FoggyStereo: Stereo Matching with Fog Volume Representation

Chengtang Yao[1,2], Lidong Yu[2]
[1]Beijing Laboratory of Intelligent Information Technology, Beijing Institute of Technology,
[2]Autonomous Driving Algorithm, NIO
yao.c.t@bit.edu.cn, yvlidong@gmail.com

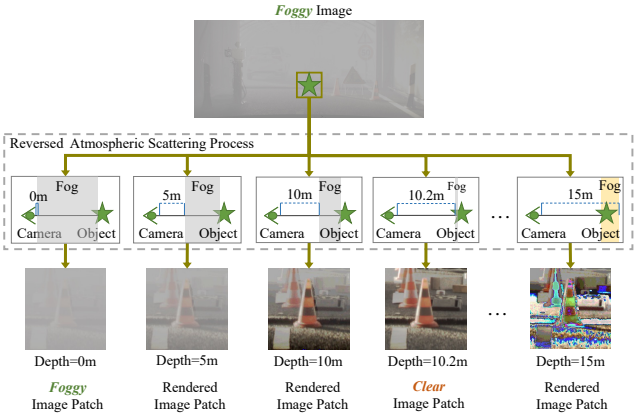
Abstract

Stereo matching in foggy scenes is challenging as the scattering effect of fog blurs the image and makes the matching ambiguous. Prior methods deem the fog as noise and discard it before matching. Different from them, we propose to explore depth hints from fog and improve stereo matching via these hints. The exploration of depth hints is designed from the perspective of rendering. The rendering is conducted by reversing the atmospheric scattering process and removing the fog within a selected depth range. The quality of the rendered image reflects the correctness of the selected depth, as the closer it is to the real depth, the clearer the rendered image is. We introduce a fog volume representation to collect these depth hints from the fog. We construct the fog volume by stacking images rendered with depths computed from disparity candidates that are also used to build the cost volume. We fuse the fog volume with cost volume to rectify the ambiguous matching caused by fog. Experiments show that our fog volume representation significantly promotes the SOTA result on foggy scenes by $10\% \sim 30\%$ *while maintaining a comparable performance in clear scenes.*

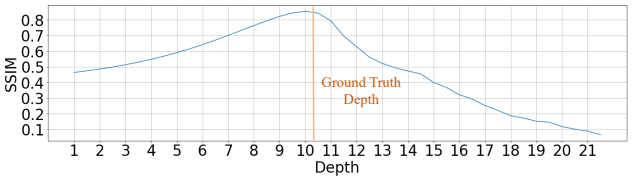
1. Introduction

Stereo matching is a pixel-wise labeling task relying on discriminative features to achieve accurate results. The discriminative features can be well extracted by existing methods in clear scenes [3, 4, 7, 15, 16]. However, it is unavoidable to encounter foggy or foggy-like scenes in the real world. The fog blurs the image and makes the features indiscriminative for stereo matching. The ambiguous matching result caused by fog restricts the application of stereo matching.

Prior methods deem fog as a noise and discard it to improve matching results [12, 26, 27, 36]. Different from them, we propose to take advantage of fog and explore depth hints for stereo matching. The intuitive observation comes from the fog rendering process. During rendering, fog is accumulated along the light path between objects and camera following the physical atmospheric scattering process [24, 34, 38]. Different depths will lead to different brightness and blur the image at different levels. So when we render the image by reversing the atmospheric scattering process, fog is removed within a selected depth range. As presented in Fig. 1a, only the depth close to the real depth will lead to a clear image. In other words, the quality of the rendered image indicates the correctness of depth used in the render-



(a) The process and results of rendering.



(b) The distribution of $SSIM \sim Depth$.

Figure 1. The Visualization of depth hints from fog. (a) We reverse the atmospheric scattering process by removing the fog among different depth ranges. Only the depth near the ground truth leads to a clear image. We deem this observation as the depth hint. (b) We further illustrate the distribution of rendered image quality in depth. We measure the image quality via the structural similarity (SSIM) metric. We find that the closer the depth candidate is to the ground truth, the better the rendered image quality is.

ing process, which is illustrated in Fig. 1b.

Based on the above observation, we introduce a fog volume representation to collect depth hints from the fog. The fog volume is built along with the cost volume using the same disparity candidates. When we sample a disparity candidate for the cost volume, we also validate its correctness by the fog volume. The fog volume representation is constructed in three steps. We first learn the parameters of the atmospheric scattering process from the left image, including the global atmospheric light and the atmospheric attenuation coefficient. Then we render a series of left images with atmospheric parameters and sampled disparity candidates by reversing the atmospheric scattering process. Finally, the rendered images are stacked together to build our fog volume. We use a 3D convolutional network on the fog volume to learn to validate the sampled disparities.

Our fog volume provides great depth hints in foggy areas where existing cost volume loses effectiveness due to image degradation. The cost volume, instead, is more suitable in good visible areas [3, 7, 11]. In order to take advantage of both kinds of volumes, we fuse them through the volume uncertainty. The volume uncertainty is computed from the variance of two volumes along the disparity dimension.

We validate our method on both synthetic and natural foggy scenes. Our method outperforms the state-of-the-art approaches in foggy scenes by more than 10% while keeping comparable performance in clear scenes. We test the ability of our method in different depth ranges and fog thicknesss to demonstrate the potential application of our method in the real world.

2. Related Work

2.1. Stereo Matching

Stereo matching has been studied for decades to get an accurate and dense matching result [21, 29, 37]. Traditional methods [1, 15, 17, 33, 40] mainly use hand-crafted features and rely on optimization/aggregation and refinement to obtain accurate dense correspondence. In recent years, state-of-the-art methods [3, 4, 7, 13, 16, 30, 39] mostly use deep neural network to learn discriminative features and rely on the 3D cost volume to rectify the matching result. Although both traditional and deep-learning-based methods have achieved significant improvement, their performance is mainly promised in clear scenes. Their matching results are severely degraded when facing foggy or foggy-like scenes.

