ATMOSPHERIC TURBULENCE REMOVAL USING CONVOLUTIONAL NEURAL NETWORK

Jing Gao, N. Anantrasirichai and David Bull

Visual Information Laboratory, University of Bristol, UK

ABSTRACT

This paper describes a novel deep learning-based method for mitigating the effects of atmospheric distortion. We have built an end-to-end supervised convolutional neural network (CNN) to reconstruct turbulence-corrupted video sequence. Our framework has been developed on the residual learning concept, where the spatio-temporal distortions are learnt and predicted. Our experiments demonstrate that the proposed method can deblur, remove ripple effect and enhance contrast of the video sequences simultaneously. Our model was trained and tested with both simulated and real distortions. Experimental results of the real distortions show that our method outperforms the existing ones by up to 3.8% in term of the quality of restored images, and it achieves faster speed than the state-of-the-art methods by up to 23 times with GPU implementation.

Index Terms— Deep learning, atmospheric turbulence restoration, image restoration

1. INTRODUCTION

Distortions due to atmospheric conditions can degrade the visual quality of image during acquisition from cameras. This phenomenon can be found over hot roads, deserts and the object around flames, where the atmospheric turbulences appear as spatio-temporal ripples in video sequences (as shown in Fig.1 left column). This phenomenon also reduces contrast, sharpness and the capacity to see objects at a long distance, such as those in surveillance applications. It often combines with other detrimental effects, like fog, which make the acquired imagery even more difficult to interpret.

Previous work on atmospheric turbulence removal, e.g. complex steerable pyramid (CSP) [1] and complex wavelet transform (CW) [2, 3], is normally based on image processing methods when referring to software-based techniques. These methods usually have three major drawbacks. Firstly, they generally involve a complex optimisation process which needs large buffers, computational processors [3, 4] and long computational time [5, 6]. Secondly, they usually deal with geometric distortion and blurry degradation separately as they were considered as irrelevant problems, which increased the total complexity. Thirdly, the complex models could be non-convex and some parameters need to be chosen manually [3]. Based on these issues, we hence consider exploring a deep learning method to reconstruct the distorted images. Here, a convolutional neural network (CNN) is employed to characterise atmospheric turbulence distortions in video sequences, and then the sporadic motions and rapid distortions will be mitigated with locally trained filters in each layer of the network. Moreover, CNN-based methods have proved that, through implementation using graphics accelerator devices, they can achieve real-time processing and require reduced memory [7].

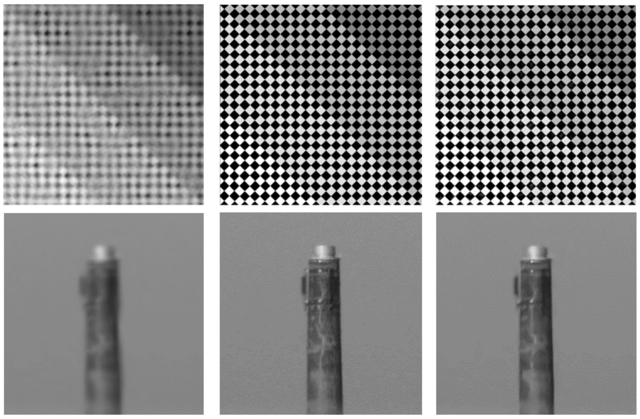

Fig. 1. Restoration of atmospheric turbulence distortion from (top) simulated turbulence on a chessboard and (bottom) real scene [8]. (Left to right) Distort image, ground truth and our restored image.

Image restoration can be addressed as an inverse problem and deep learning techniques have been employed to solved it. The learning-based image restoration for a specific distortion includes deblurring [9], denoising [10] and super-resolution [11]. More details of deep learning for inverse problems can be found in [6]. The atmospheric turbulence scenario, however, contain blur effects combined with small-scale intensity fluctuations due to scintillation, which means this problem comprises several distortions. As such, existing deep learning methods are possibly not suitable for removing atmospheric turbulence.

