Split-and-Merge Algorithm for Motion Deblurring Based on DeblurGAN

著者	ZHU YANMIN
出版者	法政大学大学院情報科学研究科
journal or publication title	法政大学大学院紀要．情報科学研究科編
volume	14
page range	1-4
year	2019-03-31
URL	http://doi.org/10.15002/00021925

Split-and-Merge Algorithm for Motion Deblurring Based on DeblurGAN

YANMIN ZHU
Graduate School of Computer and Information Sciences
Hosei University
Tokyo 180-0021, Japan
yanmin.zhu.74@stu.hosei.ac.jp

***Abstract*—Motion blur is a highly challenging problem in computer vision literature. Due to the ill-posed nature and the camera shake, the relative motion between the camera and the object in 3D space induces a spatially varying blurring effect over the entire image. This paper proposes split-and-merge algorithm based on DeblurGAN to make blurred image become relatively clear. For the same motion blur image, usually blurred degree of different parts is not the same. Therefore split-and-merge algorithm firstly split a blurred image into several parts and measure blurred degree for each part. Then the algorithm do deblurring for different times according to different blurred degree until all parts reach the acceptable degree. Finally the algorithm merge all parts into one image. The experiment results show this algorithm can help enhance sharpness of motion blur picture than original DeblurGAN.**

Keywords— Motion deblurring; Generative Adversarial Network; Image deblurring; split and merge.

I. INTRODUCTION

Motion blur is a common problem which occurs predominantly when capturing an image using light weight devices like mobile phones. Due to the finite exposure interval and the relative motion between the capturing device and the captured object, the image obtained is often blurred. Solving this problem has been an attractive research topic for the last two decades and significant progress has been recently achieved in related areas of image super-resolution [1] by applying generative adversarial networks (GANs) [2]. GANs are known for the ability to preserve texture details in images, create solutions that are close to the real image manifold and look perceptually convincing. Inspired by recent work on image super-resolution [1] and image-to-image translation by GANs, I treat image deblurring as a special case of such image-to-image translation. For enhancing image's sharpness, I propose split-and-merge algorithm to split an image into several parts and measure blurred degree for each part firstly. Then the algorithm do deblurring for different times according to different blurred degree using trained generator of DeblurGAN [11] until all parts' sharpness reach the threshold value. Finally the algorithm merge them into one clearer image. Experiment was done basing on 100 pieces of testing pictures. No matter firstly splitting into 3 parts by across cutting or 4 parts by crossing type, final results prove that split-and-merge algorithm can generate clearer image than original DeblurGAN.

Supervisor: Prof. Yuji Sato

II. RELATED WORK

A. Image deblurring

Image deblurring is a traditional inverse problem whose aim is to recover a sharp image from the blurry image. Over the years, numerous methods have been proposed to tackle the problem. Deblurring problems is divided into two types: blind and non-blind deblurring. The common formulation of non-uniform blur model is the following formulation (1):

$$I_B = \mathrm{Conv}\,(k(M),\ I_S) + N, \tag{1}$$

where I_B is a blurry image, $k(M)$ are unknown blur kernel determined by motion field M. I_S is the sharp latent image, Conv ( ) denotes the convolution relationship, N is an additive noise. Non-blind deblurring indicates that the blur kernel $k(M)$ is assumed to be known and a sharp image can be induced from both the blurry image I_B and the kernel $k(M)$. By contrast, for the blind deblurring problem, commonly the blur kernals $k(M)$ is unknown, and both latent sharp image I_S and blur kernels $k(M)$ are estimated.

With the success of deep learning, over the last few years, there appeared some approaches based on convolutional neural networks (CNNs). Sun et al. [3] use CNN to estimate blur kernel, Chakrabarti [4] predicts complex Fourier coefficients of motion kernel to perform non-blind deblurring in Fourier space whereas Gong [5] use fully convolutional network to move for motion flow estimation. All of these approaches use CNN to estimate the unknown blur function. Recently, a kernel-free end-to-end approaches by Noorozi [6] and Nah [7] that uses multi-scale CNN to directly deblur the image. Ramakrishnan et al. [8] use the combination of pix2pix framework [9] and densely connected convolutional networks [10] to perform blind kernel-free image deblurring. Such methods are able to deal with different sources of the blur.