2.2. Stereo Matching in Foggy Scenes

In order to solve the ambiguous matching problem caused by fog, prior methods mainly regard the fog as noise and discard it before stereo matching. They aim to remove fog from images or learn a noise-robust model to compute the disparity. The first approaches usually discard the fog in left and right images using dehazing methods [26, 27] or specifically-designed hardware [12, 36]. They believe clear images could be obtained, and the left-right consistency in image quality could be preserved after the restoration. The second approaches focus on the design of optimization to learn a noise-robust stereo matching model [28, 35]. These methods assume that a noise-robust model could be learned in synthetic data and fast adapt to the real world. However, the above methods only take the fog as noise and miss the beneficial depth hints from fog where nearby objects are clearer than distant objects. Compared to them, some methods notice that fog can help stereo matching. They mainly take advantage of fog in two kinds of approaches: feature fusion and objective function. The first kind of approach jointly conducts stereo matching and dehazing task, where depth hints are assumed to be explored in the features learning process [19, 31]. The other methods [2, 25] integrate the learning of atmospheric scattering parameters with stereo matching as an additional constraint in the objective function of optimization.

In this paper, we present a new view to exploring depth hints of fog. We find that through the reversed atmospheric scattering process, we can check the quality of the rendered image and verify the correctness of the disparity. Thus, our fog volume representation collects these depth hints explicitly and facilitates the learning of disparity estimation.

2.3. Volumetric Fog Rendering

Foggy or foggy-like scenes are commonly rendered according to the physical model of atmospheric scattering process [6, 9, 38]. The scattering light from the fog particles is accumulated along the light path when a 3D object is projected to an image plane, called volume ray marching. Recently, neural volume based methods [18, 20] have achieved good performance in scattering media. They use the differentiable ray marching algorithm to learn a renderable volume. In this paper, we build the fog volume representation by reversing the atmospheric scattering process. It collects depth hits from fog and fuses with cost volume to refine the disparity estimation on blurred areas.

3. Method

As shown in Fig. 2, we extract the features from left and right images taken with a calibrated camera. We estimate the atmospheric light L_∞ and the attenuation coefficient β from the left image. We then warp the extracted features to build a cost volume based on the sampled disparities $\{D^i\}_{i=0}^{i=N-1}$. The disparities are also transformed into depth $\{Z^i\}_{i=0}^{i=N-1}$ and used to render the image with L_∞ and β. We gather a set of rendered images and concatenate them to construct the fog volume. The fog volume is later fused with cost volume to rectify the matching results.

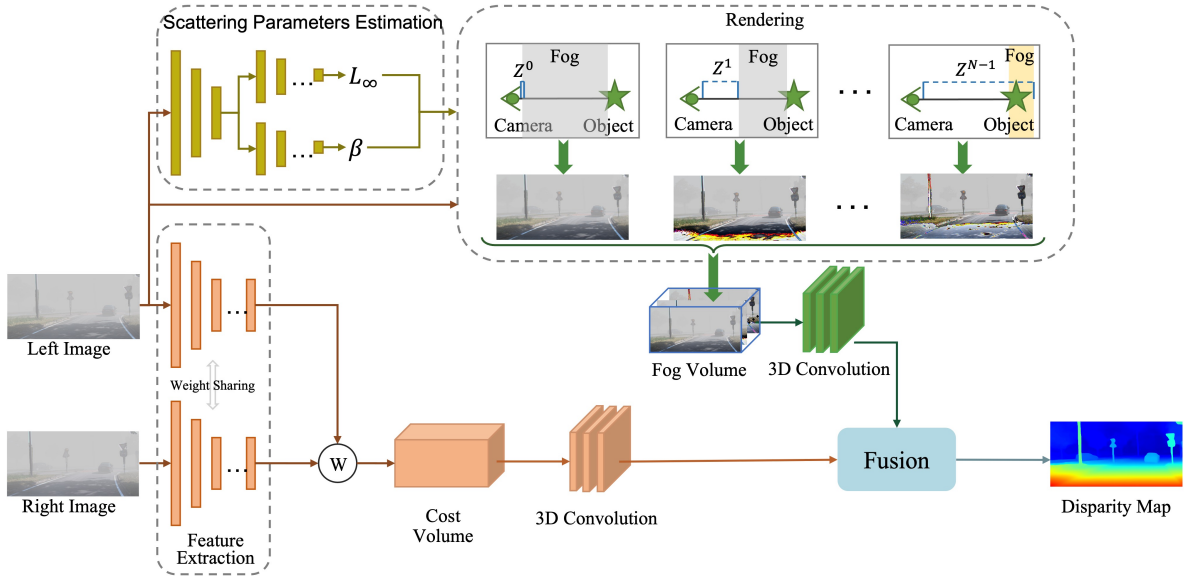


Figure 2. The overview of our method. We extract the features from left and right images to build the cost volume via warping Ⓦ. We predict the atmospheric light L_∞ and attenuation coefficient β from the left image to render a series of images with different depth Z_i. The rendered images are concatenated along channel dimension and fused with cost volume for disparity estimation.

In the following content, we will focus on the presentation of the fog volume construction, the fusion of cost volume and fog volume, and the loss functions. For other specific architectures, please refer to our supplemental materials.

3.1. Fog Volume Representation

As aforementioned, we find that the quality of rendered images indicates the correctness of depth used in the rendering process. Based on this observation, we propose a fog volume representation to explore the depth hints of fog. The fog volume is constructed by stacking a series of images rendered with different depths.

