Learning To Generate Atmospheric Turbulent Images

Shyam Nandan Rai and C. V. Jawahar

IIIT Hyderabad
shyam.nandan@research.iiit.ac.in, jawahar@iiit.ac.in

Abstract. Modeling atmospheric turbulence is a challenging problem since the light rays arbitrarily bend before entering the camera. Such models are critical to extend computer vision solutions developed in the laboratory to real-world use cases. Simulating atmospheric turbulence by using statistical models or by computer graphics is often computationally expensive. To overcome this problem, we train a generative adversarial network which outputs an atmospheric turbulent image by utilizing less computational resources than traditional methods. We propose a novel loss function to efficiently learn the atmospheric turbulence at the finer level. Experiments show that by using the proposed loss function, our network outperforms the existing state-of-the-art image to image translation network in turbulent image generation. We also perform extensive ablation studies on the loss function to demonstrate the improvement in the perceptual quality of turbulent images.

Keywords: Generative Adversarial Network · Atmospheric Turbulence · Loss Function.

1 Introduction and Related Work

The performance of computer vision algorithms drastically decreases when deployed in varying weather conditions [12,13]. Especially in applications like autonomous navigation and aerial imaging, the atmospheric condition adversely affects the performance of the underlying vision algorithm. The problem of computer vision models adapting to changing weather could be solved by collecting data for all the weather conditions and training vision algorithms on them. But, collecting data for each weather condition would require a huge cost and time. Hence, we propose a deep learning-based approach, which particularly models the hot weather conditions among all weather conditions. We also show that the computational time taken to generate hot weather images by our method was less than the traditional methods. The phenomena of geometrical distortion caused by extremely hot weather are termed as atmospheric turbulence.

The primary cause of atmospheric turbulence is the heterogeneous nature of the atmosphere between the camera and the object. The heterogeneity in the medium is caused by the time-space varying changes in temperature, air pressure,

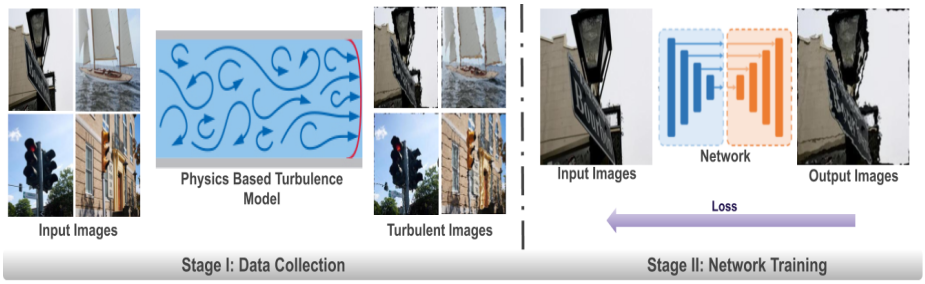

Fig. 1: Illustration of the overall pipeline to generate a turbulent image. *Stage I*: In this stage, we create a turbulent dataset using [16] method to train our network. *Stage II*: Training of our generative adversarial network using the created turbulent dataset. After training, turbulent images can be generated by simply passing the corresponding non-turbulent image through the network.

wind speed and humidity, which results in the introduction of geometrical distortion as well as a decrease in perceptual information of an imaged object. Several methods have been proposed to model the nature of atmospheric turbulence and render images with the help of these models. Turbulence modeling methods can be divided into two major approaches: Ray tracing in computer graphics [4,17] and image distortion simulation [7,16]. Initial approaches in computer graphics either used curved ray tracing [17] or solved physically-based differential equations [4,14] to estimate the parameters of atmospheric turbulence based on real turbulent images, which help in describing the trajectory of light. Another approach to model atmospheric turbulence is to statistically model the turbulence and distort images using those models. Earlier methods used light propagation through multiple phase screens [7] for this purpose. Later methods model turbulence as a simple Gaussian function [22] or derived a physics-based method [16] by which turbulent fields were generated efficiently. However, these methods are computationally expensive, which requires a large amount of computational time to generate a large dataset. Recently, Deep learning [8] has become a powerful framework to generate complex non-linear data such as images, speech, and videos. In particular, Generative Adversarial Networks(GANs) are widely used for the generation of images as it directly learns the empirical data distribution from the data samples. Application of GANs include images-to-image translation tasks such as super-resolution [9], style transfer [2], and synthetic data generation [18].

