ScatterNeRF: 
Seeing Through Fog with Physically-Based Inverse Neural Rendering

Andrea Ramazzina\textsuperscript{1} Mario Bijelic\textsuperscript{2} Stefanie Walz\textsuperscript{1} Alessandro Sanvito\textsuperscript{1} Dominik Scheuble\textsuperscript{1} Felix Heide\textsuperscript{2} 
\textsuperscript{1}Mercedes-Benz \textsuperscript{2}Princeton University

Abstract

Vision in adverse weather conditions, whether it be snow, rain, or fog is challenging. In these scenarios, scattering and attenuation severely degrades image quality. Handling such inclement weather conditions, however, is essential to operate autonomous vehicles, drones and robotic applications where human performance is impeded the most. A large body of work explores removing weather-induced image degradations with dehazing methods. Most methods rely on single images as input and struggle to generalize from synthetic fully-supervised training approaches or to generate high fidelity results from unpaired real-world datasets. With data as bottleneck and most of today's training data relying on good weather conditions with inclement weather as outlier, we rely on an inverse rendering approach to reconstruct the scene content. We introduce ScatterNeRF, a neural rendering method which adequately renders foggy scenes and decomposes the fog-free background from the participating media – exploiting the multiple views from a short automotive sequence without the need for a large training data corpus. Instead, the rendering approach is optimized on the multi-view scene itself, which can be typically captured by an autonomous vehicle, robot or drone during operation. Specifically, we propose a disentangled representation for the scattering volume and the scene objects, and learn the scene reconstruction with physics-inspired losses. We validate our method by capturing multi-view In-the-Wild data and controlled captures in a large-scale fog chamber. Our code and datasets are available at https://light.princeton.edu/scatternerf.

1. Introduction

Imaging and scene understanding in the presence of scattering media, such as fog, smog, light rain and snow, is an open challenge for computer vision and photography. As rare out-of-distribution events that occur based on geography and region \cite{8}, these weather phenomena can drastically reduce the image quality of the captured intensity images, reducing local contrast, color reproduction, and image resolution \cite{8}. A large body of existing work has investigated methods for dehazing \cite{57, 5, 49, 29, 73, 77} with the most successful methods employing learned feed-forward models \cite{57, 5, 49, 29, 73}. Some methods \cite{49, 5, 35} use synthetic data and full supervision, but struggle to overcome the domain gap between simulation and real world. Acquiring paired data in real world conditions is challenging and existing methods either learn natural image priors from large unpaired datasets \cite{74, 73}, or they rely on cross-modal semi-supervision to learn to separate atmospheric effects from clear RGB intensity \cite{57}. Unfortunately, as the semi-supervised training cues are weak compared to paired supervised data, these methods often fail to completely separate atmospheric scatter from clear image content, especially at long distances. The problem of predicting clear images in the presence of haze is an open challenge, and notably harsh weather also results in severely impaired human vision – a major driver behind fatal automotive accidents \cite{4}.

As the distribution of natural scenes with participating media is long-tailed in typical training datasets \cite{14, 60, 19, 18, 8}, this also makes training and evaluation of computer vision tasks that operate on RGB streams in bad weather challenging. For supervised approaches to scene understanding tasks, these “edge” scenarios often directly re-
result in failure cases, including detection [8], depth estimation [20], and segmentation [52]. To tackle weather-based dataset bias, existing methods have proposed augmentation approaches that either simulate atmospheric weather effects on clear images [52, 65] or they employ fully synthetic simulation to generate physically-based adverse weather scenarios [13, 23, 52]. Unfortunately, both directions cannot compete with supervised training data, either due to the domain gap between real and synthetic data, or, as a result of an approximate physical forward model [65]. As such, the capability of both physically accurate modeling and separating scattering in participating media is essential for imaging and scene understanding tasks.

In this work, we depart from both feed-forward dehazing methods and fully synthetic training data, and we address this challenge as an inverse rendering problem. Instead of predicting clean images from RGB frames, we propose to learn a neural scene representation that explains foggy images with a physically accurate forward rendering process. Once this representation is fitted to a scene, this allows us to render novel views with real-world physics-based scattering, disentangle appearance and geometry without scattering (i.e., reconstruct the dehazed scene), and estimate physical scattering parameters accurately. To be able to optimize a scene representation efficiently, we build on the large body of neural radiance field methods [45, 75, 69, 70, 72, 47, 32, 9, 51, 58, 10, 76, 7, 66]. While existing NeRF methods assume peaky ray termination distributions and free-space propagation, we propose a forward model that can accurately model participating media. As an inductive bias, the scene representation, by design, separates learning the clear 3D scene and the participating media. We validate that the proposed method accurately models foggy scenes in real-world and controlled scenes, and we demonstrate that the disentangled scene intensity and depth outperform existing dehazing and depth estimation methods in diverse driving scenes.

