Subsampled Turbulence Removal Network

Wai Ho Chak[a], Chun Pong Lau[a], Lok Ming Lui,[a,*]

[a] *Department of Mathematics, The Chinese University of Hong Kong, Hong Kong*

Abstract

We present a deep-learning approach to restore a sequence of turbulence-distorted images from turbulent deformations and space-time varying blurs. Instead of requiring a massive training sample size in deep networks, we propose a training strategy that is based on a data augmentation method to model the turbulence from a relatively small dataset. Then we incorporate a subsampling method into the deep network to enhance the restoration performance of the trained **GAN** model. The contribution of the paper is threefold: First, we introduce a simple but effective data augmentation algorithm to model the turbulence for training in the deep network; Second, we propose the **W**asserstein **GAN** combined with the multiframe input and ℓ_1 cost for successful restoration of turbulence-corrupted video sequence; Third, we incorporate a subsampling algorithm into the deep network to filter out strongly corrupted frames so as to obtain an improved restored image.

Keywords: turbulence, data augmentation, subsamping, deep learning, **WGAN**

1. Introduction

The problem of image restoration from a sequence of frames under the atmospheric turbulence is challenging due to the dramatic downgrade in the image quality from the geometric distortions and the space-time varying blurs. Mul-

*Corresponding author

Email addresses: whchak@math.cuhk.edu.hk (Wai Ho Chak), cplau@math.cuhk.edu.hk (Chun Pong Lau), lmlui@math.cuhk.edu.hk (Lok Ming Lui,)

tiple factors such as temperature changes, air turbulent flow, densities of air particles, carbon dioxide level and humidity lead to the occurrence of several turbulence layers with various changes in the refractive index [1] [2]. These factors together explain the higher chance of obtaining corrupted video sequences in locations where the variation among these factors is large. In practice, either techniques in hardware-based adaptive optics [3] [4] or methods in image processing [5] [6] [7] [8] [9] are employed to remove the turbulence distortion in the images, but those prevailing models from either way can barely address to the majority of these factors.

Due to the fact that the atmospheric turbulence is complicated to be modeled, a deep learning approach which does not heavily require the underlying assumptions is more reasonable to tackle the problem than models relying on certain assumptions on the turbulence. We are thus motivated to investigate the possibility to remove geometric distortions and restore a good-quality image by using a generative model that does not explicitly take the above-mentioned factors into consideration. However, the unavailability of massive turbulence-distorted video frames disables the application of deep learning approaches to tackle the problem.

In this paper, we introduce a simple and yet effective data augmentation method to overcome the problem of data scarcity. The method models real turbulence with different deformations and different extent of blurs in order to provide sufficient training data. Since the artificial turbulence is randomly generated with different strength of deformations and blurs, a variety of turbulence-distorted videos can be produced from a single image. In general, it is known that the performance of image restoration is commensurate with the training sample size. Nevertheless, with the data augmentation method, the size requirement of the training data is not too restrictive and demanding in our proposed deep network to restore the turbulence-distorted images.

With the augmented training data, a deep network can be trained to solve the deturbulence problem. We propose a subsampled Wasserstein Generative Adversarial Network (**WGAN**) with multiframe input and ℓ_1 cost to simultane-

ously remove geometric distortions and blurring effects of turbulence-distorted image sequences. **WGAN** is known for its effectiveness in generating a clear image from noises. Together with the ℓ_1 cost applied to the network, important features of the images can be restored even though they are corrupted. To gather enough information, it is natural to take multiple frames from the video as the input of the turbulence-removal network. Using multiple frames as input is essential to obtain a clear image from a turbulence-distorted video.

In the testing stage, we propose to incorporate a subsampling algorithm to the trained network for better performance. Usually, turbulence-distorted video consists of mildly distorted frames. The subsampling method extracts those sharp and mildly distorted frames in order to achieve an even better restoration result. We experimentally show that by incorporating the subsampling method, the performance of removing geometric distortions and blurs of the degraded images can be significantly improved.

1.1. Contributions

The main contributions of this paper are listed as follows:

1 We propose a deep-learning approach, which is a **WGAN** model with the multiframe input and ℓ_1 loss, for the restoration of turbulence-distorted images. To the best of our knowledge, it is the first work to study the feasibility of applying deep convolutional neural network for solving the deturbulence problem to simultaneously remove geometric distortions and space-time varying blurs.

2 We propose a data augmentation method to generate geometrically distorted and blurry images for training. It overcomes the problem of data scarcity. As such, the use of deep learning approaches to tackle the deturbulence problem is made possible that a sufficiently large dataset is not required.

3 We propose to incorporate a subsampling method into the trained network to obtain a better restored image. Experimental results demonstrate that

the performance of the proposed model can be significantly improved with the subsampling strategy.

2. Related Work

2.1. Restoration of turbulence-distorted images

The main tasks of restoring turbulence-corrupted images consist of the removal of geometric distortions and space-time varying blurs. It is in general challenging to discard both the geometric distortions and blurs simultaneously. Several previous works are firstly devoted to reconstruct a clean image through the process of image fusion by registering the image frames to a good reference image. Meinhardt-Llopis and Micheli [10] [11] proposed a reference extraction method by registering frames to a 'centroid' image. The basic idea is to warp each image frame by the average deformation field between it and the other images from the turbulence-degraded video. This method has an assumption that the deformation between the original image and the distorted frames is zero on average. However, the estimated movements of individual pixels can sometimes be much larger, and the mean displacement of each pixel may deviate more significantly from zero in a real turbulence-distorted video. It may pose a challenge for the centroid method to remove all the geometric distortions. Another approach to obtain a clear reference image is done by selecting a "lucky frame", which is the sharpest frame from a distorted video [12]. This method is motivated and supported by the statistical proofs [13] that show a high probability of extracting video frames with sharp texture details given a sufficient amount of frames. Nevertheless, in many situations, getting a frame that is entirely sharp everywhere is difficult. To alleviate this issue, the Lucky-Region method has been proposed by Aubailly *et al.* [14], which chooses the sharpest patch from each frame and combine them afterward. Motivated by this patch-wise sharpness selection method, another approach introduced by Anantrasirichai *et al.* [15] suggests to having a frame selection prior to registration. A composite cost function was introduced, and the selection was done in one step by sorting.