Leveraging the advantages of GANs, we train a deep adversarial network that takes an input image and gives out the corresponding turbulent image. While training, our GAN tries to learn the natural image distribution of the turbulent image. GANs provides the flexibility of sampling infinite samples from the learned distribution by simply feed-forwarding input samples into the network. Hence after training, using our trained GAN, we can generate large datasets of turbulent images at inference which would take few milliseconds to generate a turbulent image on an average GPU enabled device. Hence, we solve the problem of high computational issues encountered with traditional methods by using

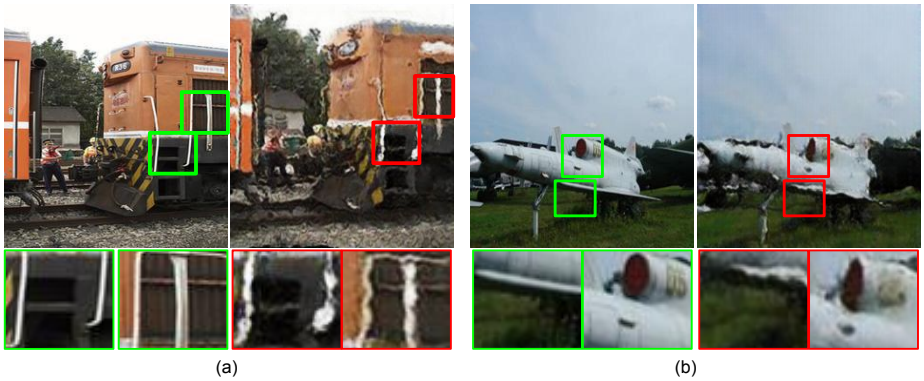

Fig. 2: (a) and (b) are the samples turbulent images generated by our network. Red boxes show the magnified patches of the input image and green boxes shows the turbulent patches.

GAN. Additionally, we introduce a novel loss function specifically intended to improve the quality of output turbulent images. To train the adversarial network, we require a substantial number of turbulent and non-turbulent image pairs. Since there was no public dataset available for training, we build a dataset by generating turbulent fields using [16], method and then we randomly placed the turbulent field onto the non-turbulent images to generate the corresponding turbulent image. Figure 1 illustrates the overall pipeline of our approach for generating turbulent images. We quantitatively and qualitatively show that the turbulent images generated by our network are close to natural turbulent images. Figure 2 shows the sample turbulent image generated by our model. These generated turbulent images can be further used as turbulent datasets in building restoration networks. Also, it increases the accuracy of various computer vision algorithms such as classification and semantic segmentation in a turbulent environment. The major contributions of this paper are:

1. We propose a deep generative network for the generation of turbulent images by taking lesser computational time than traditional methods. To the best of our knowledge, it is the first approach to generate turbulent images using deep learning.
2. We propose a novel loss function by which deep network efficiently learns to transfer atmospheric turbulence into the non-turbulent natural images. An extensive ablation study is reported to demonstrate the effectiveness of the loss function.
3. To train our network, we constructed a large scale dataset consisting of turbulent and non-turbulent image pairs.

2 Proposed Model

Generative models are widely used for generating samples from a data distribution. GAN is one of the models among all the generative models. Consider, two