Specifically, we make the following contributions:

- We propose a method to learn a disentangled representation of the participating media by introducing the Koschmieder scattering model into the volume rendering process.
- Our approach adds a single MLP used to model the scattering media proprieties and does not require any additional sampling or other procedures, making it a lightweight framework in terms of both computation and memory consumption.
- We validate that our method learns a physics-based representation of the scene and enables control over its hazed appearance. We confirm this using real data captured in both controlled and In-the-Wild settings.

2. Related Work

Imaging and Vision in Participating Media. As real world data in participating media is challenging to capture [8, 36, 2, 1], a large body of work introduces simulation techniques for snowfall [43], rainfall [24, 23], raindrops on the windshield [68], blur [33], night [53] and fog [57, 52, 39, 17]. Using this data, existing methods investigate pre-processing approaches using dehazing [57, 5, 49, 29, 73, 77, 25, 35], deraining [27, 16] and desnowing [43]. Early works on image dehazing as [25, 64] explore image statistics and physical models to estimate the airlight introduced by the fog scattering volume and invert the Koschmieder fog model. Later, CNN approaches [35] and [29] learn to predict the airlight and transmission, with the same goal of inverting the Koschmieder model. However, this disjoint optimization can lead to error accumulation. Hence, [49, 5, 57, 73] model the fog removal through a neural network learned end-to-end. As such, existing methods differ substantially in network structure and learning methodology. For example, some methods rely on GAN architectures [49, 57], transformer-based backbones [22] or encoder-decoder structures [5]. For model-learning, the approaches apply semi-/self-supervised [73, 57], fully supervised [49, 5] and test-time optimization techniques [42, 29]. All of these methods have in common that they do not explore multiple views to reconstruct a clear image. To overcome this methodological limitation we introduce a novel multi-view dataset in hazy conditions in Sec. 4 and explore reconstruction from multiple views for optimal image reconstruction through neural rendering approaches. Most similar to our method are the approaches from Sethuraman et al. [56] and from Levy et al. [34] that reconstruct scenes in underwater conditions which requires tackling strong color aberrations and specular reflections specific to the underwater domain.

Neural Rendering Methods for Large Scale Scenes A rapidly growing body of work is capable of representing unbounded outdoor scenes, such as the ones tackled in this work, with rich detail both in close and far range. NeRF++ [76] achieves this by using two NeRFs, one to model the foreground and one for the background scene. DONeRF [46] warps the space by radial distortion to bring the background closer. Recently, [7] has tackled this problem by proposing a non-linear parametrization technique suitable for the mip-NeRF algorithm [6]. For very large-scale scenes, too big to be fitted by a single NeRF, several works [62] [66] have explored the idea of learning multiple NeRFs for subparts of the scene.

Clear Scene Priors Given the under-constrained nature of scene reconstruction from a sparse set of views, novel views rendered by NeRFs are frequently afflicted by floating artifacts [7, 12] or not able to properly generalize to novel...
views and hence fail to render images from unseen poses correctly [30, 28]. To tackle this issue, several works have recently proposed the introduction of different regularization techniques. In [30], an entropy-based loss is introduced in order to enforce a sparsity constraint over the scene. Analogously, recent models exploiting an estimated depth as training cue [12, 50] implicitly enforce a sparsity constrain of the ray termination distributions by penalizing non-zero probabilities lying far from the prior estimated depth. Mip-NeRF360 [7] relies on a regularization technique aimed at encouraging a unimodal peaky distribution for the termination probability of a ray. While such methods work well for clear scenes, they are based on the assumption that most of the scene density is null, except for where there are solid objects. As such prior is not applicable in the presence of a participating media in the air, such approaches are not suitable for scenes with scattering media.

We propose to learn a separate representation of the clear scene and the participating media. This allows us to potentially use any of the above-mentioned regularization technique on the clear scene model without compromising on the haze scene reconstruction.

3. Disentangled Scattering Neural Radiance Fields

ScatterNeRF has five integral parts, namely the underlying physical model in Sec. 3.1, the formulation of the neural radiance field in Sec. 3.2, the formulation of the loss functions in Sec. 3.3, the details on ray sampling in Sec. 3.4 and implementation in Sec. 3.5. We describe these components in the following.