Rendering We render the image by reversing the atmospheric scattering process. In foggy or foggy-like scenes, the interaction of photons and particles in the transporting media results in the atmospheric scattering effect [24, 34, 38]. The atmospheric scattering effect causes the attenuation of light reflected from objects and the accumulation of environmental light. The attenuation and accumulation jointly determine the degradation of image quality.

The attenuation T from an object to the camera is commonly measured by the Beer-Lambert-Bouguer law :

$$T(Z_x) = e^{-\int_0^{Z_x} \beta(z)dz}, \tag{1}$$

where Z_x is the distance between the camera and the point of object at pixel x, $\beta(\cdot)$ is the attenuation coefficient. The attenuated light L_t projected at pixel x is then computed as

$$L_t(x) = L_\infty \, \rho(x) \, T(Z_x). \tag{2}$$

L_∞ is the atmospheric light, and $\rho(x)$ is the reflectance of pixel x on the object surface. The accumulation of environmental light makes the brightness of the object increase with the depth:

$$L_c(x) = L_\infty(1 - T(Z_x)), \tag{3}$$

where $L_c(x)$ is the accumulated light projected onto pixel x. Then, the final intensity I captured by camera is formulated as the sum of L_t and L_c:

$$\begin{aligned} I(x) &= L_t(x) + L_c(x) \\ &= J(x)T(Z_x) + L_\infty(1 - T(Z_x)), \end{aligned} \tag{4}$$

where $J(x) = L_\infty \rho(x)$ represents the data in clear scene and $I(x)$ represents the data in foggy scene.

The Eq. (4) shows that the image degradation in the foggy scene is related to the scene depth Z. We thus render images by reversing the atmospheric scattering process with different depth candidates Z_x^i to explore depth hints of fog, where the closer the depth candidate is to the ground truth, the better the rendered image quality is. This process is formulated as

$$R(x, Z_x^i) = \frac{I(x) - L_\infty(1 - T(Z_x^i))}{T(Z_x^i)}. \tag{5}$$

This process is a natural formulation of the reversed atmospheric scattering process, but it is difficult to learn the rendering of an image via this equation, as $R(x, Z_x^i)$ is changed exponentially with the increase of Z_x^i :

$$R(x, Z_x^i) = e^{\int_0^{Z_x^i} \beta(z)dz} I(x) - L_\infty(e^{\int_0^{Z_x^i} \beta(z)dz} - 1). \tag{6}$$

Once we select a large depth candidate, the gradient will explode and the learning becomes instable. In order to solve this problem, we conduct the learning in a logarithmic space:

$$R(x, Z_x^i) = ln(|I(x) - L_\infty|) + \int_0^{Z_x^i} \beta(z)dz. \quad (7)$$

In the following content, the render image is computed via Eq. (7) to build the fog volume representation.

Scattering Parameters Estimation As illustrated in Eq. (7), we use two scattering parameters to render an image, including the atmospheric light L_∞ and the attenuation coefficient β. Following prior methods [24, 34], we set L_∞ and β as global parameters under the condition of one single light source and a homogeneous transporting medium. We learn the global parameters from the left image by a fully convolutional network. The network contains a basic feature extraction module following two branches as shown in Fig. 2. Each branch outputs the atmospheric light parameter and the attenuation coefficient.

Disparity Candidates Sampling We sample disparity candidates to construct both cost volume and fog volume. Specifically, we use a network to predict the minimum disparity candidate D_x^{min} and maximum disparity candidate D_x^{max} for each pixel following DeepPruner [7]. We then uniformly sample disparity candidates D_x^i between D_x^{min} and D_x^{max} in N times. After obtaining the disparity candidates $\{D_x^i\}_{i=0}^{i=N-1}$, we compute the depth candidates $\{Z_x^i\}_{i=0}^{i=N-1}$ according to the epipolar geometry using the focal length and the baseline of camera.

Rendered Images Gathering As we set L_∞ and β as global parameters, β becomes constant for different depth. The Eq. (7) is updated as

$$R(x, Z_x^i) = ln(|I(x) - L_\infty| + \epsilon) + \beta Z_x^i, \quad (8)$$

where the ϵ is a constant for the numerical stability. We then render a series of images with the sampled depths according to Eq. (8) and construct the fog volume representation $\mathcal{V}_f$ by stacking rendered images:

$$\mathcal{V}_f(x, Z) = [R(x, Z_x^0), R(x, Z_x^1), \cdots, R(x, Z_x^{N-1})], \quad (9)$$

where $[\cdot]$ is the concatenation operation. As illustrated in Fig. 2, the fog volume is subsequently input into a 3D convolution network to explore the depth hints of fog through the quality change of rendered images.

3.2. Fusion

In foggy scenes, the cost volume works well among clear areas where the discriminative feature can be easily learned, but it becomes less effective in blurred areas. In these areas, the fog volume provides depth hints of fog, where the quality of the rendered image can validate the disparity candidate. In order to take advantage of the two kinds of volume, we fuse them together and guide the network to rely on cost volume in clear areas and use the fog volume to rectify the ambiguous matching in blurred areas.

Specifically, we use uncertainty to measure the confidence of cost volume and fog volume in different areas. We compute the variance σ of cost volume $\mathcal{V}_c$ and fog volume $\mathcal{V}_f$ along disparity dimension as the uncertainty:

$$\begin{aligned} \sigma(x, D_i) &= \sum_{i=0}^{i=N-1} P(x, D_i)(D_i - \mu_D)^2, \\ \mu_D &= \frac{1}{N} \sum_{i=0}^{i=N-1} D_i, \end{aligned} \quad (10)$$

where $P(x, D_i)$ is the probability volume of $\mathcal{V}$:

$$P(x, D_i) = \frac{e^{\mathcal{V}(x, D_i)}}{\sum_{i=0}^{i=N-1} e^{\mathcal{V}(x, D_i)}}, \quad (11)$$

Then, the fusion of $\mathcal{V}_c$ and $\mathcal{V}_f$ is realized through

$$\tilde{\mathcal{V}}(x, D_i) = [\sigma_c(x, D_i)\mathcal{V}_c(x, D_i), \sigma_f(x, D_i)\mathcal{V}_f(x, D_i)]. \quad (12)$$

As presented in Fig. 2, the fused volume $\tilde{\mathcal{V}}$ is subsequently input into a 3D convolution network to jointly leverage the beneficial information from both volumes for the disparity estimation.