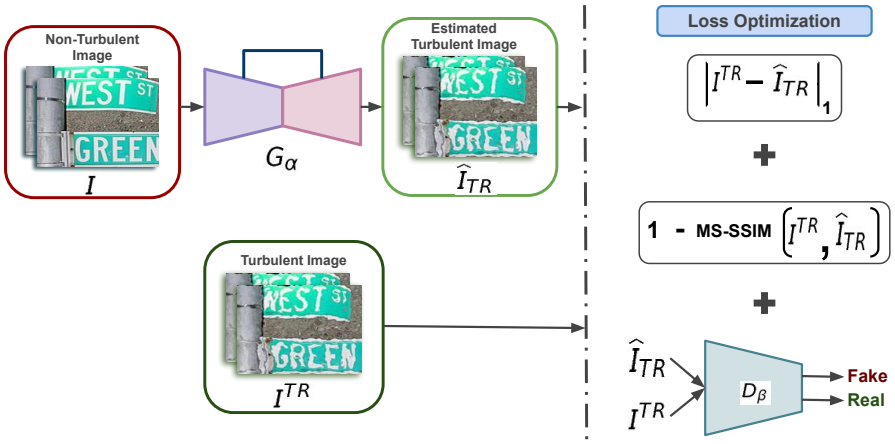

Fig. 3: Overview of our proposed generative adversarial network. While training the network, a non-turbulent image I is passed through generator G_α with learnable parameter α which gives the estimated turbulent image $\hat{I}^{TR}$. I^{TR} and $\hat{I}^{TR}$ are passed through loss optimization block to minimize the L1 loss, MS-SSIM loss and adversarial loss between them.

differential function generator G_α and D_β where α and β are learnable parameters. The generator creates fake samples that are intended to be closer to the training data distribution. The discriminator is a classifier where it examines whether the given samples belong to training data distribution or not. The generator tries to generate fake samples by adjusting its α to fool the discriminator. On the other hand, the discriminator learns to discriminate between real and fake samples. In this way the generator and discriminator their parameters α and β. This framework can be viewed as a mini-max game where the generator tries to minimize the probability of its samples to be fake whereas discriminator tries to maximize it. Formally, we assume G_α and D_β to be a deep neural network.

In our problem setting, we train a generative adversarial network that aims to estimate a turbulent image $\hat{I}^{TR}$ from a non-turbulent image I. Here, I^{TR} is the turbulent image corresponding to a non-turbulent image I. Our goal is to train a generator G_α with learnable parameter α, which can generate turbulent images by minimizing loss function and can fool the discriminator D_β. Here, D_β is a classification network which classifies between estimated non-turbulent turbulent and non-turbulent images with learnable parameter β. Figure 3 shows the overall outline of our proposed network architecture. In the subsequent subsections, we describe the architecture of our generative adversarial network along with the new loss function, which is particularly formulated to improve the quality of the turbulent image generation process.

Network Architecture: The generative adversarial network is divided into two networks: generator and discriminator. Our generator mainly follows the architecture of U-Net [15]. The architecture of our generator is divided into two paths: contracting path and expanding path. The contracting path downsamples the

input image into the feature space. It consists of four contracting blocks, where, each block contains two 4×4 convolutional layer of stride 2 and padding 1. Each convolutional layer is followed by a LeakyReLU layer and a batch normalization layer. After, each contracting block, we double the depth of the feature maps. For each contracting block, we have a corresponding expanding block in the expanding path. An expanding block contains two 4×4 transpose convolutional layer of stride 2 and padding 1. Each transpose convolutional layer is followed by a ReLU layer and a batch normalization layer. Each expanding block concatenates the cropped feature from its contracting block. The depth of the feature map decreases to half after each contracting block. The activation function of the outer most convolution layer in the expanding path is tanh as our input image pixel value ranges from -1 to 1. The contracting path and expanding path are joined by a bottleneck. The bottleneck consists of a convolution layer with LeakyReLU as activation and a transpose convolution layer with ReLu as activation. The discriminator follows the standard architecture of [1] which is used in most of the GANs architecture.