3.1. Physical Scattering Model

Large scattering volumes can be approximated by the Koschmieder model [31]. For each pixel, we model the attenuation representing the lost intensity due to scattered direct light of an object and the contribution of the airlight \(c_p\) caused by ambient light scattered towards the observer,

\[
C_F = lC_c + (1 - l)C_p,
\]

with \(l\) being the transmission, \(C_F\) corresponding to the observed pixel value and \(C_c\) the clear scene. The transmission \(l\) can be computed from the attenuation coefficient \(\sigma_p\) and the depth \(D\),

\[
l = \exp(-\sigma_p D).
\]

In existing methods [52] both parameters \(\sigma_p\) and airlight \(c_p\) are globally constant. Koschmieder’s model is equivalent to a volume rendering NeRF [45], Eq. (3), with scattering density \(\sigma_p\) and airlight \(C_p\) set constant, and ray integration from \(t_n\) to a maximum distance \(t_f\). For simplicity of notation, we omit the viewing direction \(d\). Starting with the forward model

\[
C_F(r) = \int_{t_n}^{t_f} T_F(t)\sigma_F(r(t))c_F(r(t))dt,
\]

and assuming two disjoint additive volume densities \(\sigma_c\) for the scene and \(\sigma_p\) for the scattering media (\(\sigma_F = \sigma_p + \sigma_c\)), the volume rendering equation can be formulated as

\[
C_F(r) = l(r)C_c(r) + (1 - l(r))C_p,
\]

where the integrated object color \(C_c\) is defined with emitted color \(c_e(r)\) at position \(r\),

\[
C_c(r) = \int_{D}^{t_f} T_c(t)\sigma_c(r(t))c_e(r(t))dt.
\]

The transmission then is dependent on \(r\) leading to \(\exp(-\sigma_p D)\). Here, the subscript denotes if the clear scene object \(c\) or participating media \(p\) is modeled. Finally, the value of the foggy pixel \(C_F(r)\) can be estimated by integrating \(c_F(r(t))\) along one camera ray vector \(r\). We denote \(r_i = [x_i; d_i] \in \mathbb{R}^5\) consisting of the 3D position \(x\) and direction \(d\).

Next, we can relax the constraints on \(\sigma_p\) and \(c_p\), and allow them to approximate arbitrary values. This results in,

\[
C_F(r) = \int_{t_n}^{t_f} T_F(t)\sigma_p(r(t))c_p(r(t))
+ \sigma_c(r(t))c_e(r(t)))dt,
\]

with,

\[
T_F(t) = \exp\left(-\int_{t_n}^{t}(\sigma_c(r(s)) + \sigma_p(r(s)))ds\right),
\]

\[
T_F(t) = T_p(t)T_c(t),
\]
which can be expressed as

\[
C_F(r) = \frac{\int_{t_n}^{t_f} T_c(t) \left( T_p(t) \sigma_p(r(t)) c_p(r(t)) \right) dt}{\int_{t_n}^{t_f} T_p(t) \left( T_c(t) \sigma_c(r(t)) c_c(r(t)) \right) dt} + T_p(t) \left( T_c(t) \sigma_c(r(t)) c_c(r(t)) \right) dt.
\]

(9)

**3.2. Neural Radiance Model**

Further simplifications of Eq. (3) can be found by solving the integral through numerical quadrature needed for the discrete neural forward network. The numerical quadrature leads to,

\[
C_F(r) = \sum_{i} w_F(r(t_i)) c_F(r(t_i)).
\]

(10)

\[
w_F(r_i) = T_F(r_i)(1 - \exp (\sigma_p(r_i) + \sigma_c(r_i)) \delta_i),
\]

(11)

\[
T_F(r_i) = \exp \left( - \sum_{j=1}^{i-1} (\sigma_p(r_i) + \sigma_c(r_i)) \delta_i \right),
\]

(12)

\[
\delta_i = t_{i+1} - t_i, \text{ and }
\]

\[
c_F(r_i, d) = \frac{\sigma_c(r_i) c_c(r_i) + \sigma_p(r_i) c_p(r_i)}{\sigma_p(r_i) + \sigma_c(r_i)}.
\]

(13)

(14)

To facilitate the learning of two independent volume representations, we model each part independently by one NeRF. The parameters \(\sigma_i, c_i\) for \(i \in \{c, p\}\) are predicted by a multi-layer perceptron (MLP) as,

\[
c_i, \sigma_i = f_i(\gamma(x)).
\]

(15)

For the clear scene NeRF we adopt a similar strategy as in [45] and optimize simultaneously two MLPs, \(f_{c, coarse}\) and \(f_{c, fine}\) using the same loss formulation but different sampling procedure, as detailed in Sec. 3.3 and Sec. 3.4. The coordinates are encoded by the function \(\gamma\) following [45, 63]. This learning disentanglement of the scene and scatter representation allows us to render scenes with different fog densities by scaling \(\sigma_p\) or even dehaze the image entirely by setting \(\sigma_p = 0\).