This paper presents a new method to deal with atmo-

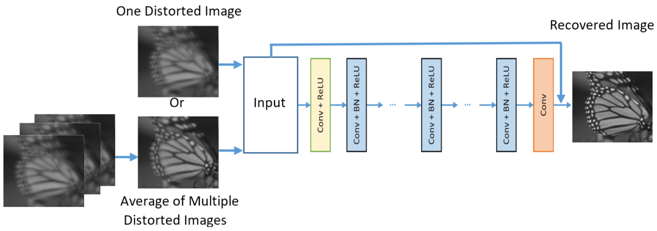

Fig. 2. The proposed architecture adapted from DnCNN [4], showing two options of input, a distorted image or the average of multiple distorted frames.

spheric turbulence effects on images using a deep neural network. To the best of our knowledge, this work is the first to apply deep learning to the task of alleviating turbulence effect on images. We propose to restore atmospheric turbulence-corrupted images using the modified residual learning of deep CNN (DnCNN) [4]. Here, the DnCNN, originally introduced for image denoising, is adapted to remove noise and spatio-temporally invariant blur simultaneously. It employs residual learning with batch normalization to achieve better image quality and faster training speed. We adapted the size of filters from 3×3 to $n \times n$, where $n > 3$, to obtain larger receptive field and more adjustable parameters for this more complicated task. We also investigate the performance of our model when using a single distorted image and an average of the multiple distorted images as the input of the network.

2. DEEP LEARNING FOR ATMOSPHERIC TURBULENCE REMOVAL

Atmospheric turbulence mitigation can be seen as an inverse problem, solving degradation model [3], $y = hx + b$, where y represents the distorted observation of an ideal image x (ground truth). h represents geometric distortion and blurry degradation, whilst b is noise from sensors or transmission process. Our target is to recover x from y. A direct mapping between ground truth and observed distorted image is $F(y; \theta) = x$, where F is the mapping function and θ represents a set of parameters of F. The key is to find F and its θ. CNN regards image pairs (x, y) as the inputs for training; each x_i and each y_i are a pair. Then CNN updates θ that include weights of filters and parameters of batch normalization through backpropagation by reducing the difference between y and x.

Recently a residual learning has proved that it can accelerate the training process by learning a residual mode [4]. In this paper, we use residual learning formulation to train a residual mapping $R(y; \theta)$ between the deformation map (residual image) and the distorted input, i.e. $x = y - R(y; \theta)$. Eq.1 shows the loss function to learn the trainable parameters θ for this mapping. The loss function is the sum square value of each pixel between ground truth images and recovered images.

$$L(\theta) = \frac{1}{2m} \sum_{1}^{m} \left(|R(y_i; \theta) - (y_i - x_i)|^2 \right) \qquad (1)$$

where L is a loss function calculated by mean square error, and m is the number of image pairs.

2.1. Network architecture

The modified architecture, shown in Fig. 2, contains three kinds of layers.

i) *Convolution + ReLU*: In the first layer, 64 filters (size $n \times n$) are used for generating 64 feature maps through convolution. Then ReLU is applied for non-linearity. It is used in the first layer (yellow block in Fig. 2). The size of the filter ($n \times n$) is defined to be large enough to cover the spatial movement of the ripples. However, this value is unknown in real scenes. In this paper we manually measured it and define n associated to the receptive field, explained in the next subsection. For future work, an optical flow technique could be employed to estimate the maximum pixel motion [12].

ii) *Convolution + Batch normalization + ReLU*: From the second to the sixteenth layer, 64 filters with the size of $n \times n \times 64$ are used. There is batch normalization between convolution and ReLU. As batch normalization has little influence on the front layers, we only add batch normalization in the hidden layers (blue blocks in Fig. 2).

iii) *Convolution*: In the last layer (orange block in Fig. 2), one filter with the size $n \times n \times 64$ is implemented to output a deformation map.

Finally, the recovered image is created by subtracting the deformation map to the distorted image.