Loss Functions: Least absolute deviations(L1 loss) is widely used as a loss function for many deep networks. However, the minimization of L1 loss for generation tasks results in blurry output images lacking in high-frequency details. Hence, we add adversarial loss to our generator encourages it to produce images that lie in natural image manifold with sharp textures. We use LS-GAN [11] over vanilla GAN for better stability and faster convergence. We also use MS-SSIM loss [20] to improve the generation of turbulence at the finer level. The final loss function of our generator network is a weighted linear combination of L1 loss, MS-SSIM loss and adversarial loss which is:

$$L_{generator} = \lambda_1 L_1 + \lambda_2 L_{Adverserial} + \lambda_3 L_{MS-SSIM} \tag{1}$$

$$L_{generator} = \lambda_1 |I^{TR} - G_\alpha(I)|_1 + \lambda_2 [D_\beta(G_\alpha(I)) - 1]^2 + \lambda_3 (1 - MS\text{-}SSIM(I^{TR}, G_\alpha(I))) \tag{2}$$

where, λ_1, λ_2 and λ_3 are the hyper-parameter of the generator loss function. Similarly, the loss of our discriminator network would be:

$$L_{discriminator} = D_\beta(G_\alpha(I))^2 + (D_\beta(I^{TR}) - 1)^2 \tag{3}$$

Training Description: We train our network end-to-end by following the training methodology from [10] and [9]. To optimize our network, we use Adam [6] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for computing running average of gradient and its squares. The value of λ_1, λ_2, and λ_3 were found to be 100, 1, and 5, respectively. We first train our network for 10,000 iterations with a learning rate of $1e-4$ and then decrease the learning rate to $5e-5$ for another 15,000 iterations with a batch size of 16. For each iteration, the discriminator and the generator are updated only once.

3 Experimentation

Dataset: We use physics-based method [16] for synthesizing our turbulent dataset. The virtual camera parameters used while imaging in the turbulence

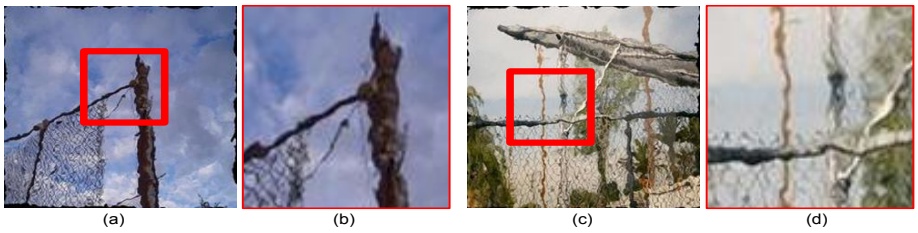

Fig. 4: (a) and (c) show the sample turbulent images of the training dataset. (b) and (d) shows the enlarged patches in red boxes.

Methods	PSNR	SSIM	MSE
Encoder-Decoder (L1 loss) [3]	12.1040	0.1071	461.8092
Encoder-Decoder (MSE loss) [3]	14.9994	0.2423	333.6431
CycleGAN [21]	14.0471	0.2803	402.6070
Pix2Pix [5]	17.8422	0.4412	254.0445
Encoder-Decoder with skip connections (L1-loss) [10]	15.4503	0.2905	320.6347
Encoder-Decoder with skip connections (MSE loss) [10]	15.5124	0.2925	318.4140
Ours	**18.2321**	**0.5211**	**234.9368**

Table 1: Quantitative results of various turbulence models. Our method outperforms baseline methods by using the proposed loss function.

environment is: focal distance = 300mm having the lens diameter of ≈ 5.57cm and a pixel size of $4 \times 10^{-3} mm$. The placement of the virtual camera is at an elevation of 5m with object distance of 3km. The value for structure constant C_n^2 which expresses the atmospheric turbulent strength is $3 \times 10^{-13} m^{-2/3}$. Using the above parameters, we rendered turbulent images by applying simulated turbulent field on ImageNet dataset. The dataset consists of 100,000 of turbulent and non-turbulent image pairs for training and the validation is performed on ImageNet validation dataset. Figure 4 shows sample turbulent images.

Evaluation Metrics: To measure the structural and perceptual similarity between the estimated turbulent image and ground-truth turbulent image, we use Peak-Signal-To-Noise-Ratio (PSNR), Structural Similarity [19] (SSIM) and Mean Squared Error (MSE). We use SSIM and PSNR as they calculate structural similarity and perceptual quality between two images. These evaluation metrics are applied to various generated turbulent images and ground-truth turbulent images to give a qualitative comparison.