For the dehazing task the image can be rerendered with \(\sigma_p = 0\) and the formulation reduces to the scene model only leading to,

\[
C_F(r) = C_c(r) = \sum_{i} w_c(r(t_i)) c_c(r(t_i)).
\]

(16)

\[
w_c(r(t_i)) = T_c(t_i)(1 - \exp (\sigma_c(r(t_i)) \delta_i),
\]

(17)

\[
T_c(t_i) = \exp \left( - \sum_{j=1}^{i-1} (\sigma_c(r(t_j)) \delta_i \right).
\]

(18)

### 3.3. Training Supervision

To learn the neural forward model we supervise the reconstructed images with a pixel loss between predicted and training frames \(L_{rgb}\), align the observable airlight \(L_A\), supervise the scene depth with \(L_D\), enforce discrete clear scene volumetric density \(L_{cc}\) and enforce the scattering density to be disjoint from the scene objects in \(L_{cf}\). In the following, network predictions are marked with a hat and ground-truth values are marked with a bar.

**Color Supervision** The image loss between ground-truth \(C_F\) and reconstructed haze images \(\hat{C}_F\) is used for direct supervision as

\[
L_{rgb} = \mathbb{E}_r \left[ ||\hat{C}_F(r) - \hat{C}_F(r)||_2^2 \right].
\]

(19)

**Airlight Color Supervision** As discussed in Sec. 3.1, the model separation allows us to supervise the airlight directly. We minimize the variance of the predicted \(\bar{c}_p\) with respect to a ground-truth airlight \(\bar{c}_p\) estimated following [64]. The target airlight \(\bar{c}_p\) is computed as follows,

\[
\bar{c}_p(r) = \frac{z^2(r)^2 + \lambda \bar{c}_p^0(r)}{z^2(r) + \lambda}.
\]

(20)

Here \(I_F\) is the relative luminance of the hazed image estimated as \(I_F(r) = \xi \cdot \log(C_F(r))\), that is a linear combination of the linearized RGB values [3] obtained by decomposing the color image by applying \(\log(C_F)\) [61]. \(c_p^0\) is an initial global constant airlight estimate computed with the dark channel prior following [25], \(z = 1/(l - 1)\) and \(\lambda\) is a weighting factor. The total loss can be written as,

\[
L_A = \mathbb{E}_r \left[ ||\bar{c}_p(r) - \bar{c}_p(r)||_2^2 \right].
\]

(21)

**Clear Scene Entropy Minimization** To regularize \(f_c\), we follow [12] according to which the rays cast in the scene have peaky unimodal ray termination distributions. Thus, we add a loss to minimize its distribution entropy,

\[
L_{cc} = \mathbb{E}_r \left[ -\sum_i \bar{w}_c(r) \cdot \log (\bar{w}_c(r)) \right].
\]

(22)

**Foggy Scene Entropy Maximization** Analogously to Eq. (22), we regularize \(f_p\). Thereby, \(\bar{\sigma}_p\) is fitted in semi-supervised fashion during the optimization. This allows us to model fog inhomogeneities, for example close to hot surfaces. To achieve this goal we apply an entropy-based loss which allows the network \(f_p\) to learn a spatially-varying media density. Based on the assumption of almost-uniformity
where \( \tilde{\sigma} \) distribute the fog volume density density \( \tilde{\sigma} \). The entropy maximization relies on the scene volume presented by the fog volume density for extended fog volume, we enforce that this has to be re-presented environment dataset. The figure demonstrates the Figure 3: Examples of our proposed In-the-Wild and con-
trolled environment dataset. The figure demonstrates the diversity of the scenes and the fog densities.

for extended fog volume, we enforce that this has to be represented by the fog volume density \( \tilde{\sigma}_F \) through maximizing the entropy as follows,

\[
L_{cF} = \mathbb{E}_r \left[ \sum_i \tilde{\sigma}_F(r) \cdot \log(\tilde{\sigma}_F(r)) \right],
\]

where \( \tilde{\sigma}_F = \sum_i \tilde{\sigma}_F(r) \) and \( \tilde{\sigma}_F = 1 - \exp((\tilde{\sigma}_p + \tilde{\sigma}_c)\delta_i) \). The entropy maximization relies on the scene volume density \( \tilde{\sigma}_c \) to disentangle both distributions and not only distribute the fog volume density \( \tilde{\sigma}_F \) throughout the scene. We minimize this loss only for \( \tilde{\sigma}_p \), not \( \tilde{\sigma}_c \).

**Estimated Depth Supervision** To supervise the scene representation further we supervise the scene depth \( \hat{D}_c \) similar to [50], through the depth \( \hat{D} \) estimated from the stereo sensor setup. Thereby, the depth can be estimated as follows,

\[
\hat{D}_c(r) = \sum_i \hat{w}_c(r(t))t_i
\]

which leads to,

\[
L_{depth} = \mathbb{E}_r \left[ ||\hat{D}_c(r) - D(r)||_2^2 \right].
\]

We use a pretrained RGB stereo algorithm [37] to compute \( \hat{D} \).