However, some of the selected frame may geometrically differ significantly from the reference image. On the other hand, the cost function assumes the reference image given by the temporal intensity mean over all frames can accurately approximate the underlying true image, which is usually not the case. Similarly, a subsampling method introduced by Roggemann [16] selects subsamples from images produced by adaptive-optics systems to generate a temporal mean with higher signal-to-noise ratio.

To enhance the accuracy of the registration onto a reference image, a feasible approach is to stabilize the video and reduce the deformation between each frame and the reference image. The SGL method purposed by Lou *et al.* [17] incorporates Sobolev gradient and Laplacian for stabilization of the video sequence, and finds the latent image by the Lucky-Region method.

Robust Principle Component Analysis (RPCA) [18] is a recent approach to solve the deturbulence problem. Low-rank decomposition method proposed by He *et al.* [19] decomposes the video sequence into the low-rank and sparse parts. A variational approach introduced by Xie *et al.* [20] is applied to improve the initial reference image as the low-rank image that captures the texture information and suppresses geometric distortions, although it usually looks blurry. Registration may sometimes fail when there is a large deformation between the observed video frames and the reference image.

Another recent approach is the joint subsampling and reconstruction variational model proposed by Lau *et al.* [21]. An advantage of the model is that there is no registration involved during the subsampling and reconstruction processes, and hence it is computationally efficient. Using the proposed energy model with various fidelity terms, restoration of turbulence-distorted images of different degrees of distortions can be achieved.

2.2. Generative Adversarial Networks

Generative adversarial networks (**GAN**s) firstly proposed by Goodfellow at al. [22] defines two separated competitors: the generator G_θ and the discriminator D_ξ. The generator is designed to produce samples from noise $\mathcal{Z}$ while

a discriminator is designed to distinguish real sample y_i and generated sample $G_\theta(z_i)$. The main objective of the generator is to generate perceptually persuasive samples that are challenging to be discriminated by the real samples. The competition between the generator G and the discriminator D can be described by the minimax objective shown as follows:

$$\min_G \max_D \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[\log(1-D(\tilde{x}))] \tag{1}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the generated distribution given by $\tilde{x} = G(z), z \sim P(z)$, where z is sampled from a noise distribution. The advantage of **GAN**s is the ability to generate clear samples with high perceptual quality. However, as described by Salimans *et al.* [23], there are undesirable issues such as vanishing gradients and mode collapse in the training. The difficulties can be explained by the fact that minimizing the objective function for **GAN**s is equivalent to minimizing the Jensen-Shannon divergence, which is locally saturated and results in vanishing gradients, between the data and model distributions.

Later, Arjovsky *et al.* [24] addressed the gradient vanishing problem by introducing the weaker Wasserstain-1 distance $W(\mathbb{P}_r, \mathbb{P}_g)$ which gives clear gradients almost everywhere in the **GAN** model. The competition between the two networks is reformulated as our minimax optimization objective:

$$\min_G \max_{D\in\mathcal{D}} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{2}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions such that $\|D\|_L \leq 1$. The original Lipschitz constraint enforcement proposed by Arjovsky at al. [24] is weight clipping to $[-c, c]$. Another approach proposed by Gulrajani *et al.* [25] is adding the gradient penality term

$$\lambda \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2] \tag{3}$$

The approach does not require hyperparameter tuning and is robust to the selection of the architecture of generator. In contrast to the conventional convolutional neural network, **GAN**s can generate clearer images. **WGAN**-ℓ_1 proposed by Kupyn *et al.* [26] has been shown effective in image deblurring.

3. TRN: Turbulence Removal Network

In this section, we describe our proposed method based on deep convolutional neural network, namely, the **T**urbulence **R**emoval **N**etwork (**TRN**). Figure 3 shows that whole network architecture for both the generator network G and the critic network D.

3.1. Data Augmentation for Turbulence

The first step of our proposed algorithm is to synthesize sufficient training data distorted by turbulence. A large sample size of training data is typically necessary for solving tasks using deep learning approaches, but unfortunately there is limited turbulence-distorted data available. This hinders the application of deep learning approaches for turbulence removal. To alleviate this issue, we introduce a new, simple but effective method for generating the training data from a few data for deep learning.

More specifically, a single (clean) frame I is transformed to another image I^t with geometric distortions and blurs as follows. We first select $Q = (w - 2N) \times (w - 2N)$ pixel positions randomly, where w and h are the width and height of the image respectively. At each randomly selected pixel position $(x, y) \in \mathcal{R}$, we consider a $N \times N$ local patch $P^N_{x,y}$ around the pixel. A motion vector field $V_{x,y} = (u, v)$ is then generated in $P^N_{x,y}$. For each $p \in P^N_{x,y}$, the vector $(u(p), v(p))$ is sampled from a normal distribution, smoothened by a Gaussian kernel and entry-wisely multiplied by a strength value of distortion. Mathematically, the vector field $V_{x,y}$ can be written as:

$$V_{x,y} = S\ (G_\sigma * \mathcal{N}_1, G_\sigma * \mathcal{N}_2), \tag{4}$$

where G_σ is the Gaussian kernel with standard deviation σ, S is the strength value, $\mathcal{N}_1$ and $\mathcal{N}_2$ are randomly selected from a normal distribution. $V_{x,y}$ is then extended to the whole image domain by setting zero outside $P^N_{x,y}$. It is then employed to wrap the original image I to get a transformed image. We repeat this process by M iterations.

Essentially, the overall motion vector field $V = (u, v)$ after M iterations is defined by fusing the vector patches together wherever overlapping. Mathematically,

$$V = \sum_{(x,y)\in\mathcal{R}} V_{x,y}, \tag{5}$$

where $\mathcal{R}$ is the collection of Q randomly selected pixel positions.

We denote the transformed image by I^t. The transformed image I^t is further blurred by a Gaussian kernel $w(n) = \exp(-\frac{n^2}{2B^2})$. The parameter B is sampled uniformly from $[0.1, 1]$. The final transformed image with geometric distortions and blurs is given by $I^i = w * I^t$. Using the proposed algorithm with randomized parameters, an original clean image I^C is transformed into a sequence of images $\{I_1, ..., I_n\}$ with geometric distortions and blurs for training. Figure 2 shows some of the transformed images with different strength values. Experimental results suggest that this proposed algorithm with random parameters can successfully cover most possible deformations and hence geometric distortions can be successfully learnt from the deep network.