2.2. Network depth

Deep depth and large convolutional filters generally give better feature extraction performance. However, they require more parameters and thus may bring in redundancy. In this paper, we use 17 layers to balance between efficiency and performance, similar to the architecture in [4]. We however leave the filter size open to $n \times n$ so that it can be freely defined to make the reception field cover the range of pixel motion according to the turbulence effects in different scenes. For severe turbulence, the pixel motion requires a larger effective receptive field. In our dataset, we observe that the maximum pixel displacement is less than 30 pixels in a single direction (total moving range is 60 pixels). The filter size of 5×5 (n=5) with a depth of 17 results the receptive field to 69×69 pixels, which cover the pixel displacement of the turbulence. We have tried the original filter size (3×3) of DnCNN, and it gave worse results for removing atmospheric turbulence. A large size of training patches is required as it helps to make maximum use of context information, leading to the better recovering performance. Here, we set the patch size of 80×80 pixels, which is slightly larger than the receptive field.

2.3. Input of the network

The atmospheric turbulence can be viewed as being quasi-periodic; therefore, averaging a number of frames yields a geometric improvement in the image, but it remains blurred by an unknown point spread function (PSF) of the same size as the pixel motions due to the turbulence [13]. In this paper, we investigate the performance of our model when using a single distorted image and using the average of group of the distorted images as the input of the network. This average approach can be simply applied to the distorted sequence with moving objects by using sliding temporal window approach.

3. RESULTS AND DISCUSSION

We tested our proposed method with both simulated and real datasets. The model was trained by with Adam optimizer with batch size of 128 and learning rate from 1e-3 to 1e-5. The batch normalisation is implemented to further stable the parameters. Same padding is implemented in all convolution layers. We set the convolution stride equals to 1. The training was done over 1000 epochs. The CPU in this paper is with Intel Core i5-6200U @2.30GHz and the GPU is Nvidia Tesla P100-PCIE-16GB.

3.1. Experiment on synthetic datasets

Datasets with ground truth is essential for supervised learning; unfortunately, the availability of the datasets for atmospheric turbulence problem is very limited. Therefore, we generated the synthetic atmospheric turbulence distortion. This was done with 9 PSFs provided in [8]. The spatially variant blur is created by applying the randomly selected PSF to the different parts of the image, and for each image the PSFs are resized randomly so that the strength of ripple effects and degree of blur vary between frames. Then a Gaussian noise is added. We employed 20 clean images with a size of 180×180 pixels [4] shown in Fig. 3 and 2 clean images: 'building' and 'chimney' ground truth provided in [8].

We generated 2 separate synthetic datasets (Fig. 4(a)) for training and testing the model. These images share the same ground truth but with different degrees of distortion. The result for recovering a simulated image with a single distorted image is presented in Fig. 4 (c). We can see that most blurs and distortions have been removed for all images. Table 1 shows the PSNR and SSIM of 6 scenes in Fig. 4. All PSNR of the restored results are over 24dB and all SSIM are over 0.76. We can see that the proposed method can improve the image quality by up to 30% from the original quality.

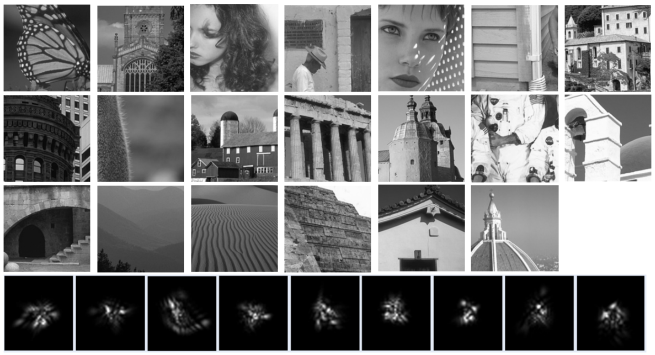

Fig. 3. Row 1-3 show clean images of simulated datasets gathered from [4]. Row 4 shows nine different atmospheric PSFs used for generated the synthetic datasets [8].

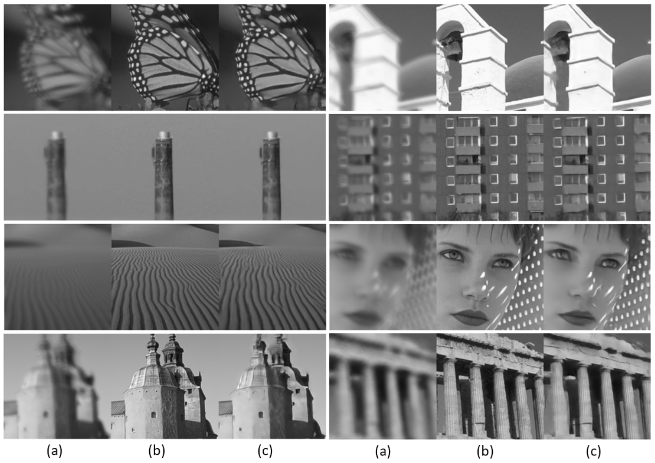

Fig. 4. Reconstruction of simulated dataset. (a) distorted image, (b) ground truth, (c) recovered image.

