Spatio-Temporal Turbulence Mitigation: A Translational Perspective

Xingguang Zhang[1] Nicholas Chimitt[1] Yiheng Chi[1] Zhiyuan Mao[2] Stanley H. Chan[1]
[1]School of Electrical and Computer Engineering, Purdue University [2]Samsung Research America
{zhan3275, nchimitt, chi14, stanchan}@purdue.edu, m940421@gmail.com

Abstract

Recovering images distorted by atmospheric turbulence is a challenging inverse problem due to the stochastic nature of turbulence. Although numerous turbulence mitigation (TM) algorithms have been proposed, their efficiency and generalization to real-world dynamic scenarios remain severely limited. Building upon the intuitions of classical TM algorithms, we present the Deep Atmospheric TUrbulence Mitigation network (DATUM). DATUM aims to overcome major challenges when transitioning from classical to deep learning approaches. By carefully integrating the merits of classical multi-frame TM methods into a deep network structure, we demonstrate that DATUM can efficiently perform long-range temporal aggregation using a recurrent fashion, while deformable attention and temporal-channel attention seamlessly facilitate pixel registration and lucky imaging. With additional supervision, tilt and blur degradation can be jointly mitigated. These inductive biases empower DATUM to significantly outperform existing methods while delivering a tenfold increase in processing speed. A large-scale training dataset, ATSyn, is presented as a co-invention to enable the generalization to real turbulence. Our code and datasets are available at https://xg416.github.io/DATUM

1. Introduction

Atmospheric turbulence is a dominant image degradation for long-range imaging systems. Reconstructing images distorted by atmospheric turbulence is an important task for many civilian and military applications. The degradation process can be considered a combination of content-invariant random pixel displacement (i.e., tilt) and random blur. Until recently, reconstruction algorithms have often been in the form of model-based solutions, often relying on modalities such as pixel registration and deblurring. Although there have been many important insights into the problem, e.g., lucky imaging, they are primarily limited to static scenes with slow processing speed.

With the development of physics-grounded data synthesis methods [7, 17, 18, 33, 65, 77], data-driven algorithms have been developed in the past two years. Most existing deep learning methods focus on single-frame problems [24, 32, 34, 44, 45, 48, 49, 53, 71]. Since the degradation is highly ill-posed, the performance of these algorithms is naturally limited, especially when attempting to generalize to real data. On the other hand, multi-frame turbulence mitigation networks [1, 26, 72] have shown greater potential for generalization across a broader spectrum of real-world test scenarios. However, these networks are adapted from generic video restoration methods and do not reflect the insights developed by traditional methods; few turbulence-specific properties are incorporated as inductive biases into their methods.

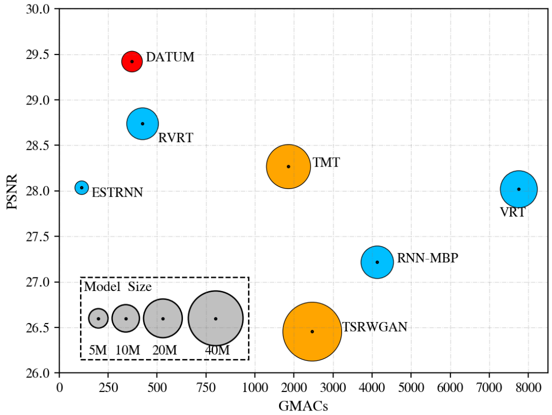


Figure 1. Benchmarking video restoration models for turbulence mitigation on our ATSyn-dynamic dataset. The circles in orange are other video-based TM networks, and the circles in blue are representative video deblurring and general restoration networks. The proposed Deep Atmospheric TUrbulence Mitigation network (DATUM) is state-of-the-art while highly efficient.

For deep learning methods to work on real-world scenarios, two common factors hinder the application of current turbulence mitigation methods: (1) the complexity of current data-driven methods is usually high, which impedes the practical deployment of these algorithms, and (2) the data synthesis models are suboptimal, either too slow to produce large-scale and diverse datasets or not accurate enough to represent the real-world turbulence profiles, restricting the generalization capability of the model trained on the data.

To overcome these pressing issues, we propose the Deep Atmospheric TUrbulence Mitigation (DATUM) network and the ATSyn dataset. We offer three contributions:

- DATUM is the first deep-learning video restoration method customized for turbulence mitigation based on classical insights. By carefully integrating the merits of classical multi-frame TM methods, we propose feature-reference registration, temporal fusion, and the decoupling of pixel rectification and deblurring as effective inductive biases in the multi-frame TM challenge.
- DATUM is the first recurrent model for turbulence restoration. It is significantly more lightweight and efficient than the prior multi-frame TM methods. On both synthetic and real data, DATUM consistently surpasses the SOTA methods while being $10\times$ faster.
- Through the integration of numerous theoretical and practical improvements in physics modeling over the Zernike-based simulators, we further propose an extensive, real-world inspired dataset ATSyn. Experiments on real-world data show that models trained on ATSyn significantly generalize better than those trained on alternative ones.

2. Related works

2.1. Turbulence modeling

Atmospheric turbulence simulation spans from computational optics to computer vision-oriented approaches. Optical simulations use split-step methods, which numerically propagate waves through phase screens that represent the atmosphere's spatially varying index of refraction [6, 21, 55, 60]. Despite the existence of moderately faster optical simulations, including brightness function-based simulations [30, 31, 66] or learning-based alternatives [46, 47], the relatively slow speed limits their application in deep learning training [43]. In computer vision simulations, pixels are first displaced according to heuristic correlation functions followed by invariant Gaussian blur [7, 33, 77], offering speed but arguably lacking physical foundations. Recent Zernike-based methods [9, 12, 13, 43] can match the statistics of optics-based simulation, achieving realistic visual quality while maintaining a fast data synthesis speed. It has been applied to turbulence mitigation [24, 25, 44, 72] to facilitate the generalization capability of those models.

2.2. Conventional turbulence mitigation

Conventional TM algorithms, since [17, 18, 65], mostly treat the TM challenge as a many-to-one restoration problem. Considering that turbulence primarily induces random tilt and blur, the common procedure in conventional algorithms is as follows. They first align the input frames to account for pixel displacements, followed by temporal fusion to combine the information from the aligned frames. Subsequently, the residual blur is often considered to be spatially invariant, allowing a blind deconvolution to be applied to produce a visually satisfactory image.

The tilt rectification is typically achieved in a two-step fashion: construct a tilt-free reference frame, then register every frame with respect to the reference. Since the pixel displacement is assumed to be zero-mean over time [17, 36], the temporal average can be assumed tilt-free [22, 40, 41, 63, 77] and hence be the reference frame. Besides that, low-rank components from all input frames are frequently used [33, 35, 69] as the reference. The registration step can be done by B-spline or optical flow based warping [40, 41, 63, 69, 77] in the spatial domain or phase correction [2, 22, 70] in the phase domain. Because of the "lucky effect" phenomenon [20] in the short-exposure turbulence, the goal of temporal aggregation is to identify and fuse the randomly emerging sharp regions, a technique known as lucky fusion [4]. [33, 42, 77] design spatial descriptors to select and score lucky regions. [2] identify and fuse sharp components in the wavelet space, and [23, 33] apply a similar principle to the sparse components derived through robust PCA. While several methods have been proposed for moving object scenarios [3, 42, 50, 52, 57], they are restricted by their assumption that the dynamic regions are rigid and can be isolated, leaving the remaining static regions to be restored using the conventional pipelines.