The data augmentation algorithm to synthesize turbulence-distorted video frames for training is summarized in Algorithm 1. Figure 1 illustrates the overall procedure of the data augmentation algorithm.

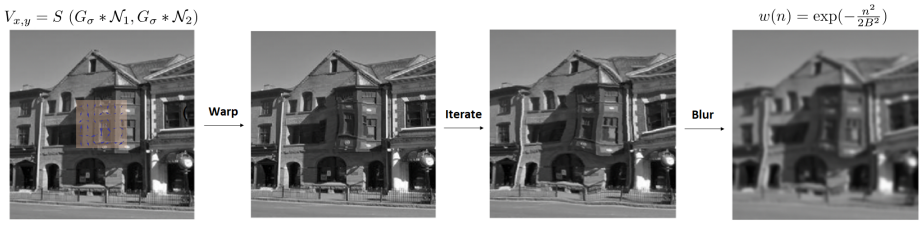
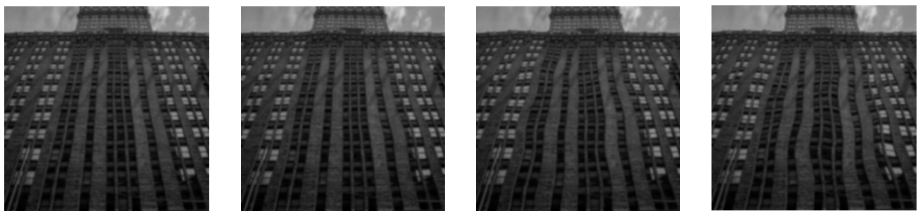


Figure 1: The overall procedure of the data augmentation algorithm.

3.2. **WGAN**$-\ell_1$ *with Multiframe Input*

The proposed turbulence removal network (**TRN**) is a multi-frame subsampled **WGAN** with the ℓ_1 cost incorporated into the model. Multiframe input is



(a) S = 0.1 (b) S = 0.2 (c) S = 0.3 (d) S = 0.4

Figure 2: Generation of turbulence-distorted frames with different distortion strength S and fixed blur constant $B = 1$.

adopted in **TRN** to absorb sufficient information on the turbulence deformation of the original image. Then, **TRN** is trained to remove geometric distortions and blurs with the **WGAN** architecture. The additional ℓ_1 cost attempts to retain the important textures of the original image.

3.2.1. Multiframe Input

We first discuss the input of our proposed **TRN**. The conventional input for **GAN**s is a noise vector $z \in \mathbb{R}^N$ randomly generated according to the normal distribution. Then the noise vector z is transformed into the desired output through the generator. Our network is similar to DeBlurGan [26]. The architecture of DeBlurGan [26] requires blurred image as an input and produce a deblurred image. Although blur is one of the consequences of turbulence observed in the frames, a single frame from a turbulence-distorted video as an input is experimentally shown to be ineffective in recovering the original image. Therefore, using the original architecture from DeBlurGan is insufficient to remove undesirable effects such as the geometric distortions. Motivated by this observation, the input in our network is a turbulence-distorted multiframe input originated from a clear image. Thus, the improved version of our new architecture is to include multiple frames as the input. Instead of taking the whole video sequence as the input, subsampled frames are selected.

Algorithm 1 Distortion and Blur Generation

Parameters:

$M = 1000$ - number of iterations

$N = 32$ - patch size

μ - mean of the Guassian kernel

σ - standard deviation of the Guassian kernel

S - distortion strength, uniform from [0.1,0.4]

B - blur constant, uniform from [0.1, 1]

procedure DISTORTBLUR (IMG, σ, N, M)

  Create a Gaussian kernel from Normal CDF $(0.2 * \text{rand} - 1, \sigma)$

  for $i = 1 \rightarrow M$ **do**

    $x \leftarrow \text{randi}(\text{width} - 2 * N) + N$

    $y \leftarrow \text{randi}(\text{height} - 2 * N) + N$

    $u(x - N : x + N, y - N : y + N)$

      $\leftarrow u(x - N : x + N, y - N : y + N) + \text{randn} * S$

    $v(x - N : x + N, y - N : y + N)$

      $\leftarrow v(x - N : x + N, y - N : y + N) + \text{randn} * S$

    Convolve the u, v vector fields with the kernel

  Wrap the image with u, v vector fields with the kernel.

  Blur the image by convoluting it with a Gaussian smoothing window $w(n) = \exp(-\frac{n^2}{2B^2})$

return Distorted Video Frames

With the data augmentation method described in last subsection, the training data is a multiple frames $I_{TD} = (I_{TD}^{(1)}, I_{TD}^{(2)}, ..., I_{TD}^{(n)})$ transformed from the original clean image I_C of size $r \times s$. In **TRN**, the input is a selected subsampled frames from I_{TD}. Instead of using the whole sequence of frames as the input, we randomly select $m = 20$ frames from the whole video in the training stage as the input for the generator in the GAN model. In the testing stage, we incorporate a subsampling method [21] to select the most useful frames as the input. The incorporating of the subsampling method into the network is shown

to be effective in obtaining a significantly better restored image.

We now describe the subsampling method we incorporate in the network in details. Given a turbulence-distorted video frames $\mathcal{I}_{TD} = (I_{TD}^{(1)}, I_{TD}^{(2)}, ..., I_{TD}^{(n)})$, we consider a variational model to get an optimal subsample set $\mathcal{J}$ of sharp and mildly distorted images. $J = \{i_1, \cdots, i_m\}$ is the index set of the subsample set $\mathcal{J}$, where $m = |J|$ is the number of chosen video frames in the subsample. Simultaneously, we obtain a reference image I_R from the subsample set J. The variational model is formulated in the following form:

$$E(I_R, J) = \frac{1}{|J|}\left(\sum_{k\in J} \mathcal{F}(I_R, I_{TD}^{(k)}) + \lambda \mathcal{Q}(I_{TD}^{(k)})\right) - \tau \mathcal{R}(J) \tag{6}$$