Table 1. Objective assessment of simulated distorted dataset (d) and the restored results (r)

Image	Butterfly	Arch	Chimney	Building	Desert	Face
PSNR$_d$	19.27	19.82	32.77	23.49	24.81	21.48
SSIM$_d$	0.61	0.73	0.95	0.78	0.65	0.73
PSNR$_r$	24.28	26.88	39.49	28.45	25.64	28.39
SSIM$_r$	0.88	0.93	0.98	0.92	0.76	0.86

3.2. Experiment on static real datasets

We employed the test datasets, 'building' and 'chimney', captured from real word [8]. Each scene contains 100 frames, and the first 80 were used for training. Test results of the state-of-the-art methods were gathered from their published papers directly. Our method mitigates the distortion in frame-by-frame basis, i.e. one restored image is outputted for one distorted image. This is different from other existing methods, where multiple inputs are required to generate one restored image. Therefore our objective results are shown as an average of PSNR and SSIM of 20 testing images ($\bar{y}_{20}$), when only sin-

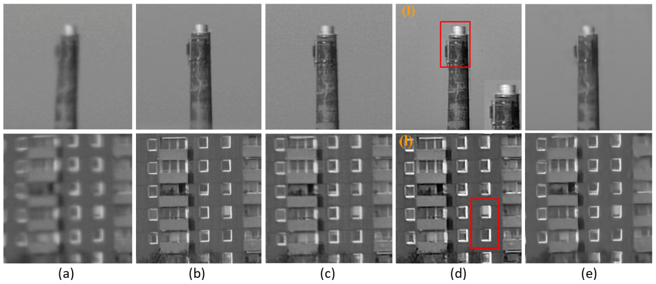

Fig. 5. Comparison of results on the real dataset. (a) distorted image, (b) ground truth, (c) CW, (d) CSP, (e) proposed method. CW and CSP are reconstructed from selected frames, whilst the input of our method is the average of 20-30 images.

gle distorted input is use. However, the visual result of the 'chimney' can not always get result as good as that in Fig. 1, because the real distortion contain more complicated distortion that we may not learn well with the limited dataset. To seek for a stable result, we investigated the performance of our model for multiple inputs by simply averaging these distorted images as the input (as shown in Fig. 2), and the result is one restored frame ($\bar{x}$).

The subjective results are shown in Fig. 5, where (a) are the distorted images, (b) are ground truth, (c) and (d) are the reconstruction of CW [3] and CSP [1] separately, and (e) are the proposed method. CW and CSP are reconstructed from a number of good frames or patches selected from 100 frames, whilst our method employed only 30 and 20 continuous input frames and averaged them for 'chimney' and 'building' separately. CSP and CW can derive sharp results, but the visual quality may be different from the ground truth.

The corresponding objective results of Fig. 5 are shown in Table 2. The best PSNR and SSIM are highlighted in bold. We can see that the proposed method with the average input ($\bar{x}$) yields the highest PSNR. Specifically, it outperforms the competing methods by 1.66dB (CW) and 2.83 dB (CSP). But the SSIM value for building is the highest for CW.

3.3. Experiment on real datasets with moving objects

The sequence 'moving car' [5] was tested. We adapted the architecture to accept 3 adjacent distorted images and 1 ground truth. Then 20 of 100 frames were used for testing whilst the rest were employed for training. In this video, the car size changes as it is coming toward us. In order to deal with different sizes of the moving car in each frame, we used random resize parameters (0.7 to 1) for data augmentation in the training process.