2.3. Learning-based turbulence mitigation

With the rapid advancements in machine learning, numerous recent learning-based methods have demonstrated superior turbulence mitigation results. The majority of them are single-frame TM methods. [32, 45, 49, 53] demonstrate promising performance using generative models with simplified turbulence properties as prior. [34, 48, 71] focus on restoring long-range face images through turbulence. [24, 44] show physics-grounded synthetic data facilitates certain degrees of generalization capability. These single-frame methods do not account for the temporal dimension and can fall short in multi-frame TM scenarios. In contrast, video-based TM algorithms [26, 72] exhibit superior adaptability by leveraging the temporal information, but their designs lack the integration of specific turbulence properties, making their model less efficient. Moreover, [26] only simulated mild turbulence effect, which restricts the generalization capability of their model. Although [72] has achieved better generalization, the point spread function (PSF) implementation is less precise, and the parameter sets are not physics-oriented. Hence, the representative of their turbulence modalities is restricted.

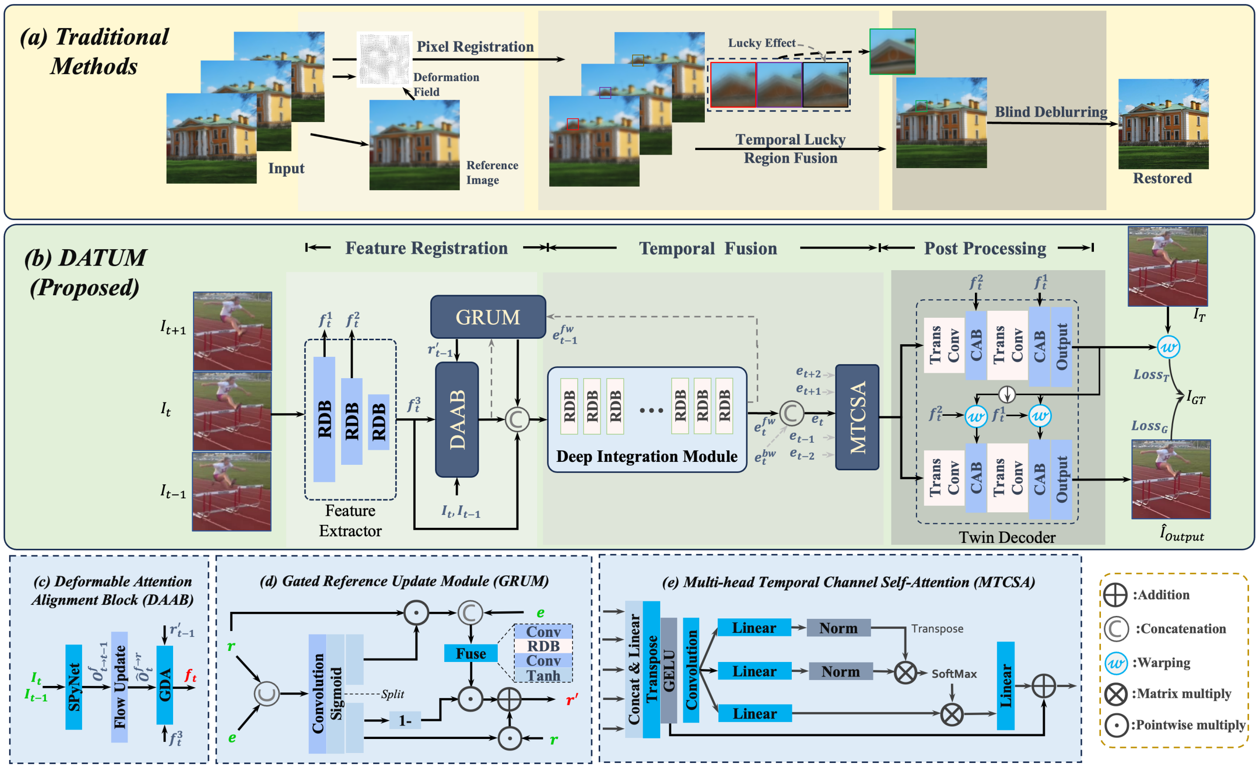


Figure 2. The proposed DATUM network. In this figure, block (a) shows the three common stages proposed by classical TM methods. The corresponding stages in DATUM are shown in block (b), which illustrates the forward time process of the t-th frame. The dashed line means the information passing from other temporal directions and frames. Block (c), (d), and (e) demonstrate the DAAB, GRUM, and MTCSA modules, respectively, where the input features are marked by green, and the output features are marked by red.

3. Proposed method

3.1. Insights from Classical Methods

Image degradation by atmospheric turbulence can be roughly described by a compositional operation of the blur $\mathcal{B}$ and the tilt $\mathcal{T}$ via the relationship $\mathbf{I} = [\mathcal{B} \circ \mathcal{T}](\mathbf{J}) + \mathbf{n}$, where $\mathbf{J}$ is the clean image, $\mathbf{I}$ is the distorted image, and $\mathbf{n}$ is the noise term. Traditional algorithms handle turbulence in three steps, as illustrated in Fig. 2:

- **Frame-to-reference registration** [77], where a reference frame is constructed from the observed images and all images are registered with respect to the reference using optical flow. In strong turbulence or dynamic scenes, constructing a reference is often difficult.
- **Lucky image fusion** [2, 27], where a "lucky" image is constructed by collecting the sharpest and most consistent patches from the inputs. However, if turbulence is strong, identifying lucky patches can be difficult.
- **Blind deconvolution** [42], where a final blind deconvolution algorithm is employed to sharpen the lucky image. The success and failure of this step depend heavily on how spatially uniform the blur in the lucky image is. Oftentimes, since the blur is spatially varying, the performance of blind deconvolution is limited.

While each step is important each has its limitations, motivating us to develop end-to-end trained networks to approximate these functions. Empowered by training on our physical-grounded dataset, our network enjoys the inductive biases of those insights while avoiding their limitations.

3.2. DATUM network

3.2.1 Overview

The block diagram of the DATUM network is depicted in Fig. 2. We first summarize these three components and describe them in detail in the next subsections.

Feature-to-reference registration. This component is analogous to the classical frame-to-reference registration. For each input frame I_t at time t, we first extract three levels of features $f_t^{\{1,2,3\}}$. We propose the Deformable Attention Alignment Block (DAAB) to register the high-level feature f_t^3 to a previously hidden reference map r'_{t-1}. We also propose the Gated Reference Update Module (GRUM) updates this reference feature recurrently, which is inspired by the gated recurrent unit [5, 15] and illustrated in Fig. 2.