The fidelity term $\mathcal{F}$ is the discrepancy term between the reference image and the video frames. In our model, we define $\mathcal{F}(I_R, I_{TD}^{(k)}) = \left\| I_R - I_{TD}^{(k)} \right\|_2^2$ for measuring the ℓ_2 distance between the reference image I_R and the subsampled video $\{I_{TD}^{(k)}\}_{k\in J}$. The quality term $\mathcal{Q}(I_{TD}^{(k)})$ for each video frame $I_{TD}^{(k)}$ is based on the normalized version of $\|\Delta I_k\|_1$:

$$\mathcal{Q}(I_{TD}^{(k)}) = \frac{\max\limits_{1\le i\le n} \|\Delta I_{TD}^{(i)}\|_1 - \|\Delta I_{TD}^{(k)}\|_1}{\max\limits_{1\le i\le n} \|\Delta I_{TD}^{(i)}\|_1 - \min\limits_{1\le i\le n} \|\Delta I_{TD}^{(i)}\|_1} \tag{7}$$

The term $\Delta I_{TD}^{(k)}$ is the convolution of $I_{TD}^{(k)}$ with the Laplacian kernel specifically highlighting the edges and features of objects in the image $I_{TD}^{(k)}$. The sharper the image $I_{TD}^{(k)}$, the higher the magnitude of $\Delta I_{TD}^{(k)}$. As a consequence, the normalized quality measure $\mathcal{Q}(I_{TD}^{(k)})$ is smaller when $I_{TD}^{(k)}$ is sharp. The term λ in the energy model $E(I_R, J)$ is a positive constant to quantify the importance of sharpness of the frame $I_{TD}^{(k)}$. The regularization term $\mathcal{R}$ is the concave increasing function $1 - e^{-\rho|J|}$, where $\rho > 0$ is a constant to quantify the importance of the number of selected frames. The function is chosen in order to acquire more information from additional video frames, whereas the effect on the quality of the reference image I_R is reduced with a marginal increase in the subsample size. The detailed formulation of the variational model is described in [21]. An alternating minimization strategy can be used to solve the model, which is described in Algorithm 2.

Algorithm 2 Image Subsampling

Parameters:

λ - sharpness parameter

τ - subsample size parameter

ρ - subsample decay rate parameter

procedure IMAGE SUBSAMPLING ($\mathcal{I}_{TD} = (I_{TD}^{(1)}, I_{TD}^{(2)}, \cdots, I_{TD}^{(n)})$, λ, τ, ρ)

  Compute $I_R^0 = \frac{1}{n}\sum_{i=1}^{n} I_{TD}^{(i)}$

  Compute the quality measure $\mathcal{Q}(I_{TD}^{(k)})$ for each video frame $\{I_{TD}^{(k)}\}_{k=1}^{n}$

  repeat

    Given J^{t-1}, I_R^{t-1}. Fixing I_R^{t-1}, solve

$$J^t = \arg\min_J \frac{1}{|J|}\left(\sum_{k\in J}\left\|I_R^t - I_{TD}^{(k)}\right\|_2^2 + \lambda\mathcal{Q}(I_{TD}^{(k)})\right) - \tau\left(1 - e^{-\rho|J|}\right)$$

    Compute $E_{1,k} = \left\|I_R^t - I_{TD}^{(k)}\right\|_2^2 + \lambda\mathcal{Q}(I_{TD}^{(k)})$ for each k and arrange $E_{1,k}$ in ascending order.

    Compute the sum S_j for each j and arrange S_j in ascending order.

    $J^t \to \{k_1, k_2, \cdots, k_{j_1}\}$

    Fixing J^t, solve

$$I^t = \arg\min_I \frac{1}{|J^t|}\left(\sum_{k\in J^t}\left\|I_R - I_{TD}^{(k)}\right\|_2^2\right)$$

    $I_R^t \to \frac{1}{|J^t|}\sum_{k\in J^t} I_{TD}^{(k)}$

  until $E_1^{t-1} - E_1^t \le \epsilon$

return subsampled image sequence $\{I_{TD}^{(k)}\}$

3.2.2. U-Net Architecture for Generator Network

The generator network G we use is the U-Net [27], which consists of five types of layers: convolutional layer, deconvolutional layer, max-pooling layer, Randomized Leaky ReLU activation layer ($\alpha = 0.2$) [28] and instance normalization layer [29]. U-Net is known to involve a contracting path for contextual preservation and a symmetric expanding path for localization, and hence was

particularly successful in image segmentation, denoising and super-resolution. Thus, U-net is used as the main architecture for our generator network G.

The subsampled turbulence-distorted multiframe passes through 7 blocks of convolutional layers and 6 blocks of deconvolutional layers to generate a clear image. The first 7 blocks $B_C^{(1)}, B_C^{(2)}, ..., B_C^{(7)}$ contains convolutional layers, followed by the 6 remaining blocks $B_D^{(1)}, B_D^{(2)}, ..., B_D^{(6)}$ consisting of deconvolutional layers. Each block $B_C^{(i)}$ contains convolutional layers, non-linear activation layers and instance normalization layers. The temporal features extracted in each block are down-sampled by max-polling except for the features of the last block $B_C^{(7)}$. The features in $B_C^{(6)}$ and $B_C^{(7)}$ are concatenated before passing through the block $B_D^{(1)}$ in order to retain the deep features without too much information loss. The feature collected in the first block $B_D^{(1)}$ is then concatenated with the feature from the block $B_C^{(5)}$ to output the feature in the second block $B_D^{(2)}$. Repeating the process, we obtain a clear image I_C which is of the same size as the original undistorted image.

The generator network is not pre-trained, since the input of the architecture is different from the conventional one. The conventional model takes the three channels of the image as input. In our case, we have a subsample of turbulence-distorted video frames $\mathcal{J}$, which are randomly chosen in the training stage and selected by the subsampling method introduced in the last subsection in the testing stage. The generator network G is trained after the critic network D is trained multiple times to give a clearer image $I_{TD} = G(I_{TD}^{(i_1)}, ..., I_{TD}^{(i_m)})$. The loss function of the generator network for removing geometric distortion and blurs is defined by

$$L_G = -\sum_{n=1}^{N} D(I_{TD}) + \frac{\gamma}{N} \|I_C - I_{TD}\|_1 \tag{8}$$

The first term in the loss function L_G is the adversarial loss that encourages solutions to reside on the manifold of natural images. In order to retain the textures inherited from the turbulence-distorted video frames, we further incorporate the ℓ_1 loss into the loss function L_G. Note that the combination of the pixel-wise error term with the adversarial loss has an advantage. It was suggested that the

minimization of the loss function that contains only the pixel-wise error term, such as the ℓ_1 or ℓ_2 error, is insufficient to produce a clear image [30]. Besides, the ℓ_2 error term can often cause image blur. Thus, we employ ℓ_1 error, instead of the ℓ_2 error, in our loss function L_G to make the image much less blurry. Experimental results demonstrate that the combination of the two terms in the loss function L_G can effectively remove geometric distortions and undesirable artifacts, such as image blurs.