Results: We compare our results of final turbulence generation model with the image-to-image translation networks: Pix2Pix [5], CycleGAN [21], Encoder-Decoder [3], and Encoder-Decoder with skip connections [10]. Our final turbulence generation model shows significant quantitative improvement over the other aforementioned methods as shown in Table 1. In Table 1, we observe that methods with MSE loss as a loss function give better results on evaluation metrics over the L1 loss. The quantitative performance of Pix2Pix is better than CycleGAN as Pix2Pix learns from paired images whereas CycleGAN

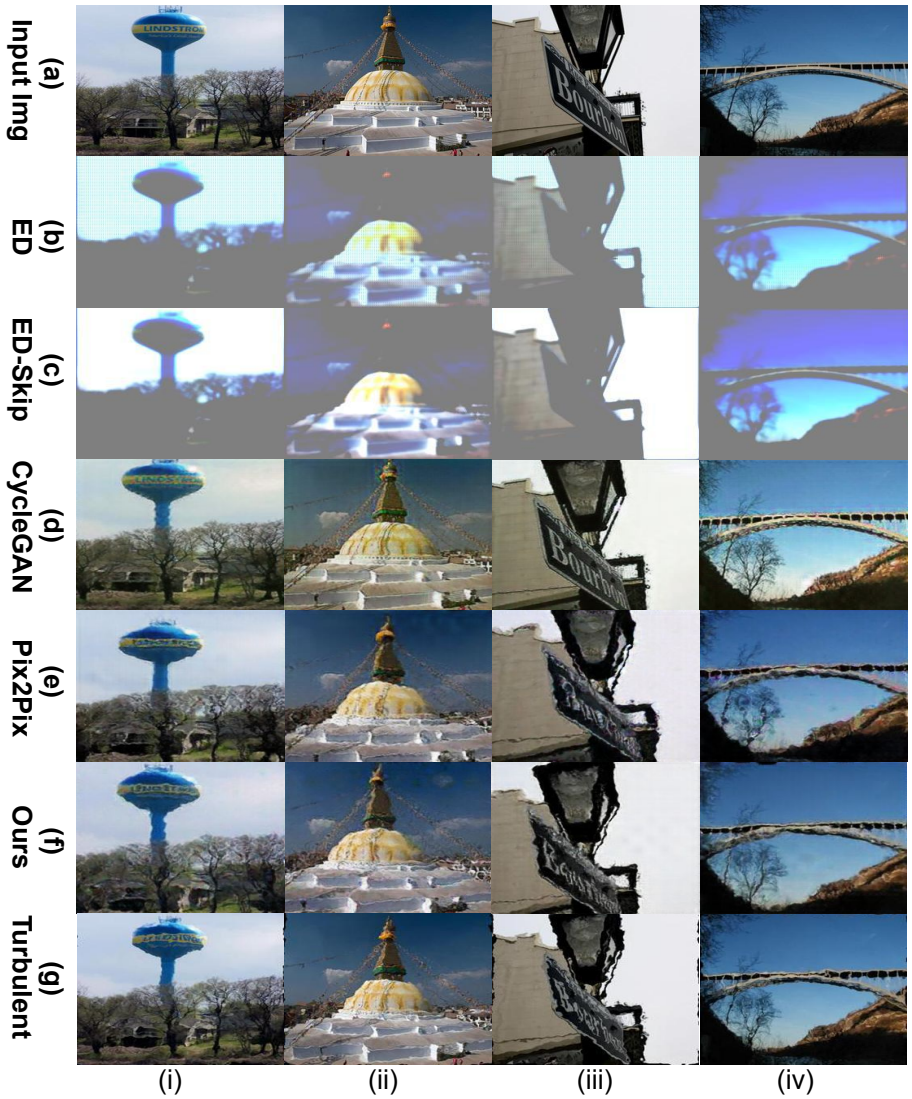

Fig. 5: Qualitative results of various turbulence models. Row a: Input Img is the non-turbulent input image to the network. Row b: ED is the Encoder-Decoder network output. Row c: ED-Skip is the Encoder-Decoder with a skip-connections network output. Row d: CycleGAN network output. Row e: Pix2Pix network output. Row f: Our network output. Row g: Turbulent is the Ground-truth turbulent image. *(Best view when zoomed)*

learns from unpaired images, which makes the generation of turbulent images difficult. Qualitative results of turbulent image generation on ImageNet images by various methods are shown in Figure 5. In Figure 5, we infer that CycleGAN and Pix2Pix struggle to generate larger geometrical distortion compared to our