3.3. Loss Function

In the above sections, we estimate the scattering parameters $\widetilde{L}_\infty$ and $\widetilde{\beta}$ for the rendering of images and predict a disparity map $\widetilde{D}$ as the final output. This section introduces the loss function we used to guide the learning of L_∞, β and $\widetilde{D}$. The predicted disparity map $\widetilde{D}$ is supervised by the ground truth disparity map D using L_1 loss:

$$\mathcal{L}_0 = L_1(D, \widetilde{D}). \quad (13)$$

For L_∞ and β, we use the reconstruction loss of clear image in a supervised manner. We first obtain the rendered image $\widetilde{R}$ according to Eq. (8) with the predicted dense disparity map $\widetilde{D}$, and convert the left clear image J into logarithmic space via

$$J'(x) = ln(|J(x) - L_\infty|). \quad (14)$$

We then use the $J'(x)$ to supervise the rendered image both in RGB space and gray space by the L_1 loss :

$$\mathcal{L}_1 = L_1(\widetilde{R}, J') + L_1(\widetilde{R}_{gray}, J'_{gray}). \quad (15)$$

We also design an unsupervised learning strategy for $\widetilde{L}_\infty$. When Z_x is large, L_∞ is approximately equal to $I(x)$. So we compute the average intensity of pixels whose disparity is smaller than 1.5 as the pseudo ground truth of $\bar{L}_\infty$. $\bar{L}_\infty$ is

Testing	Metrics	Stereo		Joint		Sequential	Ours
		PSMNet* [3]	DeepPruner* [7]	SDNet [32]	SSMDNet [31]	4Kdehazing [41] + DeepPruner [7]	
Clear	EPE	0.99	0.98	-	-	1.19	**0.81**
	3px (%)	**4.1**	5.30	-	-	6.2	**4.5**
Foggy	EPE	1.27	3.77	2.68	2.23	1.49	**1.04**
	3px (%)	8.1	14.10	26.43	9.71	10.30	**7.2**

Table 1. The comparison of algorithms on the SceneFlow dataset. We compare the results testing on clear data and foggy data. * represents our re-implementation results.

Methods		KITTI 2015				KITTI 2012			
		Foggy		Clear		Foggy		Clear	
		3px (%)	EPE	3px (%)	EPE	3px (%)	EPE	3px (%)	EPE
Stereo	PSMNet* [3]	1.3	0.54	**1.0**	0.49	3.3	0.84	3.3	0.86
	DeepPruner* [7]	3.7	0.88	8.8	1.66	4.3	0.94	5.0	1.09
Joint	SDNet [32]	13.4	1.73	-	-	11.0*	1.63*	10.7*	1.60*
	SSMDNet [31]	10.8	1.23	-	-	9.7*	1.55*	9.5*	1.53*
Sequential	4Kdehazing [41] + DeepPruner [7]	7.3	0.951	**1.1**	0.49	3.2	0.91	3.2	0.89
ours		**1.2**	**0.51**	**1.1**	**0.47**	**2.7**	**0.77**	**2.7**	**0.78**

Table 2. The comparison of algorithms on KITTI 2015 and 2012 datasets. * represents our re-implementation results.

then used to supervise the learning of $\widetilde{L}_\infty$ with the L_1 loss:

$$\mathcal{L}_2 = L_1(\bar{L}_\infty, \widetilde{L}_\infty). \tag{16}$$

The final loss is the sum of $\mathcal{L}_0$, $\mathcal{L}_1$ and $\mathcal{L}_2$ with weights γ_0 and γ_1:

$$\mathcal{L} = \gamma_0\mathcal{L}_0 + \gamma_1(\mathcal{L}_1 + \mathcal{L}_2). \tag{17}$$

4. Experiments

4.1. Datasets

Sceneflow Sceneflow [22] is a synthetic dataset, containing more than 39000 stereo frames with a resolution of 960×540. It provides dense ground truth disparity rendered from clear scenes. The dataset contains three scenarios, where 35454 and 4370 image pairs are used for training and testing, respectively.

KITTI 2012 & 2015 KITTI 2012 [10] and 2015 [23] are real-world datasets with an image resolution of 1240×376 and the sparse ground truth disparity collected by Lidar. In KITTI 2012, there are 194 pairs of training images and 200 pairs of test images. In KITTI 2015, 200 pairs of images are used for training and testing, respectively.

PixelAccurateDepth PixelAccurateDepth [11] is a real-world dataset where four typical automotive outdoor scenarios are built, including pedestrian zone, residential area, construction area, and highway. There are 1,600 samples with a resolution of 1730×734 collected under controlled weather (clear, rain, fog) and illumination (daytime, night), where 17 visibility levels are separated in fog (20-100m in 5m steps).

4.2. Implementation Details

We take DeepPruner [7] as a baseline and implement our method upon it. Thus, our network has a similar architecture of feature extraction and cost aggregation as DeepPruner. For more details, please refer to our supplemental materials and code[1].

Foggy Scene Synthesis We synthesize foggy images with datasets collected in clear scenes, including Sceneflow, KITTI 2012, and KITTI 2015. The foggy images are synthesized in left and right views for training and testing. We conduct the synthesis with Eq. (4) following previous methods [8,31] We use dense ground truth disparity maps for the synthesis in the Sceneflow dataset. As for the KITTI 2012 & 2015 datasets only containing sparse ground truth, we use LEAStereo [5] and their pre-trained model to generate the pseudo dense disparity maps for the foggy data synthesis.