3.2.3. Critic Network

The critic network D in the **WGAN** [24] is a deep **CNN** involving convolutional layers, fully-connected layer, ReLU activation layer [31] and instance normalization layer [29]. We denote the first 6 convolutional layers by $L^{(1)}, L^{(2)}, ..., L^{(6)}$ and the last fully connected layer by $L^{(7)}$. The critic values $D \circ G(I_{TD}))$ and $D \circ I_C$ are passed into the critic network to output the Wasserstain-1 distance

$$\max_{\|D\|_L \leq 1} \quad \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r}[D(x)] - \mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] \tag{9}$$

The critic network D is trained till optimal before updating the generator network G. The loss function of the critic network D in the training process is provided as follows:

$$L_D = D(G(I_{TD})) - D(G(I_C)) + \lambda\left(\left\|\nabla D\Big(\alpha I_C + (1-\alpha) I_{TD}\Big)\right\|_2 - 1\right)^2, \tag{10}$$

where α is a randomly generated number from uniform distribution $U[0, 1]$. Since there is no pre-trained model involved in the generator network G, the whole training takes a long time and the loss blows up. In order to further enforce the 1-Lipschitz assumption in the critic network D, we further impose the weight constraint λ. The weight is clipped in the interval $[-c, c]$. Together with this weight constraint, the training becomes more stable.

4. Experiments

4.1. Dataset

The dataset for training is collected from Flicker. It consists of 1500 images of buildings and 1000 images of chimneys. All collected images are resized to

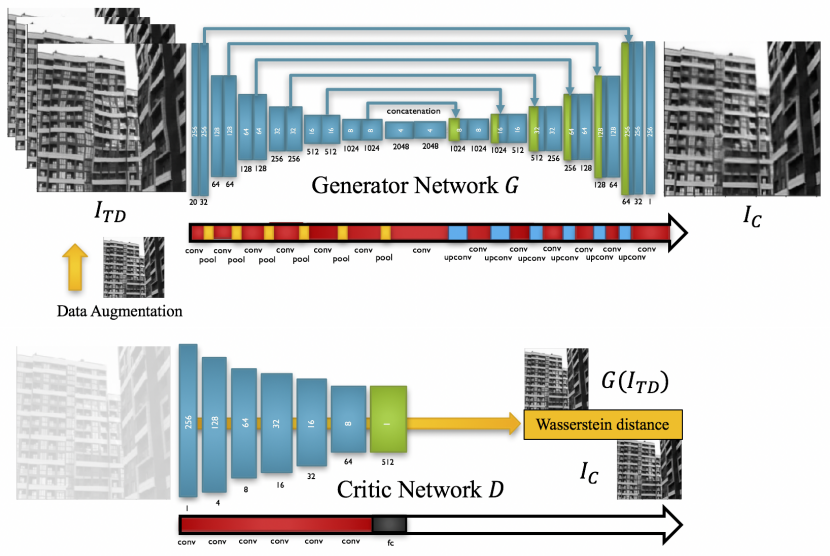


Figure 3: The generator network G has the U-net architecture, and the critic network D is the conventional convolutional neural network. The subsampled frames are concatenated before passing through the generator network.

256x256 and are synthetically deformed by our data augmentation algorithm. More specifically, each image is deformed to produce 100 deformed video sequences. Therefore, the whole dataset is enlarged by a factor of 100. We test the trained network on more than 400 testing data, which are different from the training dataset. The testing dataset consists of simulated video sequences as well as real turbulence-distorted video sequences.

4.2. Training Details

The experiments are conducted in PyTorch [32] with a CUDA-enabled GPU. The data augmentation algorithm is carried out in Matlab before the deep learning process is conducted. The strength value of distortion S and the blurring parameter B are randomly sampled from $[0.1, 0.4]$ and $[0.1, 1]$ respectively. Adam solver [33] is used for the gradient descent with a learning rate of 10^{-4}, $\beta_1 = 0.5$ and $\beta_2 = 0.99$ for both the generator G_θ and the critic D_ξ. We set 3 gradient descent steps for D_ξ and then 1 step for G_θ. We also apply the instance normal-

ization and dropout to improve the training. In addition to the gradient penalty term [25], we enforce the parameters ξ in the range $[-0.01, 0.01]$. For each epoch, we train both the network with batch size of 1, and set $\lambda = 10, \gamma = 1000$. Furthermore, we randomly select 20 frames from the video sequence as our input. The whole training process for 40 epochs takes around 3 days. Figure 4 demonstrates that the restoration performance is gradually better in the training. The network firstly discards geometric distortion from the turbulence in the first few epochs and then attempts to deblur and preserve the texture of the original image in the remaining epochs.

Figure 4: The training starting from the 1st epoch (left) to the 9th epoch (right). Each displayed image from its left image, except the first one, are generated for 1-2 epochs. The training performance is gradually improved.

5. Result

After the **TRN** is trained, we test its performance on more than 400 testing data consisting of simulated and real turbulence-distorted videos. The testing data are different from the training data. In this section, we report some of the experimental results.

5.1. Restoration of simulated turbulence-distorted videos

Figure 5 and 6 show the restoration results of some simulated turbulence-distorted image sequences capturing different buildings. The first column shows the observed frames from each turbulence distorted image sequences, which are degraded by both geometric distortions and blurs. The middle column shows

the restoration results using the **TRN** without the incorporation of the subsampling method. Note that most geometric distortions and blurs are removed, although some amount of distortions can still be observed. The right column shows the restoration results using the **TRN** with the incorporation of the subsampling method. With subsampling, the geometric distortions and blurs can be removed more successfully. The restoration results are more satisfactory compared to those without subsampling. It demonstrates the incorporation of the subsampling method into the deep network is beneficial.

We have test our deep network on some 'chimney' image sequences. Figure 7 and 8 shows the restoration results of some simulated turbulence-distorted image sequences capturing different chimneys. Again, the first column shows the observed frames from each turbulence distorted image sequences, which are degraded by both geometric distortions and blurs. The middle column shows the restoration results using the **TRN** without the incorporation of the subsampling method. The right column shows the restoration results using the **TRN** with the incorporation of the subsampling method. Again, with the incorporation of the subsampling method, the geometric distortions and blurs can be removed more successfully. The restoration results are more satisfactory compared to those without subsampling. It again demonstrates the benefit of incorporating the subsampling method into the deep network.