Temporal fusion. This component is analogous to the classical lucky fusion step. The registered feature f_t, together with r'_{t-1} and f_t^3, are fused by a new Deep Integration Module (DIM). DIM consists of a series of Residual Dense Blocks (RDB) [73] and is used to produce the forward embedding e_t^{fw}. Since e_t^{fw} is a deep feature, it is presumed to be free of tilt and is thus utilized for updating the reference feature for the subsequent frame. After the bidirectional recurrent process, we perform a temporal fusion of e_t^{fw} by augmenting it with the backward embedding e_t^{bw} and bidirectional embeddings from neighboring frames. We propose the Multi-head Temporal-Channel Self-Attention (MTCSA) module for this purpose.

Post processing. In the final stage, the temporally fused features are decoded to form the turbulence-free image. This decoding involves a twin of decoders. The first predicts a reverse tilt map that rectifies the shallow features, and the second subsequently reconstructs the clean image.

3.2.2 components

Feature registration via Deformable Attention Alignment Block (DAAB). In classical methods, a crucial stage for turbulence mitigation is registering the input frames to the tilt-free reference frame. This reference frame is usually obtained by temporal averaging or using variants of principle component analysis. However, these methods may not be applicable to dynamic videos. Since learning-based video TM is possible [26, 72], the deep feature of a video TM network can be considered tilt-mitigated to work as the reference feature for the next input feature. This section explains our method to use deformable attention to facilitate feature registration in our DATUM network.

The computations in the DDAB are summarized in Algorithm 1, where $(A;B)$ denotes warping A by deformation field B, $\phi(A;p)$ denotes sampling A by positions p. W_K, W_V, W_Q, and W are linear projections on the channel dimension, and σ denotes the SoftMax. The optical flow at line 3 is estimated with the SPyNet [54], and lines 6-11 are inspired by the guided deformation attention (GDA) [38].

Algorithm 1 Deformation Attention Alignment Block

```
1:  Input: Current frame feature $f_t^3$, reference feature $r'_{t-1}$
    and alignment flow from last frame $O_{t-1}^{f\to r}$, two down-
    sampled frames $I_t$ and $I_{t-1}$
2:  Output: Updated feature $f_t$ and flow $O_t^{f\to r}$
    ▷ Estimate rough deformation field $\hat{O}_t^{f\to r}$ that register
    feature $f_t^3$ to reference $r'_{t-1}$
3:  Estimate the optical flow $O^f_{t\to t-1}$ from $I_t$ and $I_{t-1}$.
4:  $\hat{O}_t^{f\to r} \leftarrow O_{t-1}^{f\to r} + (O^f_{t\to t-1}; O_{t-1}^{f\to r})$
5:  Pre-align $\hat{f}_t \leftarrow (\hat{O}_t^{f\to r}, f_t^3)$
    ▷ Register input feature to reference frame using multi-
    group multi-head deformation attention
6:  for all group $g$ do
        ▷ Predict offsets $o_t^{(g)}$
7:      $\Delta o_t^{(g)} \leftarrow \text{RDB}(\text{Concat}(\hat{f}_t, r'_{t-1}, \hat{O}_t^{f\to r}))$
8:      $o_t^{(g)} \leftarrow \hat{O}_t^{f\to r} + \Delta o_t^{(g)}$
        ▷ Compute the $g$-th aligned feature $\hat{f}_t^{(g)}$:
9:      $K^{(g)} \leftarrow \phi(f_t^3 W_K; o_t^{(g)})$, $V^{(g)} \leftarrow \phi(f_t^3 W_V; o_t^{(g)})$
10:     $Q \leftarrow r'_{t-1} W_Q$, $\hat{f}_t^{(g)} \leftarrow \sigma(QK^{(g)T}/\sqrt{C})V^{(g)}$
11: end for
12: Fuse all groups $f_t \leftarrow \text{Concat}(\{\hat{f}_t^{(g)}\})W$
13: Update final alignment flow $O_t^{f\to r}$ by mean of $\{o_t^{(g)}\}$
14: Output $f_t \leftarrow f_t + \text{FeedForward}(f_t)$
```

Temporal fusion via Multi-head Temporal Channel Self-Attention (MTCSA). After feature registration and deep integration, we propose to augment the embedding with contra-directional information, which is essential to ensure consistent restoration quality across various frames. In addition, like classical methods, a spatially adaptive fusion with adjacent frames is advantageous. We propose the Multi-head Temporal-Channel Self-Attention (MTCSA), as illustrated in Fig. 2. The MTCSA begins by concatenating channels from multiple frames, followed by a 1×1 convolution to shrink the channel dimension. Separable convolution is used to construct the spatially varying query, key, and value on the temporal and channel dimensions, and the dynamic fusion is facilitated by self-attention. Finally, a residual connection is used to stabilize training. Considering the quadratic complexity of MTCSA relative to window size, this size is kept moderate. Additionally, we integrate a hard-coded positional embedding wherein features from the focal frame are positioned at the end. This strategy is essential for boundary frames with disproportionate neighboring frames on either side.

Twin decoder and loss function Given the refined feature embedding from the MTCSA, we also developed a twin decoder to progressively remove the tilt and blur, as shown in Fig. 2. The decoder uses transposed convolution for upsampling and channel attention blocks (CAB) [68] for decoding. Before decoding in higher levels, the deep features are concatenated with the shallow features to facilitate the residual connection like a typical UNet [56]. Since the deep and shallow features are misaligned by the random tilt $\mathcal{L}$, we propose to first rectify the shallow features by the estimated inverse tilt field $\hat{\mathcal{T}}^{-1}$ estimated in the first stage. The tilt-rectification is optimized by reducing the loss:

$$\mathcal{L}_{\text{tilt}} = \mathcal{L}_{\text{char}}(\boldsymbol{I}_{\text{GT}}, (\boldsymbol{I}_{\text{tilt}}; \hat{\mathcal{T}}^{-1})) \qquad (1)$$

Where $\mathcal{L}_{\text{char}}$ denotes the Charbonnier loss [10], $\boldsymbol{I}_{\text{GT}}$ is the input frame and $\boldsymbol{I}_{\text{tilt}}$ is the tilt-only frame that can be produced without additional cost by our data synthesis method. In the second stage, the rectified shallow features are jointly decoded with the deep features to generate the final recon-

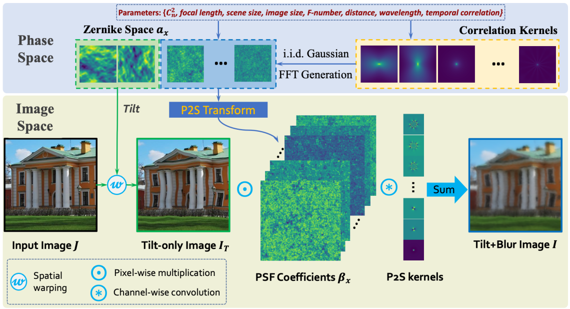


Figure 3. Scheme of our data synthesis method.

struction $\hat{\boldsymbol{I}}$. The overall loss function is computed by:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{tilt}} + \alpha_2 \mathcal{L}_{\text{char}}(\boldsymbol{I}_{\text{GT}}, \hat{\boldsymbol{I}}) \tag{2}$$

where weights α_1 and α_2 are empirically set to 0.2 and 0.8.