**Total Training Loss** Combining the five losses, we obtain the following loss formulation,

\[
L_{tot} = \psi_1 L_{rgbF} + \psi_2 L_A + \psi_3 L_{ec} + \psi_4 L_{cF} + \psi_5 L_{depth},
\]

where \( \psi_1...5 \) are the loss weights, provided in the Supplementary Material.

**3.4. Sampling**

We follow the hierarchical volume sampling strategy proposed in [45] to sample the query points for the radiance field networks. However, instead of sampling across the whole volume density we adapt the approach to our decomposed volume densities \( \sigma_c \) and \( \sigma_F \). As \( \sigma_F \) is regularized to be approximately constant across the scene, the resampling procedure is not going to be performed using the scene weights \( W_F = T_F(1 - \exp(\sigma\delta)) \) but rather using the clear scene \( w_c = T_c(1 - \exp(\sigma\delta)) \). We apply this approach to follow an importance sampling and reconstruct scene objects by sampling close to object boundaries.

**3.5. Implementation Details**

We train for 250’000 steps and a batch size of 4096 rays. As optimizer we use ADAMW [44] with \( \beta_1 = 0.9 \), \( \beta_2 = 0.999 \), learning rate \( 5 \cdot 10^{-4} \) and weight decay factor of \( 10^{-2} \). Our code implementation is based on Pytorch [48] and we train on four NVIDIA RTX A6000. The NeRF MLPs \( f_{coarse} \) and \( f_{fine} \) follow Mildenhall et al. [45], while we use fewer hidden layers for \( f_p \). The network architecture and other hyper-parameters are listed in the Supplementary Material.

**4. Dataset**

To evaluate the proposed method, we collect both an automotive In-the-Wild foggy dataset and a controlled fog dataset. Example captures illustrating the dataset are shown in Fig. 3. In total, we collect 2678 In-the-Wild foggy images throughout nine different scenarios. The sensor set for the In-the-Wild dataset consists of an Aptina-0230 stereo camera pair and a Velodyne HDL64S3D laser scanner. Camera poses are estimated with the hierarchical localization approach [54, 55], a structure from motion pipeline optimized for robustness to changing conditions. The training and testing split was done by randomly choosing 30% of the images as test set. Each scene contains between 150 and 300 images. Ground-truth depth data is estimated with the stereo-camera method described in [38]. The controlled fog dataset is captured in a fog chamber where fog with varying visibilities can be generated. We capture 903 images in a large-scale fog chamber with clear ground-truth and two different fog densities. Further ground-truth depths are captured through a Leica ScanStation P30 laser scanner (360°/290° FOV, 1550 nm, with up to 1M points per second, up to 8” angular accuracy, and 1.2mm + 10 parts per million (ppm) range accuracy). Each point cloud consists of approximately 157M points and we accumulate multiple point clouds from different positions to reduce occlusions and increase resolution.