Number of Images	1	100	500	1000
Schwartzman *et al.* [16]	184.367	197.357	280.472	389.559
Ours	0.018	0.979	5.243	10.646

Table 2: Computational resource time comparison between the turbulent image generation methods on different number of turbulence images to be generated. (All time values reported in seconds.)

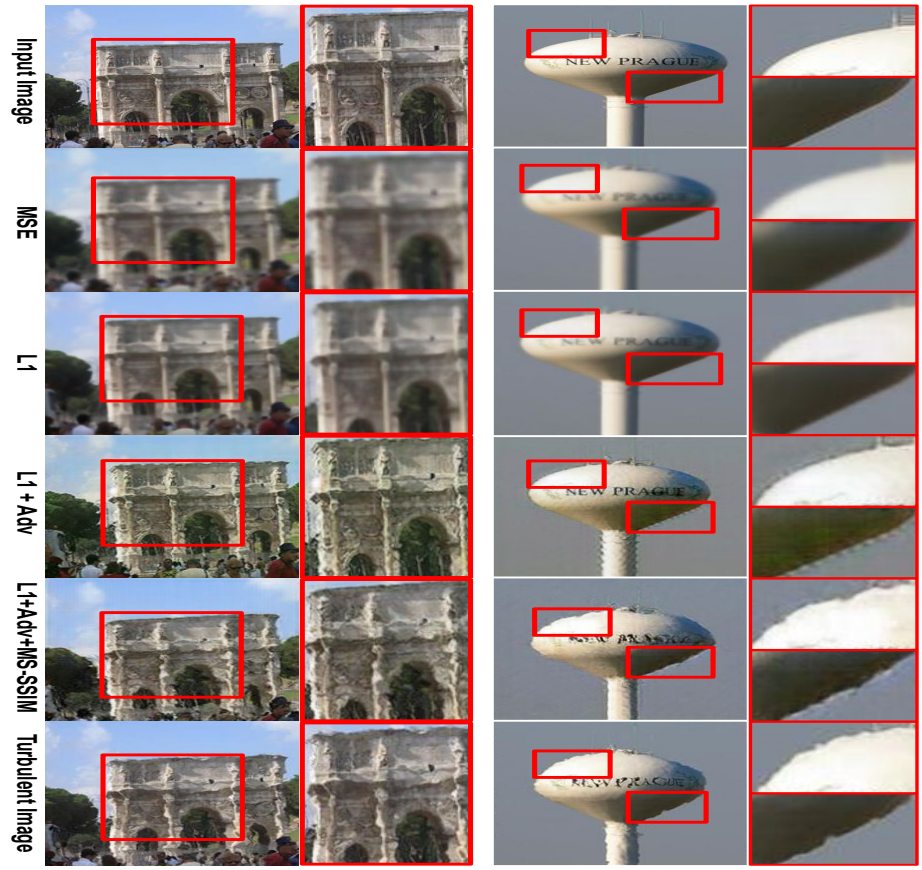

Fig. 6: Qualitative results of the ablative study on the various loss function. Row 1: Input image to the network. Row 2: MSE loss output. Row 3: L1 loss output. Row 4: L1 + Adv is the output of the combined output of L1 loss and adversarial loss. Row 5: L1 + Adv + MS-SSIM is the output of the combined output of L1 loss, MS-SSIM, and adversarial loss. Row 6: Ground-truth turbulent image. Red boxes show the zoomed image patch. *(Best viewed when zoomed)*

method. Moreover, Pix2Pix suffers from color artifacts which could be seen in Figure 5 (column e)(iii)(iv). Encoder-Decoder and Encoder-Decoder with skip connections failed to generate turbulent images with sharp details. Although encoder-decoder suffers from the checkerboard effect which is eliminated by skip