3.3. ATSyn dataset

3.3.1 Physics-based data synthesis

As introduced previously, the ground truth image $\boldsymbol{J}$ is first geometrically distorted and then blurred to produce the degraded image $\boldsymbol{I}$ in our synthesis method. Data synthesis for the turbulence effect essentially requires a physics-grounded representation of and . We adopted the Zernike-based turbulence simulator [12, 13] and improved it with non-trivial modifications. Fig. 3 presents the scheme of our implementation. The and is generated from the phase distortion represented by Zernike polynomials $\{\mathbf{Z}_i\}$ [51] as the basis, with corresponding coefficients $\mathbf{a}_i$ where i ranging from 1 to 36. Among all 36 coefficients, $i = 1$ denotes the current component, $i = 2, 3$ controls the by a constant scale, and the rest high order Zernike coefficients contribute to the blur effect.

The phase distortion can be assumed as a wide sense stationary (WSS) random field [13]. Hence, it can be sampled with Fast Fourier Transform (FFT) from white Gaussian noise and the autocorrelation map. Transforming the phase distortion to the spatial domain point spread functions (PSF) can be achieved by the Phase-to-Space (P2S) transform, which transforms the sampled Zernike coefficients to spatial coefficients $\boldsymbol{\beta}$, assuming the PSFs can be represented by a low-rank approximation of 100 basis $\boldsymbol{\psi}$ and corresponding $\boldsymbol{\beta}$. The overall degradation in the spatial domain is implemented by

$$\boldsymbol{I} = \sum_{k=1}^{100} \boldsymbol{\psi}_k \circledast (\boldsymbol{\beta}_k \cdot (\boldsymbol{J};)) + \boldsymbol{n}, \tag{3}$$

where $\circledast$ denotes the depth-wise convolution. Although subtle, this fundamentally generates more reliable degradation than the simulator in [72], as elaborated in [14]. Except for this, our correlation kernels are more precise by incorporating the continuous C_n^2 path technique [11].

3.3.2 Guideline of implementation

With the proposed simulator, we created the ATSyn dataset to match various real-world turbulence conditions and benchmark deep neural networks for turbulence mitigation. This dataset is segmented into two distinct subsets based on scene type: the *ATSyn-dynamic* and *ATSyn-static*. The dynamic sequences contain camera or object motion, whereas the static sequences are each associated with only one underlying clean image. We adopted parameters including focal length, F-number, distance, wavelength, scene size, and sensor resolution to control the simulation. In comparison with the synthetic dataset introduced in [72], which utilized the D/r_0 [19] and empirically chosen blur kernel size, our dataset's parameter space more closely aligns with actual camera settings, making it more representative.

ATSyn-dynamic contains 4,350 training and 1,097 validation instances synthesized from [26, 58], and ATSyn-static contains 2,000 and 1,000 instances synthesized from the Places dataset [75] for training and validation, respectively. Those instances have varying numbers of frames, each with a distinct turbulence parameter set. Besides ground truth and fully degraded videos, ATSyn further provides associated $\mathcal{T}$-only videos to facilitate the training of $\mathcal{L}_{\text{tilt}}$ in Eq. 1. We categorize the turbulence parameters by three levels: *weak*, *medium*, and *strong*. The range of turbulence parameters is determined by matching with a large-scale, long-range video dataset [16] and other real-world videos, with more details in the supplementary document.

4. Experiments

4.1. Training setting

This section describes how we trained our DATUM and other models. Except for turbulence mitigation networks [26, 44, 72], we also benchmarked several representative video restoration [37, 38] and deblurring networks [74, 76] for a more thorough comparison.

To train the proposed model, we used the Adam optimizer [29] with the Cosine Annealing learning rate schedule [39]. The initial learning rate is 2×10^{-4}, and batch size is 8. All dynamic scene TM networks in this experiment are trained end-to-end from scratch for 800K iterations. To get their static-scene variant, we fine-tuned them on the static-scene modality with half the initial learning rate and 400K iterations. We clip the gradient if the L2 norm exceeds 20 to prevent gradient explosion during inference.

We trained the ESTRNN [74], RNN-MBP [76], and RVRT [74] with the same configuration as DATUM. The number of input frames of DATUM and ESTRNN during training is set to 30 for ATSyn-dynamic and 36 for ATSyn-static. Since RNN-MBP and RVRT require much more resources to train, the number of input frames is set to 16. Because TSRWGAN [26], TMT [72], and TurbNet [44] are

Methods	TurbNet [44]	TSRWGAN [26]	VRT [37]	TMT [72]	RNN-MBP [76]	ESTRNN [74]	RVRT [38]	DATUM [ours]
PSNR	24.2229	26.3262	27.6114	27.7419	27.7152	27.3469	27.8512	**28.5875**
SSIM$_{CW}$	0.8230	0.8596	0.8691	0.8741	0.8730	0.8617	0.8788	**0.8803**

Table 1. Preliminary study: evaluate on TMT's synthetic dynamic scene data [72]. SSIM$_{CW}$ denotes Complex Wavelet SSIM.

Turbulence Level	Weak		Medium		Strong		Overall		Cost	
Methods	PSNR	SSIM$_{CW}$	PSNR	SSIM$_{CW}$	PSNR	SSIM$_{CW}$	PSNR	SSIM$_{CW}$	Size	FPS
TSRWGAN [26]	27.0844	0.8575	26.7046	0.8514	25.4230	0.8372	26.4541	0.8493	46.28	0.87
TMT [72]	29.1183	0.8836	28.5050	0.8791	26.9744	0.8552	28.2665	0.8734	26.04	0.80
VRT [37]	28.8453	0.8797	28.2628	0.8769	26.7492	0.8506	28.0179	0.8699	18.32	0.17
RNN-MBP [76]	27.9243	0.8699	27.4742	0.8642	26.0812	0.8495	27.2161	0.8618	14.16	1.14
ESTRNN [74]	28.9805	0.8750	28.3338	0.8697	26.8897	0.8463	28.1347	0.8645	2.468	27.65
RVRT [38]	29.6080	0.8845	28.9605	0.8806	27.5344	0.8595	28.7672	0.8756	13.50	2.43
DATUM [ours]	**30.2058**	**0.8857**	**29.6203**	**0.8829**	**28.2550**	**0.8640**	**29.4222**	**0.8781**	5.754	9.17

Table 2. Performance comparison on the ATSyn-dynamic set, we list the image quality scores on different turbulence levels and frame-wise resource consumption (measured with 960 × 540 frame sequences on RTX 2080 Ti).

all designed for turbulence mitigation, we trained them following the original paper and public code.