Training We train our model in two crop sizes, 256×512 and 512×512, using Adam optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. During the training, we first normalize the image into $[0, 1]$ and then randomly synthesize the foggy images with $L_\infty \in (0.7, 1)$ and $\beta \in (0, 0.1)$ for all datasets. The number ratio of clear data and foggy data is set as $7 : 3$. We use hyperparameter $\gamma_0 = 1.0$, $\gamma_1 = 0.05$ for all datasets,. We train the model from scratch on the Sceneflow dataset with 100 epochs and an initial learning rate of 0.001. We then use the model pre-trained in the Sceneflow dataset and finetune it on the KITTI 2015 training set for 1000 epochs with an initial learning rate of 0.0001. As for the PixelAccurateDepth dataset, we use the model pre-trained in the KITTI 2015 dataset. Following the training protocol of the PixelAccurateDepth dataset, we finetune

[1] https://yaochengtang.github.io/FoggyStereo-Stereo-Matching-with-Fog-Volume-Representation/

Method		RMSE ↓	tRMSE ↓	MAE ↓	tMAE ↓	logRMSE ↓	SRD ↓	ARD ↓	SIlog ↓	δ_1 (%) ↑	δ_2 (%) ↑	δ_3(%) ↑
Stereo	SGM [14]	1.90	1.40	0.96	0.86	0.14	0.27	8.12	13.32	90.74	98.44	99.50
	PSMNet [3]	2.75	1.96	1.44	1.22	0.18	0.56	9.91	16.07	89.14	97.21	98.80
	DeepPruner* [7]	1.81	1.37	0.80	0.70	0.12	0.21	5.52	11.78	93.57	98.08	99.50
Joint	SDNet* [32]	1.89	1.53	1.03	0.94	0.13	0.26	7.94	12.87	92.52	98.22	**99.57**
	SSMDNet* [31]	1.95	1.53	1.00	0.90	0.12	0.22	7.05	12.17	92.75	**98.53**	**99.68**
Sequential	4Kdehazing [41] + DeepPruner [7]	**1.79**	1.32	0.77	0.67	**0.11**	**0.19**	5.12	**10.95**	**94.41**	**98.45**	**99.66**
Lidar (int.) [11]		1.89	1.36	**0.70**	**0.59**	0.13	0.23	**4.78**	12.58	93.62	98.13	99.36
RGB+Lidar [11]		3.05	2.04	1.61	1.29	0.26	0.53	10.85	24.01	84.69	94.77	97.05
Ours		**1.82**	**1.31**	**0.75**	**0.64**	**0.11**	**0.20**	**5.01**	**11.11**	**94.07**	**98.45**	**99.56**

Table 3. The comparison of algorithms on the clear data of PixelAccurateDepth dataset. * represents our re-implementation results.

Method		RMSE ↓	tRMSE ↓	MAE ↓	tMAE ↓	logRMSE ↓	SRD ↓	ARD ↓	SIlog ↓	δ_1 (%) ↑	δ_2 (%) ↑	δ_3(%) ↑
Stereo	SGM [14]	3.00	1.81	1.56	1.20	0.21	1.00	14.02	20.75	84.34	94.91	97.22
	PSMNet [3]	3.01	2.10	1.65	1.35	0.19	0.61	11.10	16.94	84.95	96.34	98.65
	DeepPruner* [7]	2.61	1.75	1.30	1.00	0.16	0.40	8.10	15.16	87.24	95.61	**98.92**
Joint	SDNet* [32]	2.63	1.88	1.48	1.22	0.18	0.47	10.67	16.86	85.83	95.70	98.50
	SSMDNet* [31]	2.69	1.83	1.42	1.13	0.17	0.42	9.23	16.12	87.42	96.13	98.54
Sequential	4Kdehazing [41] + DeepPruner [7]	3.32	1.81	1.69	1.06	0.23	0.76	9.91	20.71	85.08	92.13	95.01
Lidar (int.) [11]		3.67	2.01	1.68	1.13	0.39	0.91	12.21	35.19	80.57	87.27	91.66
RGB+Lidar [11]		3.81	2.52	2.34	1.83	0.35	0.91	16.88	28.67	69.77	85.16	92.74
Ours		**2.55**	**1.64**	**1.19**	**0.91**	**0.15**	**0.38**	**7.38**	**14.77**	**89.28**	**96.33**	**98.66**
Ours (PixelAccurateDepth Clear)		**1.74**	**1.20**	**0.80**	**0.61**	**0.10**	**0.22**	**4.50**	**9.04**	**93.14**	**97.42**	**99.72**

Table 4. The comparison of algorithms on the foggy data of PixelAccurateDepth dataset. * represents our re-implementation results.

the pre-trained model on the training set of Gated2Depth dataset [12] without any foggy image synthesis.

Evaluation We evaluate the performance of our method in four kinds of settings, (a) training and testing on Sceneflow dataset, (b) finetuning and testing on KITTI 2015, (c) finetuning on KITTI 2015 and testing on KITTI 2012, d) finetuning on Gated2Depth dataset and testing on PixelAccurateDepth dataset. In each testing stage, we evaluate the results in clear scenes and foggy scenes respectively, without any domain adaptation or post-processing. We use the End-Point-Error (EPE) and 3-pixel (3px) error rate as evaluation metrics in the first three settings. In the last setting, we follow the metrics used in PixelAccurateDepth [11], including the root mean squared error (RMSE), the root mean squared thresholded error (tRMSE), the mean absolute error (MAE), the mean absolute thresholded error (tMAE), the root mean squared logarithmic error (tRMSE), the root mean squared logarithmic error (logRMSE), the squared relative distance (SRD), the absolute relative distance (ARD), the scale invariant logarithmic error (SIlog), and the threshold metric $\delta_i < 1.25^i$ for $i \in \{1, 2, 3\}$.