**5. Assessment**

Next, we validate the proposed method by ablating the different components, confirming their effectiveness, assessing the quality of scene reconstructions in foggy scenes, the decomposition into dehazed scene content, and the scene depth reconstruction.
### Table 1: Quantitative comparison of the proposed ScatterNeRF and state-of-the-art methods on In-the-Wild sequences. Best results in each category are in **bold** and second best are underlined. Last column in the second row presents the average over all sequences.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug-NeRF [11]</td>
<td>0.248</td>
<td>41.29</td>
<td>0.979</td>
<td>0.242</td>
<td>41.75</td>
<td>0.981</td>
<td>0.302</td>
<td>38.99</td>
<td>0.964</td>
<td>0.286</td>
<td>40.74</td>
<td>0.978</td>
<td>0.274</td>
<td>40.32</td>
<td>0.973</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DS-NeRF [12]</td>
<td>0.273</td>
<td>42.59</td>
<td>0.977</td>
<td>0.252</td>
<td>43.22</td>
<td>0.982</td>
<td>0.318</td>
<td>40.94</td>
<td>0.964</td>
<td>0.29</td>
<td>43.6</td>
<td>0.98</td>
<td>0.284</td>
<td>42.16</td>
<td>0.975</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVGO [59]</td>
<td>0.376</td>
<td>38.51</td>
<td>0.947</td>
<td>0.349</td>
<td>40.14</td>
<td>0.966</td>
<td>0.376</td>
<td>38.92</td>
<td>0.95</td>
<td>0.368</td>
<td>40.55</td>
<td>0.968</td>
<td>0.384</td>
<td>38.81</td>
<td>0.951</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mip-NeRF [6]</td>
<td>0.272</td>
<td>41.88</td>
<td>0.976</td>
<td>0.25</td>
<td>42.83</td>
<td>0.981</td>
<td>0.321</td>
<td>40.2</td>
<td>0.963</td>
<td>0.291</td>
<td>43.08</td>
<td>0.98</td>
<td>0.292</td>
<td>41.5</td>
<td>0.973</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mip-NeRF360 [7]</td>
<td>0.245</td>
<td>42.37</td>
<td>0.978</td>
<td>0.225</td>
<td>42.55</td>
<td>0.981</td>
<td>0.268</td>
<td>41.53</td>
<td>0.969</td>
<td>0.241</td>
<td>42.97</td>
<td>0.981</td>
<td>0.243</td>
<td>41.99</td>
<td>0.976</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeRF [45]</td>
<td>0.287</td>
<td>41.84</td>
<td>0.975</td>
<td>0.256</td>
<td>43.67</td>
<td>0.992</td>
<td>0.352</td>
<td>40.36</td>
<td>0.962</td>
<td>0.305</td>
<td>43.39</td>
<td>0.979</td>
<td>0.293</td>
<td>42.39</td>
<td>0.976</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeRF++ [76]</td>
<td>0.292</td>
<td>42.38</td>
<td>0.975</td>
<td>0.267</td>
<td>43.62</td>
<td>0.98</td>
<td>0.336</td>
<td>40.69</td>
<td>0.962</td>
<td>0.312</td>
<td>43.86</td>
<td>0.978</td>
<td>0.306</td>
<td>41.96</td>
<td>0.972</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plenoxels [15]</td>
<td>0.481</td>
<td>29.85</td>
<td>0.933</td>
<td>0.456</td>
<td>29.72</td>
<td>0.943</td>
<td>0.497</td>
<td>29.49</td>
<td>0.922</td>
<td>0.473</td>
<td>29.47</td>
<td>0.938</td>
<td>0.482</td>
<td>29.92</td>
<td>0.931</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref-NeRF [67]</td>
<td>0.382</td>
<td>39.61</td>
<td>0.962</td>
<td>0.376</td>
<td>40.81</td>
<td>0.967</td>
<td>0.365</td>
<td>40.04</td>
<td>0.96</td>
<td>0.339</td>
<td>42.81</td>
<td>0.976</td>
<td>0.369</td>
<td>40.21</td>
<td>0.964</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ScatterNeRF</td>
<td>0.231</td>
<td>44.45</td>
<td>0.981</td>
<td>0.228</td>
<td>45.27</td>
<td>0.983</td>
<td>0.255</td>
<td>42.96</td>
<td>0.973</td>
<td>0.265</td>
<td>44.64</td>
<td>0.982</td>
<td>0.247</td>
<td>43.88</td>
<td>0.978</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Quantitative comparison of the proposed ScatterNeRF and state-of-the-art methods on controlled scenes. Best results in each category are in **bold** and second best are underlined.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug-NeRF [11]</td>
<td>0.521</td>
<td>29.81</td>
<td>0.934</td>
<td>0.48</td>
<td>25.08</td>
<td>0.822</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DS-NeRF [12]</td>
<td>0.535</td>
<td>29.52</td>
<td>0.926</td>
<td>0.502</td>
<td>26.71</td>
<td>0.853</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVGO [59]</td>
<td>0.613</td>
<td>22.65</td>
<td>0.801</td>
<td>0.607</td>
<td>22.22</td>
<td>0.847</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mip-NeRF [6]</td>
<td>0.513</td>
<td>30.42</td>
<td>0.937</td>
<td>0.611</td>
<td>24.88</td>
<td>0.87</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mip-NeRF360 [7]</td>
<td>0.53</td>
<td>30.59</td>
<td>0.938</td>
<td>0.623</td>
<td>26.59</td>
<td>0.878</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeRF [45]</td>
<td>0.505</td>
<td>29.81</td>
<td>0.937</td>
<td>0.614</td>
<td>24.97</td>
<td>0.877</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeRF++ [76]</td>
<td>0.507</td>
<td>30.0</td>
<td>0.937</td>
<td>0.611</td>
<td>25.47</td>
<td>0.879</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plenoxels [15]</td>
<td>0.582</td>
<td>20.48</td>
<td>0.872</td>
<td>0.607</td>
<td>22.07</td>
<td>0.852</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref-NeRF [67]</td>
<td>0.498</td>
<td>29.11</td>
<td>0.933</td>
<td>0.616</td>
<td>24.7</td>
<td>0.864</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ScatterNeRF</td>
<td>0.505</td>
<td>30.82</td>
<td>0.938</td>
<td>0.509</td>
<td>27.42</td>
<td>0.878</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3: Ablation study of the ScatterNeRF contribution for a subset of the In-the-Wild dataset.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>LPIPS</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeRF</td>
<td>0.278</td>
<td>39.06</td>
<td>0.964</td>
</tr>
<tr>
<td>NeRF + Koschmieder</td>
<td>0.281</td>
<td>39.54</td>
<td>0.960</td>
</tr>
<tr>
<td>NeRF + Depth [50]</td>
<td>0.290</td>
<td>40.03</td>
<td>0.972</td>
</tr>
<tr>
<td>ScatterNeRF w/o cleared sampling</td>
<td>0.219</td>
<td>40.70</td>
<td>0.975</td>
</tr>
<tr>
<td>ScatterNeRF</td>
<td>0.22</td>
<td>41.45</td>
<td>0.975</td>
</tr>
</tbody>
</table>