4.2. Comparison on dynamic scene modality

We first trained and evaluated all networks for comparison on a previous Zernike-based synthetic dataset [72] for preliminary study. We choose PSNR and Complex Wavelet Structure Similarity [59] (CW-SSIM) as the criterion in this paper, and the reason for selecting CW-SSIM rather than SSIM is provided in the supplementary document. The result in Table 1 shows our DATUM outperforms the previous state-of-the-art TMT [72] with 5× fewer parameters and over 10× faster inference speed. We also benchmark a representative single-frame TM network [44] to demonstrate the superiority of multi-frame TM methods. Next, we present extensive results from the ATSyn-dynamic dataset in Table 2. Our model outperforms all other networks by a significant margin, while it is the second smallest network among all models and the most efficient network among all existing turbulence mitigation networks.

4.3. Comparison on static scene modality

When training on the ATSyn-static, the loss is computed between the single ground truth and all output frames. For testing, we instead calculate the average score of the central four frames in the entire output sequence (for single-directional models, we use the last 4). We evaluated the performance on the ATSyn-static and the turbulence text dataset [64], and the result is shown in Table 3. The turbulence text dataset contains 100 sequences of text images, each a static scene of degraded text pattern captured at 300 meters or farther. Real-world turbulence videos do not have ground truth, while [64] uses the accuracy score of pre-trained text recognition models CRNN [61], DAN [67], and ASTER [62] as metrics, where a better turbulence mitigation offers better recognition performances. Our model is trained on a wide range of turbulence conditions and generic data, without specific augmentation tricks, yet performs on par with the best systems in the UG2+ turbulence challenge [64]. Our model outperforms other networks trained on the ATSyn-static dataset by an even larger margin.

Benchmark	ATSyn-static		Turb-Text (%)
Methods	PSNR	SSIM$_{CW}$	CRNN/DAN/ASTER
TSRWGAN [26]	23.16	0.8407	60.30 / 73.90 / 74.40
TMT [72]	24.51	0.8716	80.90 / 87.25 / 88.55
VRT [37]	24.27	0.8641	76.30 / 84.45 / 83.60
RNN-MBP [76]	24.64	0.8775	51.35 / 65.00 / 64.30
ESTRNN [74]	26.23	0.9017	87.10 / 97.80 / 96.95
RVRT [38]	25.71	0.8876	86.40 / 89.00 / 89.20
DATUM [ours]	**26.76**	**0.9102**	**93.55 / 97.95 / 97.25**

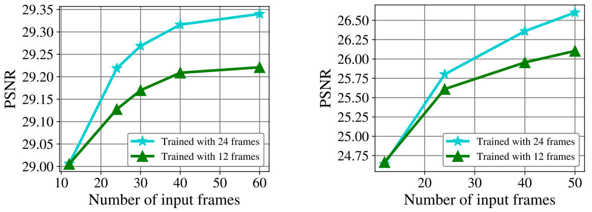
Table 3. Static scene modality. CRNN/DAN/ASTER are the text recognition rates of these three models from the restored images.

4.4. Ablation study

Our ablation study examines key elements that introduce effective inductive biases of our model, including the use of additional frames, recurrent reference updating, feature-reference registration, and multi-frame embedding fusion.

Influence of the number of input frames. The number of input frames for both training and inference matters for recurrent-based networks, especially in turbulence mitigation. Since turbulence degradation is caused by zero-mean stochastic phase distortion, the more frames the network can perceive, the better the non-distortion state it can evaluate. This is particularly valid for static scene sequences, where the pixel-level turbulence statistics are much easier to track and analyze through time.

We trained two models with 12-frame and 24-frame inputs and presented their respective performance during inference in Fig. 4. This figure shows in the temporal range of our experimental setting, a positive correlation between the



(a) On ATSyn-dynamic

(b) On ATSyn-static

Figure 4. Influence of the number of input frames.

performance and the number of input frames always exists, especially on the static scene modality where an over 1 dB boost can be obtained with more frames. This phenomenon suggests one of the success factors for turbulence mitigation is the capability of fusing more frames, similar to the video super-resolution problem [8].

Influence of DAAB, MTCSA, GRUM, and twin decoder. The design of DAAB and MTCSA are inspired by pixel registration and lucky fusion in the conventional TM methods. Although our spatial registration and temporal fusion are implemented at the feature level, they are still effective in turbulence mitigation, as shown in Table 4.

While the MTCSA fuses embeddings from multiple frames in a sliding window manner, determining the optimal window size is crucial. If the window size is too small, the temporal fusion only relies on the implicit temporal propagation by the recurrent unit, limiting the performance; if the window size is too large, because of the quadratic complexity along the temporal dimension, the MTCSA becomes very resource-demanded, and the network becomes less flexible to deal with a small number of input frames. We investigated the temporal window size of the MTCSA module, as shown in Fig. 4, where we found that five frames meet the trade-off between performance and efficiency.

The GRUM utilizes a gating mechanism in the recurrent network to facilitate more extended temporal dependency [5, 15]. It fuses the reference feature with deeper embeddings in a more adaptive manner, which also turns out to be effective. Finally, in the post-processing stage, we compared the two-stage twin decoder with the one-stage plain decoder. We found that by incorporating additional supervision and rectifying shallow features in the decoding stage, better performance can be obtained.

4.5. Comparison on real-world data

In this section, we demonstrate our data's generalization capability qualitatively and quantitatively on real-world data.

Given the impracticality of directly obtaining ground truth images for real-world turbulence scenarios, quantitative performance evaluation typically involves applying restored images to downstream tasks, as noted in [24, 44, 49]. Adopting this approach, we evaluated various restoration methods using the turbulence text dataset. The results are presented in Fig. 5, revealing two key insights: 1) our proposed ATSyn-static dataset enhances the generalization capabilities of other TM methods. 2) on both synthetic and real-world sequences, DATUM consistently outperforms other models trained on our dataset. To further validate the effectiveness of our modifications to the Zernike-based simulator, we extensively compared DATUM trained on our ATSyn-dynamic dataset and TMT's dataset [72]. We first enhance the long-range subset in the BRIAR dataset [16] by those two versions, run the same pre-trained face recognition model [28] on the enhanced images, and it yields the result provided in Table 5. We can observe the ATSyn-dynamic dataset improved network performance on real-world videos compared to the [72] dataset. These comparisons demonstrate our method facilitates better generalization of both scene types than other existing datasets.

Components	PSNR / SSIM	Size	GMACs
Base (MTCSA-1f)	28.62 / 0.8465	3.912	261.5
Base (MTCSA-3f)	28.79 / 0.8497	4.131	272.7
∗ Base (MTCSA-5f)	28.87 / 0.8522	4.768	304.2
Base (MTCSA-7f)	28.92 / 0.8532	5.808	358.1
+ GRUM	29.06 / 0.8576	4.894	317.7
+ DAAB	29.33 / 0.8638	5.241	351.8
+ Twin Decoder	29.42 / 0.8647	5.754	372.7

Table 4. Ablation study. We conducted experiments on the ATSyn-dynamic set by adding each proposed component progressively and observed a constant performance improvement.