Compared SOTA Methods We mainly compare to three kinds of methods, stereo matching designed for clear scenes [3, 7], jointly learning stereo matching and dehazing [31, 32], sequential learning dehazing and stereo matching [41]. We use '*Stereo*' to represent the first kind of methods, '*Joint*' to represent the second kind of methods, and '*Sequential*' to represent the last kind. It should be noted that we implement 4Kdehazing [41] + DeepPruner [7] for a fair comparison in the 'Sequential' method.

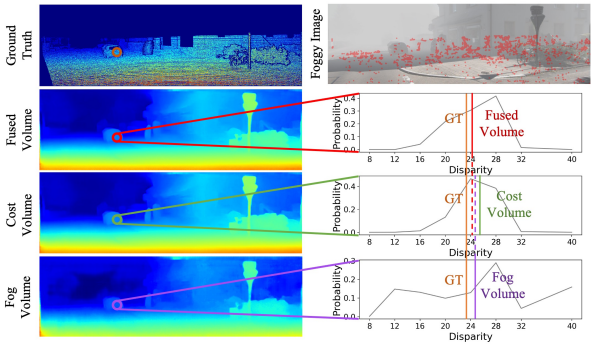


Figure 3. The visualization of three kinds of volume. The ground truth disparity map and corresponding foggy image are presented on the top. In the foggy image, red points illustrate areas where the results of fused volume are the best while the results of fog volume are better than that of cost volume. The disparity map computed from each volume is presented on the left row. The corresponding probability distribution of disparity candidates at the circled area is presented on the right side, where the ground truth and final predictions are illustrated via the vertical line in a different color.

4.3. Benchmark Performance

Sceneflow As shown in Tab. 1, our method achieve at least 20% improvement on EPE in foggy scenes when compared to 'Stereo', 'Joint', and 'Sequential' methods. Furthermore, we keep the good performance in clear scenes while the 'sequential' approach's performance has obvi-

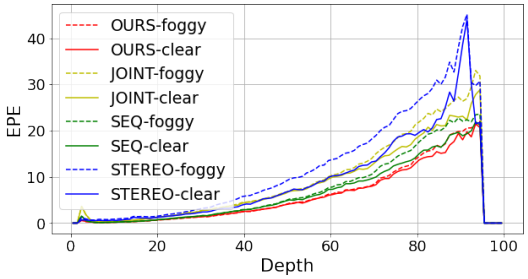


Figure 4. The visualization of error rate (EPE) distribution over the depth. The EPE is computed after transforming the predicted and ground-truth disparity into depth. 'OURS' represents the distribution result of our method. 'JOINT' represents the result of SSMDNet [31]. 'SEQ' represents the result of 4Kdehazing [41] + DeepPruner [7]. 'STEREO' represents the result of DeepPruner [7].

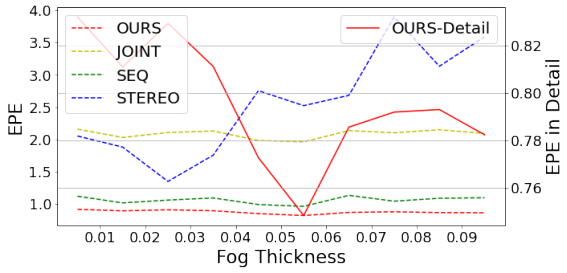


Figure 5. The error rate (EPE) distribution over the fog thickness. The fog thickness is defined as the attenuation coefficient β. The distribution of four kinds of methods are presented in dashed lines. The detailed EPE distribution of our method is presented with a solid line for better visualization. 'OURS' represents the result of our method. 'JOINT' represents the result of SSMDNet [31]. 'SEQ' represents the result of 4Kdehazing [41] + DeepPruner [7]. 'STEREO' represents the result of DeepPruner [7].

ously decreased. We also find that PSMNet is much better than DeepPruner in foggy scenes. The reason is that DeepPruner uses PatchMatch in the first disparity candidates generation, which is less powerful and robust than the full cost volume used in PSMNet. Although using the same disparity candidate sampling method, our method achieves almost 3 times improvement of accuracy with the fog volume compared to DeepPruner, which shows the power of the fog volume representation in foggy scenes.

KITTI 2012 & 2015 We use KITTI 2012 & 2015 with synthetic fog to validate our method in the real world. As shown in Tab. 2, we achieve the best result in foggy scenes and a comparable result in clear scenes. Similar to results on Sceneflow, the performance of PSMNet is much better than DeepPruner, which shows *the learning stability problem in mixed clear and foggy data*. Instead, our method promotes DeepPruner by 30% and achieves almost 10% improvement over PSMNet in KITTI 2012, which shows the fog volume representation can alleviate this problem to a large extent. For more details about the learning stability, please refer to the supplemental materials.

PixelAccurateDepth PixelAccurateDepth is a real-world dataset with different visibility of foggy scenes. We compare with SOTA methods on this dataset to illustrate the generalization ability of our method in real foggy scenes. As shown in Tab. 3, our method is almost the best except for the 'Sequential' method. The better performance of 'Sequential' method in clear scenes is due to the finetuning in the Gated2Depth dataset. We finetune 4Kdehazing + DeepPruner end-to-end without any foggy data. In this situation, the 4Kdehazing gradually becomes a feature extraction and improves the performance of DeepPruner. However, in foggy scenes, the 'Sequential' method becomes worse than DeepPruner, as shown in Tab. 4, while our method is the best among all the methods. *This phenomenon proves that our method is able to preserve the knowledge learned from prior data even after a long time of learning on a different dataset.* We also present the result of our method after finetuning with the clear data of the PixelAccurateDepth dataset. As shown in Tab. 4, we further achieve a great improvement, which means our method using synthetic data can generalize well to real foggy scenes.