### 5.1 Ablation

In order to assess the role and contribution of the different components of our framework, we conduct an ablation study whose results are presented in Tab. 3. We consider as starting point the "NeRF" [45]. Its PSNR on the In-the-Wild subset dataset is 39.06 dB. By adding the depth supervision Eq. (25) and Eq. (22) the model’s PSNR improves by 0.97 dB. However, as shown in Fig. 2, a NeRF trained in such a way does not produce a physically accurate representation of the scene, as it does not model the participating media present in the scene, but rather represents it as a clear scene. Adding the scattering media $f_p$ in "ScatterNeRF w/o cleared sampling", together with its two losses $L_A$ and $L_F$ leads to a PSNR of 40.7 dB. Finally, adding the sampling strategy in the full "ScatterNeRF" helps the model achieve better results, with a PSNR of 41.45 dB, summing up to a
5.3. Generalizability

As our proposed method does not require major changes to the sampling procedure, rendering or scene representation architecture, it can be easily integrated with existing neural rendering methods. To demonstrate this, we extend NSG [47] to foggy scenes and integrate the decoupling of scene and scattering media. A qualitative comparison between the base model and the one enhanced with our framework is shown in Fig. 4, where it is possible to observe both the finer details at long distances in addition to the possibility to remove the scattering media and better reconstruct the vehicle leaving the scene. Quantitatively, this improvement is reflected in an improvement of scene reconstruction PSNR, increasing from 29.78 dB to 30.67 dB. This also validates that it is possible to employ our framework also in the presence of moving objects, which is a common occurrence in automotive scenes. We discuss in the supplementary material other examples of the integration of ScatterNeRF in state-of-the-art neural radiance fields to augment their scope.

5.4. Dehazing

We quantify the ability of ScatterNeRF to learn a disentangle representation between objects and scattering media, and hence to render the corresponding clear scene which is effectively dehazing the foggy image. To this end, we compare against the state-of-the-art dehazing methods PFF [5], D4 [73], EPND [49], ZeroScatter [57], ZeroRestore [29] and MAP-Net [71]. For the In-the-Wild scenes, we rely on the FiD score [26] to evaluate the quality of the dehazing. As ground-truth data is unavailable, this score allows us to compare the dehazed sequences with a similar clear-weather scene collected with the automotive setup described in Sec. 4. For the controlled environment dataset the PSNR can directly be evaluated by warping a nearby clear-weather image to the respective foggy image. In contrast to other dehazing methods operating on single frames, ScatterNeRF learns a consistent representation of the entire scene. As a result, the dehazing is consistent across consecutive frames, whereas the baseline methods often are affected by flickering effects. This is also reflected in the quantitative results in Tab. 4. Here, ScatterNeRF outperforms the baseline methods on almost all sequences indicating a consistent dehazing across the entire sequence. Furthermore, the qualitative results in Fig. 5 from the controlled fog chamber dataset reveal another strength of ScatterNeRF: ScatterNeRF is the only method able to reconstruct the cart behind the vehicle as it can leverage information from the learned representation of the entire sequence. This results in a higher PSNR indicating an improved dehazing on the respective sequence. Additionally, as evident in Fig. 5, ScatterNeRF achieves visually enhanced contrast compared to existing dehazing methods.