To test the performance of our trained network to handle general large deformations, we randomly generate geometric distortions of an original image using large quasi-conformal deformations. More specifically, we randomly select some pixel positions in the image domain. A patch-wise triangular mesh is formed with the chosen position as the center. The method we use to generate artificial turbulence is deformation using Laplace-Beltrami solver (LBS) [34]. We propose to assign the Beltrami coefficient μ which is a measure of nonconformality on each face vertex as follows:

$$\mu = \left[(0.6+\epsilon_1)\cos\left((4+\epsilon_2)\pi x^2\right)\right] + \left[(0.6+\epsilon_3)\sin\left((6+\epsilon_4)\pi y^2\right)\cos\left((8+\epsilon_5)\pi xy\right)\right] i,$$

where ϵ_i are numbers randomly chosen in the range $[0, 0.3]$ for $i = 1, 2$ and $[-1, 1]$

Figure 5: Restoration of turbulence-distorted 'building' images. Column (a) shows the observed frames from each video. Column (b) shows the restoration results using the proposed **TRN** without subsampling. Column (c) shows the restoration results using **TRN** with subsampling.

Figure 6: Restoration of another set of turbulence-distorted 'building' images. Column (a) shows the observed frames from each video. Column (b) shows the restoration results using the proposed **TRN** without subsampling. Column (c) shows the restoration results using **TRN** with subsampling.

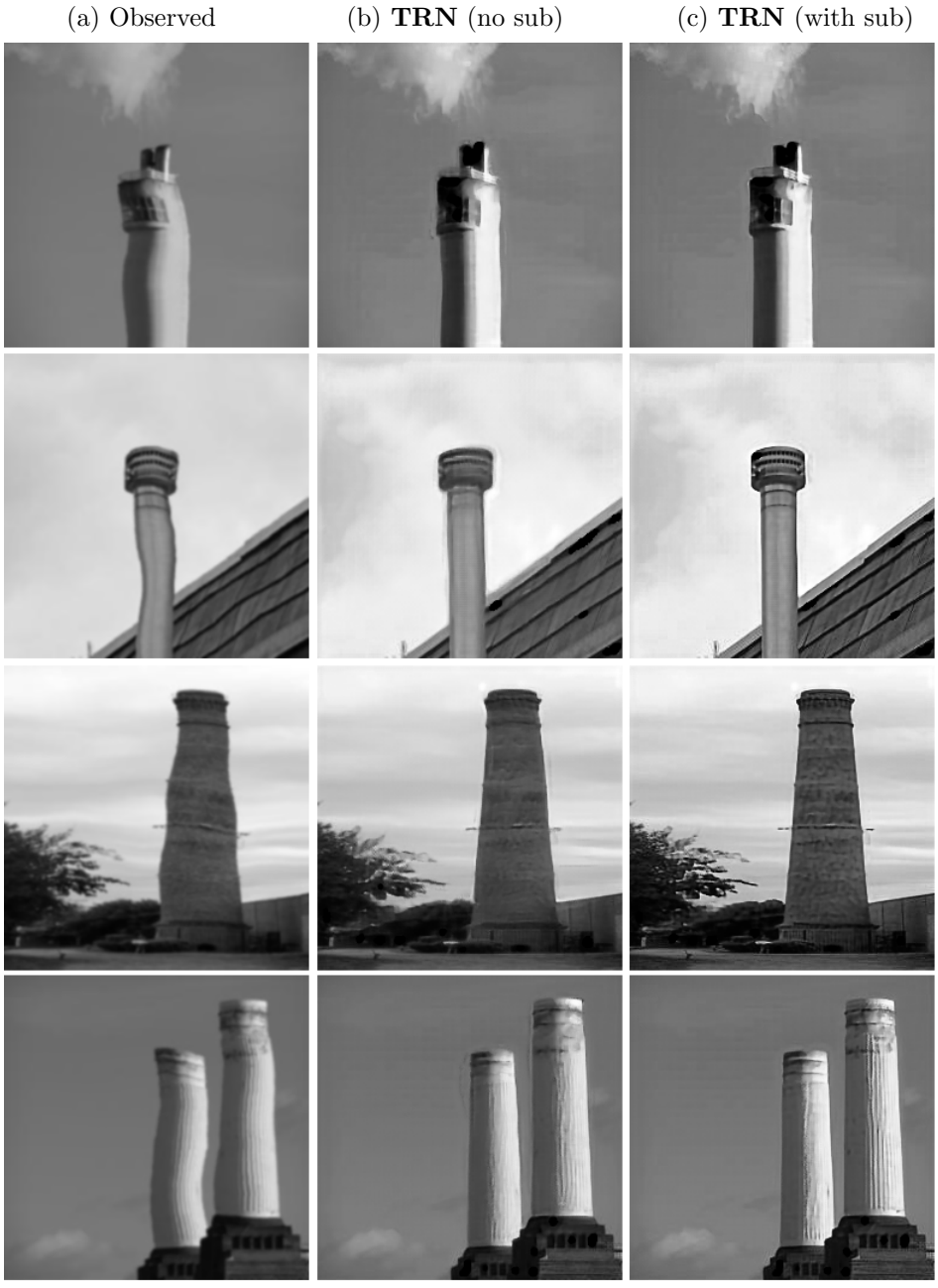


Figure 7: Restoration of turbulence-distorted 'chimney' images. Column (a) shows the observed frames from each video. Column (b) shows the restoration results using the proposed **TRN** without subsampling. Column (c) shows the restoration results using **TRN** with subsampling.

Figure 8: Restoration of another set of turbulence-distorted 'chimney' images. Column (a) shows the observed frames from each video. Column (b) shows the restoration results using the proposed **TRN** without subsampling. Column (c) shows the restoration results using **TRN** with subsampling.

for $i = 3, 4, 5$. Then we obtain the deformation field by using the LBS solver, and wrap the image. By introducing image blurs to each deformed images, we obtain an image sequence with large geometric distortions and blurs. Note that the quasi-conformal deformations have never been seen in the training process. Our aim is to investigate whether the trained deep network can deal with general deformations. The experimental results are shown in Figure 9. In the Figure 9, the first column shows the observed frames from each distorted image sequences with large quasi-conformal deformations. The second column shows the restored images using **TRN**. The geometric distortions and blurs are successfully removed. These results show that the **TRN** can effectively handle general large deformations.