Loss Function	PSNR	SSIM	MSE
L1 loss	16.7413	0.3523	293.5724
MSE loss	17.0215	0.3915	273.3178
L1 loss + Adversarial loss	17.5243	0.4215	259.2911
L1 loss + MS-SSIM loss + Adversarial Loss	**17.6515**	**0.4402**	**256.5432**

Table 3: Ablation study results of our loss function.

connections. Table 2 shows the comparison between the computational time taken by Schwartzman *et al.* [16] and our method to generate atmospheric turbulent images. We can observe from the table that our approach requires a small fraction of computational time to generate turbulent images.

Ablation Study: We perform an ablation study on our loss function to show its advantages qualitatively and quantitatively. The ablative study is performed by training our generator architecture individually on L1 loss, MSE loss, L1 loss + Adversarial Loss, and L1 loss + Adversarial loss + MS-SSIM loss. We use L1 loss instead of MSE loss as it leads to sharper images. All the networks were trained for 10,000 iterations with the learning rate of $1e-4$ and batch size of 16. From Table 3, we observe that adding MS-SSIM loss into the total loss leads to higher PSNR and SSIM which implies that it improves the structural as well as perceptual information of the generated image. Figure 6 shows the qualitative results of the ablation where we can observe in Figure 6(row 5) the output generated by ours looks more realistic. Although in Figure 6(row 4) the output looks more promising when zoomed, it looks more like a uniform artifact.

4 Conclusion

In this paper, we proposed a turbulent image generation model by training a deep adversarial network. Unlike, the traditional turbulent image generation methods which rely on a statistical model, our approach uses data to learn the parameters of the generative adversarial network, which transforms a non-turbulent image into a turbulent image. We proposed a novel loss function that encourages the learning of finer turbulence fields. To support our claim, we performed extensive ablation studies on the loss function.

References

1. A. Radford, L.M., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In International Conference on Learning Representations (ICLR) (2016)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7564–7573 (2018)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)

4. Gutierrez, D., Seron, F.J., Munoz, A., Anson, O.: Simulation of atmospheric phenomena. Computers & Graphics **30**(6), 994–1010 (2006)
5. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Lane, R., Glindemann, A., Dainty, J., et al.: Simulation of a kolmogorov phase screen. Waves in random media **2**(3), 209–224 (1992)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436 (2015)
9. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE/CVPR conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
10. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Advances in neural information processing systems. pp. 2802–2810 (2016)
11. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
12. Nayar, S.K., Narasimhan, S.G.: Vision in bad weather. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 820–827. IEEE (1999)
13. Purohit, K., Mandal, S., Rajagopalan, A.N.: Multilevel weighted enhancement for underwater image dehazing. J. Opt. Soc. Am. A **36**(6), 1098–1108 (Jun 2019)
14. Riley, K., Ebert, D.S., Kraus, M., Tessendorf, J., Hansen, C.D.: Efficient rendering of atmospheric phenomena. Rendering Techniques **4**, 374–386 (2004)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
16. Schwartzman, A., Alterman, M., Zamir, R., Schechner, Y.Y.: Turbulence-induced 2d correlated image distortion. In: 2017 IEEE International Conference on Computational Photography (ICCP). pp. 1–13. IEEE (2017)
17. Seron, F.J., Gutierrez, D., Gutiérrez, G., Cerezo, E.: Implementation of a method of curved ray tracing for inhomogeneous atmospheres. Computers & Graphics **29**(1), 95–108 (2005)
18. Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J.M., Chari, V.: Learning to generate synthetic data via compositing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 461–470 (2019)
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
20. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging **3**(1), 47–57 (2016)
21. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
22. Zhu, X., Milanfar, P.: Removing atmospheric turbulence via space-invariant deconvolution. IEEE transactions on pattern analysis and machine intelligence **35**(1), 157–170 (2012)