Table 4: Quantitative dehazing comparison on In-the-Wild dataset with FiD score and on controlled dataset with PSNR. The best results in each category are in bold and the second best are underlined.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>ScatterNeRF</th>
<th>PFF</th>
<th>D4</th>
<th>EPND</th>
<th>ZeroScatter</th>
<th>ZeroRestore</th>
<th>MAP-Net</th>
<th>FI-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Farm</td>
<td>395.52</td>
<td>376.59</td>
<td>342.10</td>
<td>343.66</td>
<td>355.27</td>
<td>340.21</td>
<td>361.20</td>
<td></td>
</tr>
<tr>
<td>Intersection</td>
<td>389.37</td>
<td>387.09</td>
<td>355.63</td>
<td>355.27</td>
<td>397.71</td>
<td>366.82</td>
<td>366.04</td>
<td></td>
</tr>
<tr>
<td>Suburb</td>
<td>283.43</td>
<td>403.71</td>
<td>295.73</td>
<td>303.61</td>
<td>327.98</td>
<td>355.11</td>
<td>336.59</td>
<td></td>
</tr>
<tr>
<td>Speed Control</td>
<td>278.06</td>
<td>409.94</td>
<td>352.78</td>
<td>352.08</td>
<td>358.97</td>
<td>339.19</td>
<td>289.20</td>
<td></td>
</tr>
<tr>
<td>Road Fork</td>
<td>284.69</td>
<td>410.05</td>
<td>318.64</td>
<td>317.06</td>
<td>355.30</td>
<td>325.45</td>
<td>342.25</td>
<td></td>
</tr>
<tr>
<td>Addt</td>
<td>322.19</td>
<td>441.33</td>
<td>328.55</td>
<td>330.71</td>
<td>380.70</td>
<td>342.91</td>
<td>334.87</td>
<td></td>
</tr>
<tr>
<td>Construction</td>
<td>329.22</td>
<td>381.51</td>
<td>315.28</td>
<td>315.16</td>
<td>339.66</td>
<td>316.11</td>
<td>320.18</td>
<td></td>
</tr>
<tr>
<td>Countryside</td>
<td>205.06</td>
<td>435.51</td>
<td>320.09</td>
<td>321.15</td>
<td>337.48</td>
<td>321.29</td>
<td>343.70</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Qualitative comparison of dehazed images on real-world automotive measurements. The proposed ScatterNeRF enables enhanced contrast and visibility compared to state-of-the-art descattering methods.

6.12% PSNR increase over the baseline. We also analyze an additional model, here called "NeRF+Koschmieder", in which a Koschmieder model is added to the NeRF output, and we can note in Tab. 3 its limited performances due to its over-simplifications.

5.2. Foggy Scene Reconstruction

We compare our proposed method with NeRF [45], Mip-NeRF [6], DVGO [59], Plenoxel [15], Ref-NeRF [67], two methods for unbounded scenes [76, 7] and two NeRFs with auxiliary regularization [50, 11]. Qualitative results are presented in Fig. 6. The baseline methods struggle with object edges and fine structures, which our method is able to reconstruct for novel views. Our method is the only approach able to reconstruct the edges cleanly achieving to the highest PSNR scores for In-the-Wild captures and fog chamber captures. On average, the proposed method improves on average by 2.13 dB the PSNR of NeRF and by 1.64 dB to the next best model. For single sequences with high fog densities, improvements in the reconstruction of up to 2.39 dB are measurable. For the SSIM metric it outperforms all other approaches except on the light foggy scene dubbed "car accident" where it seconds [15].
5.5. Depth Estimation

Depth reconstruction in foggy conditions is unreliable and performs poorly for far objects due to the loss of scene information caused by the scattering medium [21]. This can be seen in Fig. 7 where the second pole and the building in the far background is not visible in the depth estimated using [37], due to the lack of contrast in stereo matching. On the other hand, our approach enforces consistency across multiple frames and therefore achieves accurate depth also for areas distant from the cameras. Quantitatively, we compare our results with state-of-the-art RGB depth-estimation methods, namely the monocular algorithms DepthFormer [40] and BinsFormer [41], as well the stereo algorithm CREStereo [37] with a 22cm baseline. Using the LiDAR pointcloud as groundtruth, the Mean Absolute Depth Prediction Error (MAE) for our method is 3.72m, while we get 5.84m for DepthFormer, 5.85m for BinsFormer and 4.73m for CREStereo. Our algorithm outperforms the baselines by more than 21% MAE, demonstrating the superiority of our approach for accurate depth reconstruction of foggy scenes.

6. Conclusion

We introduce ScatterNeRF, a neural rendering method that represents foggy scenes with disentangled representations of the participating media and scene. We model light transport in the presence of scattering with disentangled volume rendering, separately modeling the clear scene and fog, and introduce a set of physics-based losses designed to enforce the division between media and scene. Extensive experiments with both In-the-Wild and controlled scenario measurements validate the proposed approach. We demonstrate that ScatterNeRF is capable of rendering the learned scene without scattering media and can be hence used to alter or remove the haze from a sequence video, reaching quality comparable to state-of-the-art image dehazing algorithms – solely by fitting image observations without any forward neural network for dehazing or denoising.

Acknowledgments This work was supported by the AI-SEE project with funding from the FFG, BMBF, and NRC-IRA. We thank the Federal Ministry for Economic Affairs and Energy for support via the PEGASUS-family project “VVM-Verification and Validation Methods for Automated Vehicles Level 4 and 5”. Felix Heide was supported by an NSF CAREER Award (2047359), a Packard Foundation Fellowship, a Sloan Research Fellowship, a Sony Young Faculty Award, a Project X Innovation Award, and an Amazon Science Research Award.
References


[34] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz. Seathru-nerf: Neural radiance fields in scattering media, 2023. 2


[40] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation, 2022. 8

[41] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation, 2022. 8