5.2. Comparison with other methods

We also compare our proposed deep-learning based algorithms with other existing methods, namely, the **SGL** method [17] and the **IRIS** method [21]. Some experimental results are shown in Figure 10. In the Figure 10, the first column shows the restoration results of some turbulence-distorted image sequences using **TRN**. The second column shows the restoration results using **SGL**. The last column shows the restoration results using **IRIS**. The restoration results using **SGL** is generally blurry and geometrically distorted. The results restored by **IRIS** have less geometric distortions and blurs, although some geometric deformations can still be visualized. In general, **TRN** gives the best restoration results with least geometric distortions and blurs. These visual results are also validated quantitatively using PSNR and SSIM, as reported in Table 1.

5.3. Restoration of real turbulence-distorted videos

We also test the **TRN** on real turbulence-distorted videos that do not have a clear ground-truth image. Figure 11 shows the restoration results of a real 'chimney' turbulence-distorted image sequence. (a) shows an observed frame from the image sequence. (b) shows the restoration results using **TRN** without subsampling. Most geometric distortions and blurs are suppressed. (c) shows

(a) Observed (b) Restored by **TRN**

Figure 9: Restoration of image sequence distorted by large quasi-conformal deformations. Column (a) shows the observed frames from each video. Column (b) shows the restoration results using the proposed **TRN**.

(a) Restored by **TRN** (b) Restored by **SGL** (c) Restored by **IRIS**

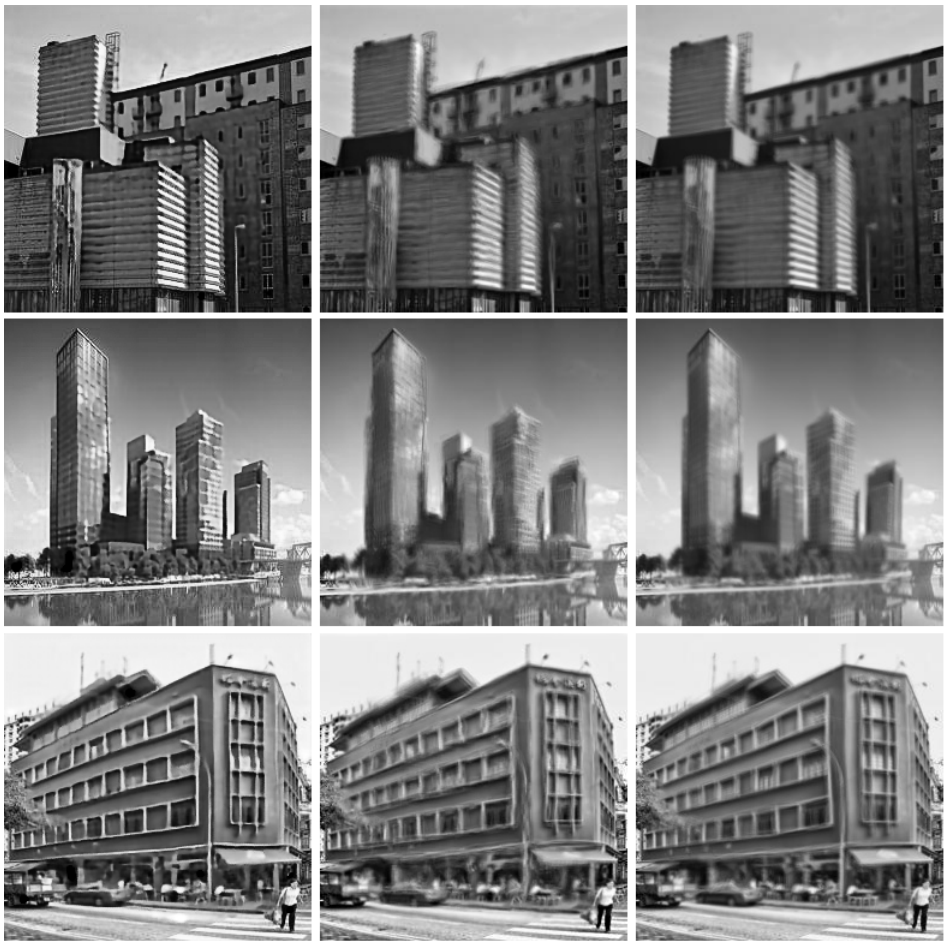


Figure 10: Comparison between **TRN**, **SGL** and **IRIS** on 'building 1', 'building 2' and 'building 3'. (a) shows the restoration results by **TRN**. (b) shows the restoration results by **SGL**. (b) shows the restoration results by **IRIS**

.

	PSNR			SSIM		
	TRN	**SGL**	**IRIS**	**TRN**	**SGL**	**IRIS**
building 1	21.3	19.3	20.1	0.838	0.697	0.749
building 2	23.7	22.1	23.0	0.808	0.714	0.740
building 3	24.7	23.8	24.5	0.836	0.770	0.793

Table 1: PSNR and SSIM of the restored images using different deturbulence models.

the restoration results using **TRN** with subsampling. With subsampling, the results are more satisfactory compared to those without subsampling. It again demonstrates the effectiveness of incorporating the subsampling model into the deep network.

Figure 12 shows the restoration results of another real turbulence-distorted image sequence capturing a building. Again, (a) shows an observed frame from the image sequence. (b) shows the restoration results using **TRN** without subsampling. (c) shows the restoration results using **TRN** with subsampling. As before, with subsampling, the results are more satisfactory than those without subsampling.

(a) Observed (b) **TRN** (no sub) (c) **TRN** (with sub)

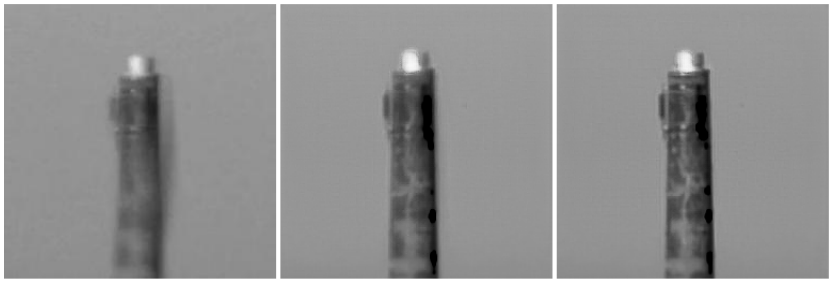


Figure 11: Restoration of real turbulence-distorted image sequence capturing a chimney. (a) shows an observed frame from the image sequence. (b) shows the restored image using **TRN** without subsampling. (c) shows the restored image using **TRN** with subsampling.