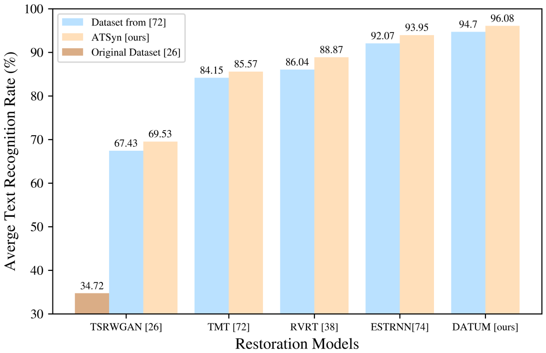
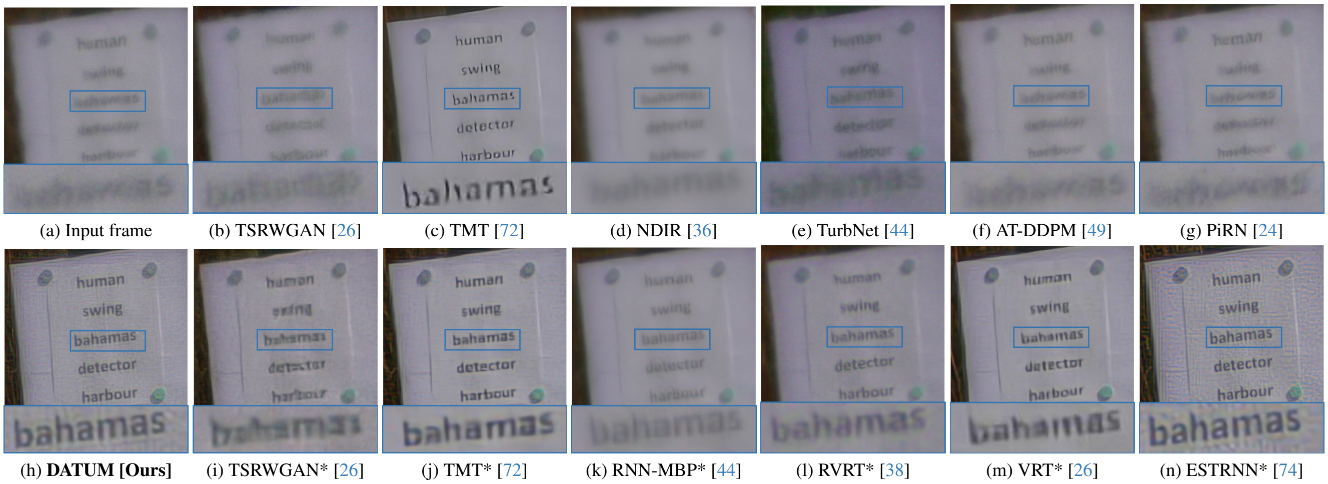


Figure 5. Comparison on the real-world turbulence-text dataset. The metric is the average text recognition accuracy of CRNN, DAN, and ASTER tested on the restored images.

Face Retrieval	Degraded	Simulator in [72]	Our simulator
Rank 5	37.75%	38.83%	**39.18%**
Rank 10	40.59%	41.83%	**42.18%**
Rank 20	45.29%	46.40%	**46.70%**

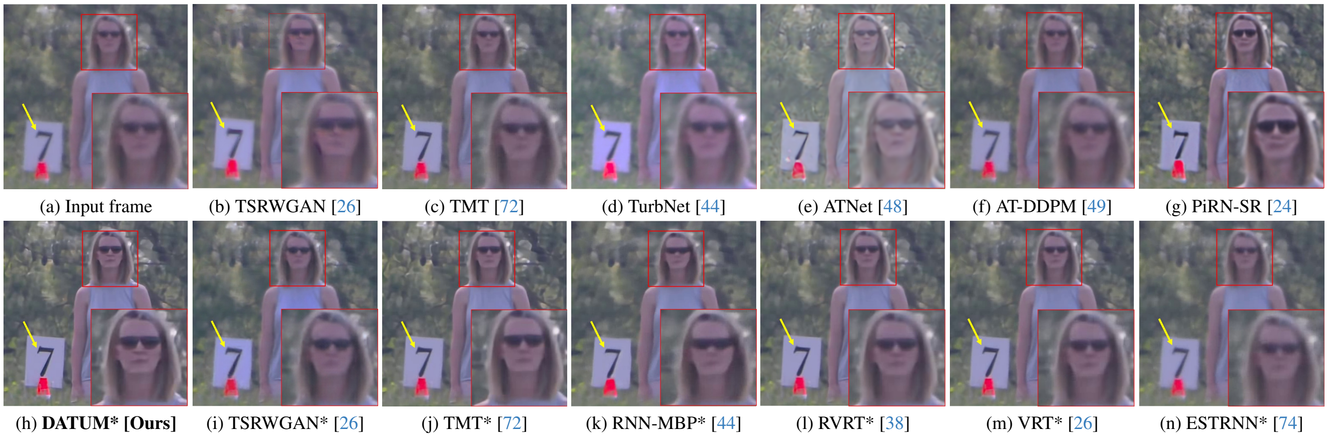
Table 5. Face recognition results on a subset of the BRIAR dataset.

We also provide a qualitative comparison in Fig. 6 and 7 to demonstrate the advance of our network and dataset. By comparing the same networks trained by our data and their original checkpoints, our data enhances their general-


(a) Input frame (b) TSRWGAN [26] (c) TMT [72] (d) NDIR [36] (e) TurbNet [44] (f) AT-DDPM [49] (g) PiRN [24]

(h) **DATUM [Ours]** (i) TSRWGAN* [26] (j) TMT* [72] (k) RNN-MBP* [44] (l) RVRT* [38] (m) VRT* [26] (n) ESTRNN* [74]

Figure 6. Qualitative comparison on the turbulence-text dataset [64]. The input frame (a) is the 49th frame of the 94th sequence in [64]. Figures on the top row are restoration results of corresponding TM methods using their original model and checkpoints. Figures on the bottom row are TM or general restoration models (marked by *) trained on our *ATSyn-static* dataset.


(a) Input frame (b) TSRWGAN [26] (c) TMT [72] (d) TurbNet [44] (e) ATNet [48] (f) AT-DDPM [49] (g) PiRN-SR [24]

(h) **DATUM* [Ours]** (i) TSRWGAN* [26] (j) TMT* [72] (k) RNN-MBP* [44] (l) RVRT* [38] (m) VRT* [26] (n) ESTRNN* [74]

Figure 7. Qualitative comparison on a dynamic scene sample from the BRIAR dataset [16]. Figures on the top row are the original restoration results of corresponding TM methods. Figures on the bottom row are models (marked by *) trained on *ATSyn-dynamic* dataset.

ization capability. On the other hand, by comparison among all networks trained on our dataset, our model significantly outperforms other networks.