4.4. Ablation Study and Analysis

Influence of Fusion In order to validate the effectiveness of fusion, we provide the visualization of the fog volume, cost volume, and fused volume. As shown in Fig. 3, the fog volume outperforms the cost volume in the red areas. According to the probability distribution of disparity candidates, the final disparity computed from the fused volume becomes closer to the ground truth. As for the quantity comparison, our method using fusion is much better than baseline DeepPruner as presented in Tab. 1∼Tab. 4.

Influence of Depth Range In order to show the robustness of our method in different depth ranges, we visualize the EPE error rate distribution over the depth in both clear and foggy scenes. As shown in Fig. 4, our method achieves similar performance in clear and foggy scenes, while other methods usually have very different curves in the two scenes, especially in the case of greater depth.

Influence of Fog Thickness In order to show the robustness of our method under different fog thicknesses, we define the thickness of fog as the attenuation coefficient β and present the EPE error rate distribution in different β. As shown in Fig. 5, our method achieves the best performance in different β, while DeepPruner [7] get a worse result as the fog thickness increases. We also find that our fog volume representation works well mostly at or around the interval $\beta \in [0.05, 0.06]$. As illustrated in Fig. 5, the performance of our method firstly gets better and then becomes a little worse. This change of performance is caused by the clip of

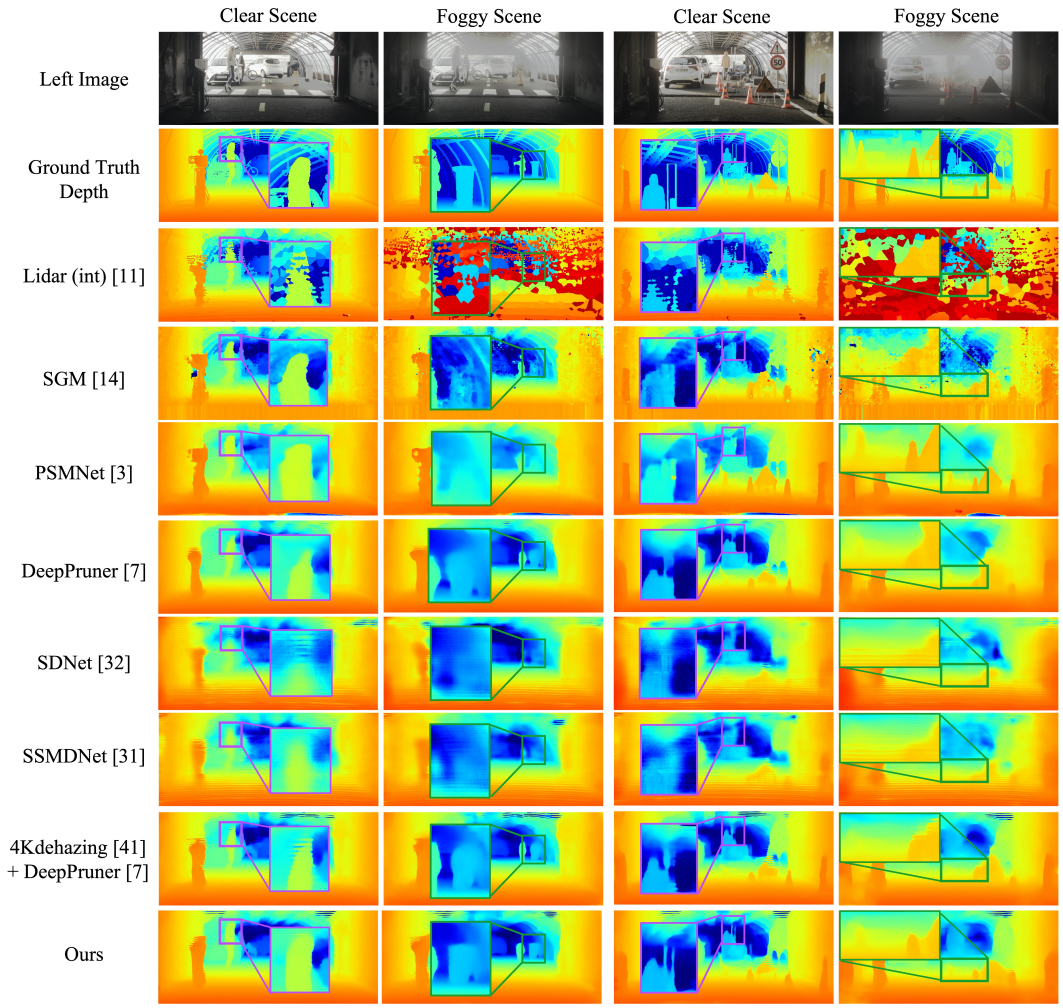


Figure 6. The visualization of depth map on PixelAccurateDeth dataset with real foggy scenes.

color value. *Since the storage of the color value is generally 8bit, the value will be clipped, resulting in loss of information, when the brightness of the fog accumulates too much.*

Visualization We provide the visualization of the depth map in the PixelAccurateDeth dataset to show the ability of our fog volume on distant objects. As shown in Fig. 6, the Lidar (int) achieves great results in clear scenes but loses effectiveness in foggy scenes. The deep learning based methods are more robust than Lidar (int), while our method is the best.

4.5. Limitations and Discussion

As shown in the above experiments, we have made great progress in stereo matching in foggy scenes. However, our method also has limitations, e.g., the assumption over atmospheric parameters. We assume atmospheric parameters to be global constant for the current stage. This assumption is not fully applicable to scenes with inhomogeneous scattering median and multi-light sources. Besides, although we only focus on foggy scenes in this paper, our idea can be flexibly extended to other scattering media by adjusting the physical model, such as haze, rain, water, e.t.c.