Fig. 6 shows the results, where (a) is one of the distorted frame, (b) is clear frame, and (c) is its directly reconstructed image. Then, we used the average of 5 adjacent frames for each input channel (i.e. at time t, the inputs of each channel are frame t-6 to t-2, frame t-5 to t-1, and frame t-4 to t) when perform testing. The result of average input is shown in Fig. 6 (d), of which the subjective quality is comparable with that of the CW [5] (Fig. 6 (b)).

Table 2. Objective performance comparison of real dataset

Image	Index	CW	CSP	Proposed	
				($\bar{y}_{20}$)	($\bar{x}$)
Chimney	PSNR	32.02	31.05	30.67	**33.68**
	SSIM	0.96	0.92	0.95	**0.97**
Building	PSNR	25.18	25.37	23.23	**26.02**
	SSIM	**0.91**	0.79	0.82	0.88

$\bar{y}_{20}$=average PSNR or SSIM of 20 recovered images
$\bar{x}$=input is the average of 30 images for 'chimney' and 20 for 'building'

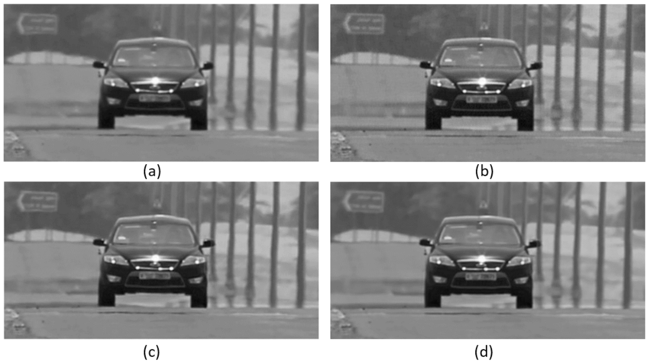

Fig. 6. Reconstruction of 'moving car'. (a) distorted image, (b)'ground truth' [5] (c) directly recovered images, (d) recovered images with averaged frames.

The processing time of image restoration of this dataset (size of 512×256 pixels) by using GPU is 0.08 seconds/frame, which equal to 1638.4k pixels per second. It is faster than the CW reported in [5] (70.6k pixels per second). When using CPU, the computational time for dealing the same image is 6.84 seconds, which equal to 19.2k pixels per second.

4. CONCLUSIONS

This paper presents a deep learning method with residual learning for removing atmospheric turbulence distortion using modified DnCNN. It was tested with simulated datasets, real datasets both static and dynamic scenes. We showed that the proposed model can handle turbulence with unknown distortion levels and restore image with only one distorted image, and better result can be achieved with an average of multiple distorted frames. Our method gives equal or better results than the state-of-the-art methods without prior selecting good frames. With the simple averaging strategy, we can deal with distortions from the real work including the videos with moving objects. Moreover, it can achieve on-line mitigation for videos with static scene with GPU computation.

5. REFERENCES

[1] Chao Zhang, Bindang Xue, Fugen Zhou, and Wei Xiong, "Removing atmospheric turbulence effects in unified complex steerable pyramid framework," *IEEE Access*, vol. 6, pp. 75855–75867, 2018.

[2] N. Anantrasirichai, A. Achim, D. Bull, and N. Kingsbury, "Mitigating the effects of atmospheric distortion using DT-CWT fusion," in *IEEE International Conference on Image Processing*, Sept 2012, pp. 3033–3036.

[3] N. Anantrasirichai, A. Achim, N.G. Kingsbury, and D.R. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *Image Processing, IEEE Transactions on*, vol. 22, no. 6, pp. 2398–2408, 2013.

[4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[5] Nantheera Anantrasirichai, Alin Achim, and David Bull, "Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 2895–2899.

[6] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos, "Using deep neural networks for inverse problems in imaging: beyond analytical methods," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20–36, 2018.

[7] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[8] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 607 –614.

[9] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891.

[10] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3155–3164.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[12] H. Chen and N. Kingsbury, "Efficient registration of nonrigid 3-d bodies," *Image Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 262 –272, Jan. 2012.

[13] Yu Mao, Jérôme Gilles, and Selim Esedoglu, "Non rigid geometric distortions correction - application to atmospheric turbulence stabilization," *Inverse Problems & Imaging*, vol. 6, no. 3, pp. 531–546, 2012.