5. Conclusion

In this research, we introduced a novel approach leveraging deep learning to address the enduring challenge of atmospheric turbulence mitigation. Taking a translational perspective, our method integrated the strengths of traditional turbulence mitigation (TM) techniques into a neural network architecture. This fusion elevated our network to state-of-the-art performance while ensuring significantly enhanced efficiency and speed compared to prior TM models. Additionally, we developed a physics-based synthesis method that accurately models the degradation process. This led to the creation of an extensive synthetic dataset covering a diverse spectrum of turbulence effects. Utilizing this dataset, we facilitated a stronger generalization capability for data-driven models than other existing datasets.

Acknowledgments and Disclosure of Funding The research is based upon work supported in part by the Intelligence Advanced Research Projects Activity (IARPA) under Contract No. 2022-21102100004, and in part by the National Science Foundation under the grants CCSS-2030570 and IIS-2133032. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] Nantheera Anantrasirichai. Atmospheric turbulence removal with complex-valued convolutional neural network. *Pattern Recognition Letters*, 171:69–75, 2023. 1

[2] Nantheera Anantrasirichai, Alin Achim, Nick G. Kingsbury, and David R. Bull. Atmospheric turbulence mitigation using complex wavelet-based fusion. *IEEE Transactions on Image Processing*, 22(6):2398 – 2408, 2013. 2, 3

[3] Nantheera Anantrasirichai, Alin Achim, and David Bull. Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion. In *IEEE International Conference on Image Processing*, pages 2895 – 2899, 2018. 2

[4] Mathieu Aubailly, Mikhail A. Vorontsov, Gary W. Carhart, and Michael T. Valley. Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach. In *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation III*. Proc. SPIE 7463, 2009. 2

[5] Nicolas Ballas, Li Yao, Christopher J. Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *4th International Conference on Learning Representations (ICLR)*, 2016. 3, 7

[6] Jeremy P. Bos and Michael C. Roggemann. Technique for simulating anisoplanatic image formation over long horizontal paths. *Optical Engineering*, 51(10):101704, 2012. 2

[7] Wai Ho Chak, Chun Pong Lau, and Lok Ming Lui. Subsampled turbulence removal network. *Mathematics, Computation and Geometry of Data*, 1:1 – 33, 2021. 1, 2

[8] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5962–5971, 2022. 7

[9] Stanley H Chan and Nicholas Chimitt. Computational imaging through atmospheric turbulence. *Foundations and Trends® in Computer Graphics and Vision*, 15(4):253–508, 2023. 2

[10] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997. 4

[11] Nicholas Chimitt and Stanley Chan. Anisoplanatic optical turbulence simulation for near-continuous Cn2 profiles without wave propagation. *Optical Engineering*, 62(7):078103, 2023. 5

[12] Nicholas Chimitt and Stanley H. Chan. Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients. *Optical Engineering*, 59(8):083101, 2020. 2, 5

[13] Nicholas Chimitt, Xingguang Zhang, Zhiyuan Mao, and Stanley H Chan. Real-time dense field phase-to-space simulation of imaging through atmospheric turbulence. *IEEE Transactions on Computational Imaging*, 2022. 2, 5

[14] Nicholas Chimitt, Xingguang Zhang, Yiheng Chi, and Stanley H. Chan. Scattering and gathering for spatially varying blurs. *IEEE Transactions on Signal Processing*, 72:1507–1517, 2024. 5

[15] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014. 3, 7

[16] David Cornett, Joel Brogan, Nell Barber, Deniz Aykac, Seth Baird, Nicholas Burchfield, Carl Dukes, Andrew Duncan, Regina Ferrell, Jim Goddard, et al. Expanding accurate person recognition to new altitudes and ranges: The briar dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 593–602, 2023. 5, 7, 8

[17] D. H. Frakes, J. W. Monaco, and M. J. T. Smith. Suppression of atmospheric turbulence in video using an adaptive control grid interpolation approach. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1881 – 1884, 2001. 1, 2

[18] Donald Fraser, Glen Thorpe, and Andrew Lambert. Atmospheric turbulence visualization with wide-area motion-blur restoration. *JOSA A*, 16(7):1751–1758, 1999. 1, 2

[19] D. L. Fried. Statistics of a geometric representation of wavefront distortion. *Journal of the Optical Society of America*, 55(11):1427 – 1435, 1965. 5

[20] D. L. Fried. Probability of getting a lucky short-exposure image through turbulence. *Journal of Optical Society of America*, 68(12):1651 – 1658, 1978. 2

[21] R. C. Hardie, J. D. Power, D. A. LeMaster, D. R. Droege, S. Gladysz, and S. Bose-Pillai. Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis. *Optical Engineering*, 56(7):071502, 2017. 2

[22] Russell C Hardie, Michael A Rucci, Santasri Bose-Pillai, and Richard Van Hook. Application of tilt correlation statistics to anisoplanatic optical turbulence modeling and mitigation. *Applied Optics*, 60(25):G181–G198, 2021. 2

[23] R. He, Z. Wang, Y. Fan, and D. Feng. Atmospheric turbulence mitigation based on turbulence extraction. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1442 – 1446, 2016. 2

[24] Ajay Jaiswal, Xingguang Zhang, Stanley H. Chan, and Zhangyang Wang. Physics-driven turbulence image restoration with stochastic refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12170–12181, 2023. 1, 2, 7, 8

[25] Weiyun Jiang, Vivek Boominathan, and Ashok Veeraraghavan. NeRT: Implicit neural representations for unsupervised atmospheric turbulence mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4236–4243, 2023. 2

[26] D. Jin, Y. Chen, Y. Lu, J. Chen, P. Wang, Z. Liu, S. Guo, and X. Bai. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3:876 – 884, 2021. 1, 2, 4, 5, 6, 8

[27] Neel Joshi and Michael F. Cohen. Seeing mt. rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal. In *2010 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2010. 3

[28] Minchul Kim, Anil K. Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18750–18759, 2022. 7

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. 5

[30] Svetlana L. Lachinova, Mikhail A. Vorontsov, Vadim V. Dudorov, Valeriy V. Kolosov, and Michael T. Valley. Anisoplanatic imaging through atmospheric turbulence: brightness function approach. In *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation*, page 67080E. SPIE, 2007. 2

[31] Svetlana L. Lachinova, Mikhail A. Vorontsov, Grigorii A. Filimonov, Daniel A. LeMaster, and Matthew E. Trippel. Comparative analysis of numerical simulation techniques for incoherent imaging of extended objects through atmospheric turbulence. *Optical Engineering*, 56(7), 2017. 2

[32] C. P. Lau and L. M. Lui. Subsampled turbulence removal network. *Mathematics, Computation and Geometry of Data*, 1(1):1 – 33, 2021. 1, 2

[33] C. P. Lau, Y. H. Lai, and L. M. Lui. Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps. *Inverse Problems*, 2019. 1, 2

[34] C. P. Lau, H. Souri, and R. Chellappa. ATFaceGAN: Single face semantic aware image restoration and recognition from atmospheric turbulence. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):240 – 251, 2021. 1, 2

[35] Dalong Li, Russell M Mersereau, and Steven Simske. Atmospheric turbulence-degraded image restoration using principal components analysis. *IEEE Geoscience and Remote Sensing Letters*, 4(3):340–344, 2007. 2