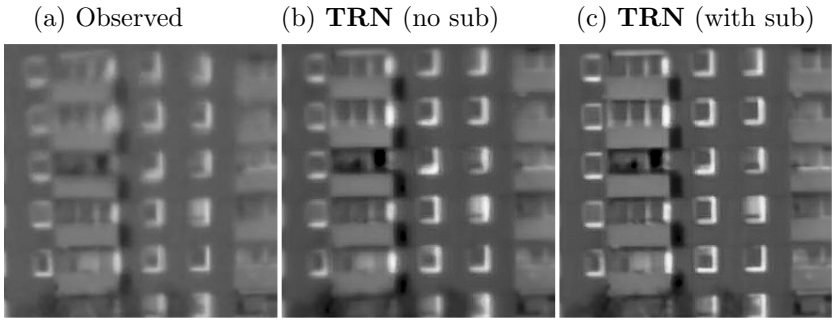


Figure 12: Restoration of real turbulence-distorted image sequence capturing a building. (a) shows an observed frame from the image sequence. (b) shows the restored image using **TRN** without subsampling. (c) shows the restored image using **TRN** with subsampling.

6. Conclusion

We introduce the turbulence removal network (**TRN**), which is a Generative Adversarial Network (**GAN**) incorporated with ℓ_1 objective function, to suppress geometric distortions as well as removing blurs of image sequences distorted by turbulence. Although there is only a limited amount of available data corrupted by real turbulence, we proposed a data augmentation method to synthetically generate turbulence-distorted image frames for training. A subsampling method is further incorporated into the trained network to obtain an improved restoration result. Extensive experiments have been carried out to test the deep network, which demonstrates the effectiveness of the proposed model to restore turbulence-distorted images. In the future, we will explore the possibility to develop a turbulence removal network to restore turbulence-distorted video with moving objects.

Acknowledgment

We would like to thank Mr. M. Hirsch and Dr. S. Harmeling from Max Planck Institute for Biological Cybernetics for sharing the real chimney and building video sequence. Lok Ming Lui is supported by HKRGC GRF (Project ID: 402413).

References

[1] R. Hufnagel, N. Stanley, Modulation transfer function associated with image transmission through turbulent media, JOSA 54 (1) (1964) 52–61.

[2] M. C. Roggemann, B. M. Welsh, B. R. Hunt, Imaging through turbulence, CRC press, 2018.

[3] J. E. Pearson, Atmospheric turbulence compensation using coherent optical adaptive techniques, Applied optics 15 (3) (1976) 622–631.

[4] R. Tyson, Principles of adaptive optics, CRC press, 2010.

[5] M. Shimizu, S. Yoshimura, M. Tanaka, M. Okutomi, Super-resolution from image sequence under influence of hot-air optical turbulence, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[6] S. M. Seitz, S. Baker, Filter flow, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 143–150.

[7] D. Li, R. M. Mersereau, S. Simske, Atmospheric turbulence-degraded image restoration using principal components analysis, IEEE Geoscience and Remote Sensing Letters 4 (3) (2007) 340–344.

[8] M. A. Vorontsov, Parallel image processing based on an evolution equation with anisotropic gain: integrated optoelectronic architectures, JOSA A 16 (7) (1999) 1623–1637.

[9] M. Hirsch, S. Sra, B. Schölkopf, S. Harmeling, Efficient filter flow for space-variant multiframe blind deconvolution, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 607–614.

[10] E. Meinhardt-Llopis, M. Micheli, Implementation of the centroid method for the correction of turbulence, Image Processing On Line 4 (2014) 187–195.

[11] M. Micheli, Y. Lou, S. Soatto, A. L. Bertozzi, A linear systems approach to imaging through turbulence, Journal of mathematical imaging and vision 48 (1) (2014) 185–201.

[12] M. A. Vorontsov, G. W. Carhart, Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images, JOSA A 18 (6) (2001) 1312–1324.

[13] D. L. Fried, Probability of getting a lucky short-exposure image through turbulence, JOSA 68 (12) (1978) 1651–1658.

[14] M. Aubailly, M. A. Vorontsov, G. W. Carhart, M. 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, Vol. 7463, International Society for Optics and Photonics, 2009, p. 74630C.

[15] N. Anantrasirichai, A. Achim, N. G. Kingsbury, D. R. Bull, Atmospheric turbulence mitigation using complex wavelet-based fusion, IEEE Transactions on Image Processing 22 (6) (2013) 2398–2408.

[16] M. C. Roggemann, C. A. Stoudt, B. M. Welsh, Image-spectrum signal-to-noise-ratio improvements by statistical frame selection for adaptive-optics imaging through atmospheric turbulence, Optical Engineering 33 (10) (1994) 3254–3265.

[17] Y. Lou, S. H. Kang, S. Soatto, A. L. Bertozzi, Video stabilization of atmospheric turbulence distortion, Inverse Probl. Imaging 7 (3) (2013) 839–861.

[18] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, Journal of the ACM (JACM) 58 (3) (2011) 11.

[19] R. He, Z. Wang, Y. Fan, D. Fengg, Atmospheric turbulence mitigation based on turbulence extraction, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 1442–1446.

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

[21] C. P. Lau, Y. H. Lai, L. M. Lui, Variational models for joint subsampling and reconstruction of turbulence-degraded images, arXiv preprint arXiv:1712.03825.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.

[24] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5769–5779.

[26] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, Deblurgan: Blind motion deblurring using conditional adversarial networks, arXiv preprint arXiv:1711.07064.

[27] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[28] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853.

[29] D. Ulyanov, A. Vedaldi, V. Lempitsky, Instance normalization: the missing ingredient for fast stylization. corr abs/1607.08022 (2016).

[30] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, arXiv preprint.

[31] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[33] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[34] K. C. Lam, L. M. Lui, Landmark-and intensity-based registration with large deformations via quasi-conformal maps, SIAM Journal on Imaging Sciences 7 (4) (2014) 2364–2392.