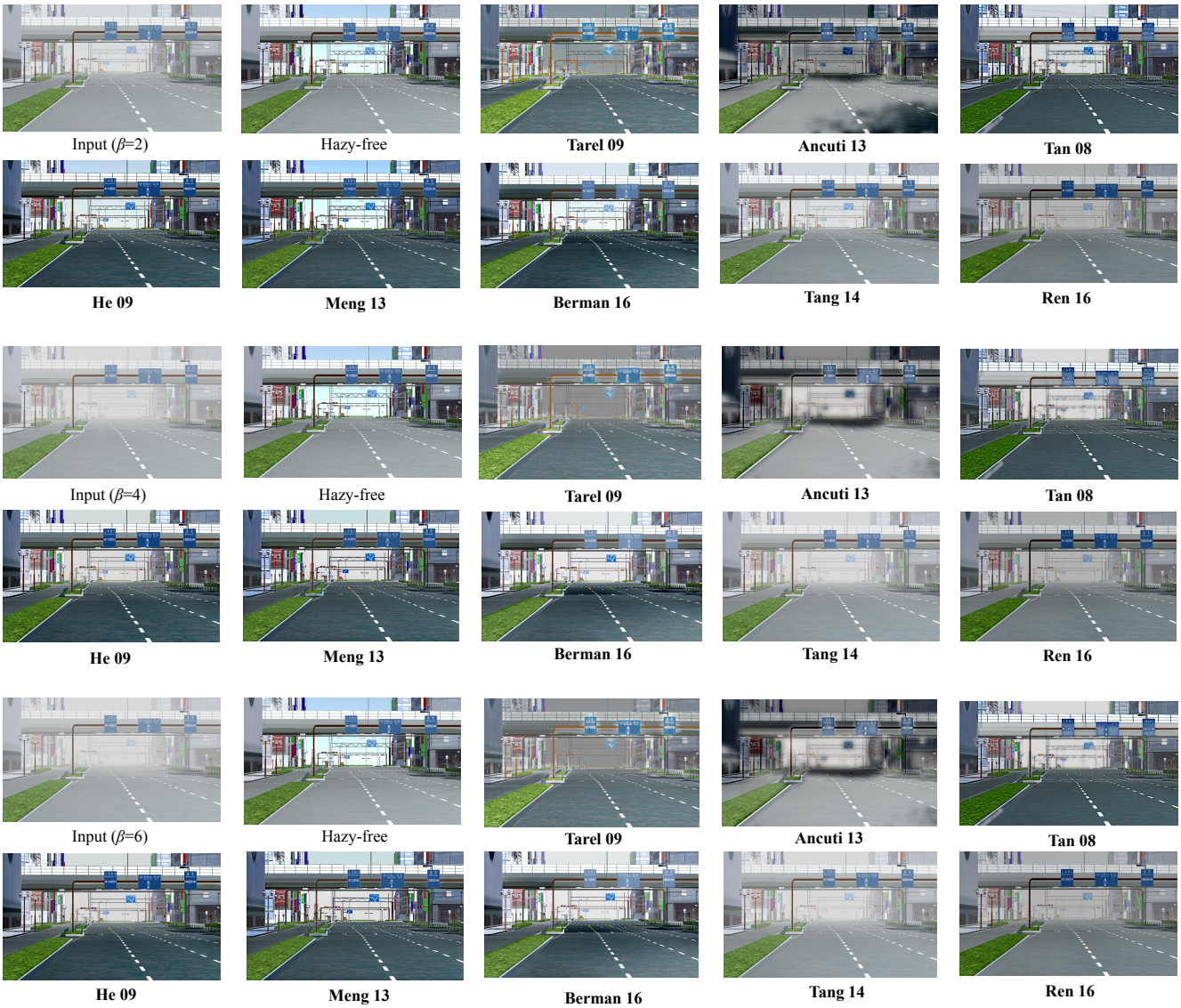

Figure 8. The result of one frame is shown as an example. First row and second row are visibility enhancement results on the sample synthetic image with low level of fog in our dataset. Third row and second row are visibility enhancement results on the sample synthetic image with medium level of fog in our dataset. Fifth row and sixth row are visibility enhancement results on the sample synthetic image with high level of fog in our dataset.

References

[1] C. O. Ancuti and C. Ancuti. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing*, 2013.

[2] N. Barnes, G. Loy, and D. Shaw. The regular polygon detector. *Pattern Recognition*, 2010.

[3] D. Berman, S. Avidan, et al. Non-local image dehazing. In *CVPR*, 2016.

[4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *CVPR*, 2015.

[5] 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 *CVPR*, 2016.

[6] R. Fattal. Dehazing using color-lines. *ACM Transactions on Graphics (TOG)*, 2014.

[7] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[8] M. Gazzi, T. Georgiadis, and V. Vicentini. Distant contrast measurements through fog and thick haze. *Atmospheric Environment*, 2001.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[10] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV*, 2003.

[11] H. Hattori, V. Naresh Boddeti, K. M. Kitani, and T. Kanade. Learning scene-specific pedestrian detectors without real data. In *CVPR*, 2015.

[12] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Y. Li, F. Guo, R. T. Tan, and M. S. Brown. A contrast enhancement framework with jpeg artifacts suppression. In *ECCV*, 2014.

[16] Y. Li, R. T. Tan, and M. S. Brown. Nighttime haze removal with glow and multiple light colors. In *ICCV*, 2015.

[17] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, 2016.

[18] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Single image rain streak separation using layer priors. *IEEE Transactions on Image Processing*, 2017.

[19] Y. Li, S. You, M. S. Brown, and R. T. Tan. Haze visibility enhancement: A survey and quantitative benchmarking. *arXiv preprint arXiv:1607.06235*, 2016.

[20] J. Marin, D. Vázquez, D. Gerónimo, and A. M. López. Learning appearance in virtual scenarios for pedestrian detection. In *CVPR*, 2010.

[21] E. J. McCartney. Optics of the atmosphere: scattering by molecules and particles. *New York, John Wiley and Sons, Inc., 1976. 421 p.*, 1976.

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

[23] S. G. Narasimhan, C. Wang, and S. K. Nayar. All the images of an outdoor scene. In *ECCV*, 2002.

[24] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[26] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 2016.

[27] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015.

[29] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3d models. In *BMVC*, 2012.

[30] D. Shaw, N. Barnes, et al. Perspective rectangle detection. In *Proceedings of the Workshop of the Application of Computer Vision, in conjunction with ECCV*, 2006.

[31] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010.

[32] R. T. Tan. Visibility in bad weather from a single image. In *CVPR*, 2008.

[33] K. Tang, J. Yang, and J. Wang. Investigating haze-relevant features in a learning framework for image dehazing. In *CVPR*, 2014.

[34] J.-P. Tarel and N. Hautiere. Fast visibility restoration from a single color or gray level image. In *ICCV*, 2009.

[35] J.-P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 2012.

[36] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *CVPR*, 2007.

[37] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *Image and Vision Computing New Zealand*, 2008.

[38] D. Vernon, G. Metta, and G. Sandini. A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, 2007.

[39] P. Wang, C. Shen, N. Barnes, and H. Zheng. Fast and robust object detection using asymmetric totally corrective boosting. *IEEE Transactions on Neural Networks and Learning Systems*, 2012.

[40] N. Wong-Anan. Worst haze from indonesia in 4 years hits neighbors hard. *Reuters*, 2010.

[41] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, 2015.

[42] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop detection and removal in video. *CVPR*, 2013.

[43] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Raindrop detection and removal from long range trajectories. In *ACCV*, 2014.

[44] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.