[36] Nianyi Li, Simron Thapa, Cameron Whyte, Albert W. Reed, Suren Jayasuriya, and Jinwei Ye. Unsupervised non-rigid image distortion removal via grid deformation. In *IEEE/CVF International Conference on Computer Vision*, pages 2522 – 2532, 2021. 2, 8

[37] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. VRT: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022. 5, 6

[38] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. In *Advances in Neural Information Processing Systems*, 2022. 4, 5, 6, 8

[39] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations (ICLR)*, 2017. 5

[40] Y. Lou, S. Ha Kang, S. Soatto, and A. Bertozzi. Video stabilization of atmospheric turbulence distortion. *Inverse Problems and Imaging*, 7(3):839 – 861, 2013. 2

[41] Y. Mao and J. Gilles. Non rigid geometric distortions correction - application to atmospheric turbulence stabilization. *Inverse Problems and Imaging*, 3:531 – 546, 2012. 2

[42] Z. Mao, Nicholas Chimitt, and Stanley H. Chan. Image reconstruction of static and dynamic scenes through anisoplanatic turbulence. *IEEE Transactions on Computational Imaging*, 6:1415 – 1428, 2020. 2, 3

[43] Z. Mao, N. Chimitt, and S. H. Chan. Accelerating atmospheric turbulence simulation via learned phase-to-space transform. In *IEEE/CVF International Conference on Computer Vision*, pages 14759 – 14768, 2021. 2

[44] Zhiyuan Mao, Ajay Jaiswal, Zhangyang Wang, and Stanley H Chan. Single frame atmospheric turbulence mitigation: A benchmark study and a new physics-inspired transformer model. In *European Conference on Computer Vision*, pages 430–446. Springer, 2022. 1, 2, 5, 6, 7, 8

[45] Kangfu Mei and Vishal M Patel. LTT-GAN: Looking through turbulence by inverting GANs. *IEEE Journal of Selected Topics in Signal Processing*, 2023. 1, 2

[46] Kevin J. Miller and Todd Du Bosq. A machine learning approach to improving quality of atmospheric turbulence simulation. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXXII*, page 117400N. Proc. SPIE 11740, 2021. 2

[47] Kevin J. Miller, Bradley Preece, Todd W. Du Bosq, and Kevin R. Leonard. A data-constrained algorithm for the emulation of long-range turbulence-degraded video. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXX*, page 110010J. International Society for Optics and Photonics, SPIE, 2019. 2

[48] N. G. Nair and V. M. Patel. Confidence guided network for atmospheric turbulence mitigation. In *IEEE International Conference on Image Processing*, pages 1359 – 1363, 2021. 1, 2, 8

[49] Nithin Gopalakrishnan Nair, Kangfu Mei, and Vishal M Patel. AT-DDPM: Restoring faces degraded by atmospheric turbulence using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3434–3443, 2023. 1, 2, 7, 8

[50] R. Nieuwenhuizen, J. Dijk, and K. Schutte. Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects. *EURASIP Journal on Image and Video Processing*, 2(2), 2019. 2

[51] R. J. Noll. Zernike polynomials and atmospheric turbulence. *Journal of Optical Society of America*, 66(3):207 – 211, 1976. 5

[52] O. Oreifej, X. Li, and M. Shah. Simultaneous video stabilization and moving object detection in turbulence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):450 – 462, 2013. 2

[53] Shyam Nandan Rai and C. V. Jawahar. Removing atmospheric turbulence via deep adversarial learning. *IEEE Transactions on Image Processing*, 31:2633 – 2646, 2022. 1, 2

[54] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161 – 4170, 2017. 4

[55] Michael C. Roggemann, Byron M. Welsh, Dennis Montera, and Troy A. Rhoadarmer. Method for simulating atmospheric turbulence phase effects for multiple time slices

and anisoplanatic conditions. *Applied Optics*, 34(20):4037 – 4051, 1995. 2

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241. Springer, 2015. 4

[57] A. Shteinman S. Gepshtein and B. Fishbain. Restoration of atmospheric turbulent video containing real motion using rank filtering and elastic image registration. In *Proc. European Signal Processing Conference*, pages 477 – 480, 2004. 2

[58] Seyed Morteza Safdarnejad, Xiaoming Liu, Lalita Udpa, Brooks Andrus, John Wood, and Dean Craven. Sports videos in the wild (SVW): A video dataset for sports analysis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1 – 7. IEEE, 2015. 5

[59] Mehul P Sampat, Zhou Wang, Shalini Gupta, Alan Conrad Bovik, and Mia K Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385–2401, 2009. 6

[60] J. D. Schmidt. *Numerical simulation of optical wave propagation: With examples in MATLAB*. SPIE Press, 2010. 2

[61] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2016. 6

[62] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9): 2035–2048, 2018. 6

[63] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8, 2008. 2

[64] UG2+. Bridging the gap between computational photography and visual recognition: $5th$ UG2+ prize challenge. http://cvpr2022.ug2challenge.org/dataset22_t3.html, 2022. Track 3. 6, 8

[65] Mikhail A. Vorontsov and Gary W. Carhart. Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images. *Journal of Optical Society of America A*, 18(6):1312 – 1324, 2001. 1, 2

[66] Mikhail A. Vorontsov and Valeriy Kolosov. Target-in-the-loop beam control: basic considerations for analysis and wave-front sensing. *Journal of Optical Society of America A*, 22(1):126 – 141, 2005. 2

[67] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *Proceedings of the AAAI conference on artificial intelligence*, pages 12216–12224, 2020. 6

[68] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 4

[69] Y. Xie, W. Zhang, D. Tao, W. Hu, Y. Qu, and H. Wang. Removing turbulence effect via hybrid total variation and deformation-guided kernel regression. *IEEE Transactions on Image Processing*, 25(10):4943 – 4958, 2016. 2

[70] Bindang Xue, Yi Liu, Linyan Cui, Xiangzhi Bai, Xiaoguang Cao, and Fugen Zhou. Video stabilization in atmosphere turbulent conditions based on the Laplacian-Riesz pyramid. *Optics Express*, 24(24):28092 – 28103, 2016. 2

[71] R. Yasarla and V. M. Patel. CNN-Based restoration of a single face image degraded by atmospheric turbulence. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):222 – 233, 2022. 1, 2

[72] Xingguang Zhang, Zhiyuan Mao, Nicholas Chimitt, and Stanley H. Chan. Imaging through the atmosphere using turbulence mitigation transformer. *IEEE Transactions on Computational Imaging*, 10:115–128, 2024. 1, 2, 4, 5, 6, 7, 8

[73] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 4

[74] Zhihang Zhong, Ye Gao, Yinqiang Zheng, Bo Zheng, and Imari Sato. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *International Journal of Computer Vision*, pages 1–18, 2022. 5, 6, 8

[75] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5

[76] Chao Zhu, Hang Dong, Jinshan Pan, Boyang Liang, Yuhao Huang, Lean Fu, and Fei Wang. Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3598–3607, 2022. 5, 6

[77] X. Zhu and P. Milanfar. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):157–170, 2013. 1, 2, 3