5. Conclusion

In this paper, we have proved that fog contains depth hints beneficial for stereo matching in foggy scenes. We presented the fog volume representation to collect these depth hints. Our fog volume representation explored depth hints of fog by reversing the atmospheric scattering process and validated each disparity candidate used for the cost volume. By fusing our fog volume with the cost volume, the explored depth hints can help the cost volume to rectify the ambiguous matching caused by fog. Experiments proved that our fog volume can stabilize the learning and improve the disparity estimation in foggy scenes without sacrificing the performance in clear scenes.

References

[1] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision (IJCV)*, 35(3):269–293, 1999. 2

[2] Laurent Caraffa and Jean-Philippe Tarel. Stereo reconstruction and contrast restoration in daytime fog. In *Asian Conference on Computer Vision (ACCV)*, pages 13–25. Springer, 2012. 2

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018. 1, 2, 5, 6

[4] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2524–2534, 2020. 1, 2

[5] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:22158–22169, 2020. 5

[6] Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, pages 99–107, 2002. 2

[7] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4384–4393, 2019. 1, 2, 4, 5, 6, 7

[8] Raanan Fattal. Single image dehazing. *ACM Transactions on Graphics (TOG)*, 27(3):1–9, 2008. 5

[9] Ignacio Garcia-Dorado, Daniel G Aliaga, Saiprasanth Bhalachandran, Paul Schmid, and Dev Niyogi. Fast weather simulation for inverse procedural design of 3d urban models. *ACM Transactions on Graphics (TOG)*, 36(2):1–19, 2017. 2

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 5

[11] Tobias Gruber, Mario Bijelic, Felix Heide, Werner Ritter, and Klaus Dietmayer. Pixel-accurate depth evaluation in realistic driving scenarios. In *International Conference on 3D Vision (3DV)*, 2019. 2, 5, 6

[12] Tobias Gruber, Frank Julca-Aguilar, Mario Bijelic, and Felix Heide. Gated2depth: Real-time dense lidar from gated images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1506–1516, 2019. 1, 2, 6

[13] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504, 2020. 2

[14] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 807–814. IEEE, 2005. 6

[15] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):328–341, 2008. 1, 2

[16] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 66–75, 2017. 1, 2

[17] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515. IEEE, 2001. 2

[18] Maria Kolos, Artem Sevastopolsky, and Victor Lempitsky. Transpr: Transparency ray-accumulating neural 3d scene point renderer. In *International Conference on 3D Vision (3DV)*, pages 1167–1175. IEEE, 2020. 2

[19] Zhuwen Li, Ping Tan, Robby T Tan, Danping Zou, Steven Zhiying Zhou, and Loong-Fah Cheong. Simultaneous video defogging and stereo reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4988–4997, 2015. 2

[20] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[21] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979. 2

[22] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 5

[23] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 5

[24] Shree K Nayar and Srinivasa G Narasimhan. Vision in bad weather. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 820–827. IEEE, 1999. 1, 3, 4

[25] Shahriar Negahdaripour and Amin Sarafraz. Improved stereo matching in scattering media by incorporating a backscatter cue. *IEEE Transactions on Image Processing (TIP)*, 23(12):5743–5755, 2014. 2

[26] Jing Nie, Yanwei Pang, Jin Xie, Jing Pan, and Jungong Han. Stereo refinement dehazing network. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021. 1, 2

[27] Yanwei Pang, Jing Nie, Jin Xie, Jungong Han, and Xuelong Li. Bidnet: Binocular image dehazing without explicit disparity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5931–5940, 2020. 1, 2

[28] Min-Gyu Park and Kuk-Jin Yoon. Leveraging stereo matching with learning-based confidence measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 101–109, 2015. 2

[29] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47(1-3):7–42, 2002. 2

[30] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13906–13915, 2021. 2

[31] Taeyong Song, Youngjung Kim, Changjae Oh, Hyunsung Jang, Namkoo Ha, and Kwanghoon Sohn. Simultaneous deep stereo matching and dehazing with feature attention. *International Journal of Computer Vision (IJCV)*, pages 1–19, 2020. 2, 5, 6, 7

[32] Taeyong Song, Youngjung Kim, Changjae Oh, and Kwanghoon Sohn. Deep network for simultaneous stereo matching and dehazing. In *British Machine Vision Conference (BMVC)*, page 5, 2018. 5, 6

[33] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(7):787–800, 2003. 2

[34] Robby T Tan. Visibility in bad weather from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 1, 3, 4

[35] Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr. Learning to adapt for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9661–9670, 2019. 2

[36] Wayne Treible, Philip Saponaro, Scott Sorensen, Abhishek Kolagunda, Michael O'Neal, Brian Phelan, Kelly Sherbondy, and Chandra Kambhamettu. Cats: A color and thermal stereo benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017. 1, 2

[37] Charles Wheatstone. Xviii. contributions to the physiology of vision.—part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical transactions of the Royal Society of London*, (128):371–394, 1838. 2

[38] Bartlomiej Wronski. Volumetric fog: Unified compute shader-based solution to atmospheric scattering. In *ACM SIGGRAPH*, 2014. 1, 2, 3

[39] Chengtang Yao, Yunde Jia, Huijun Di, Pengxiang Li, and Yuwei Wu. A decomposition model for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6091–6100, 2021. 2

[40] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):650–656, 2006. 2

[41] Zhuoran Zheng, Wenqi Ren, Xiaochun Cao, Xiaobin Hu, Tao Wang, Fenglong Song, and Xiuyi Jia. Ultra-high-definition image dehazing via multi-guided bilateral learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16185–16194, 2021. 5, 6, 7