B. Generative Adverserial Networks

Generative adversarial networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework. They were introduced by Ian Goodfellow [2] et al. in 2014. This technique can generate photographs that look at least superficially authentic to human observers, having many realistic characteristics. The two competing networks of GANs are the discriminator and the generator. The generator receive

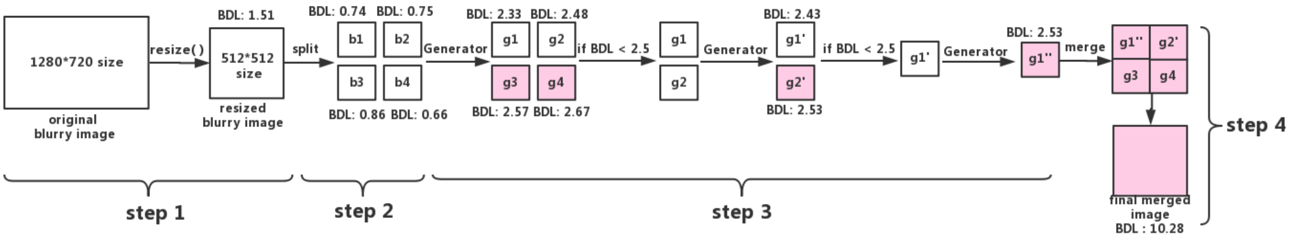

Figure 2. An example of Split-and-Merge process

noise as an input and generates a sample. A discriminator receives a real and generated sample and is trying to distinguish between them. The goal of the generator is to fool the discriminator by generating perceptually convincing samples that can not be distinguished from the real one. Formulation (2) show the minimax objective relationship between the generator G and discriminator D :

$$\min_{G} \max_{D} V(D,G) = E_{x \to p_{data}(x)}[logD(x)] + E_{z \to p_z(z)}[log(1-D(G(z)))]. \quad (2)$$

where x means real image sample from $p_{data(x)}$. $p_{data(x)}$ means the distribution of real data. $D(x)$ means the probability that Discriminator think real image as real. z means the sample from $p_{(z)}$. $p_{(z)}$ means the distribution of Generator. $G(z)$ means the generated image by Generator, $D(G(z))$ is the probability that Discriminator think generated image as real. Figure 1 shows the training process of GANs.

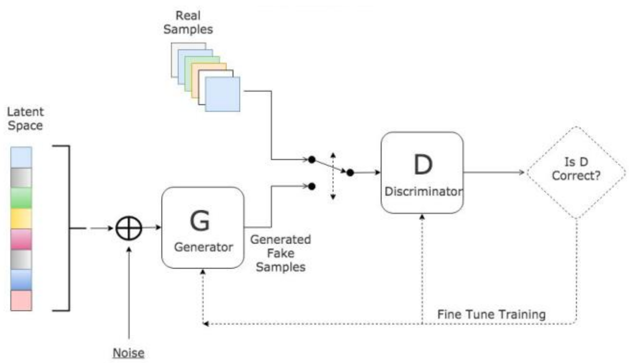

Figure 1. Training process of GANs

Figure 1 describes the training process of GANs. Typically, the generative network's training objective is to increase the error rate of the discriminative network, which learns to map from a latent space with some noise added to a particular data distribution of interest. While the discriminative network's training objective is to increase judgment ability, which discriminates between instances from the real samples and fake samples produced by the generator. If the judgment ability is too strong, then the generator need to do fine tune training. If not, the discriminator needs to do fine tune training. The competitive relationship between generator and discriminator makes each other better.

III. Proposal of Split-and-Merge Algorithm

In most cases for the same image, the blurred degree of separate position is usually different. After one time of deblurring, some areas of the whole blurred image have reach the specified acceptable sharpness we want while some other areas haven't reach the threshold. But some part could be over-deblurred if the whole image is deblurred for many times. Under this condition, Split-and-Merge algorithm is proposed. An example of Split-and-Merge process is shown in Figure 2. The BDL in the figure means BDL_value mentioned in Step 1.

Step 1: Reshape the given original blurry image into 512*212 size using resize ( ) method from PIL (Python Image Library). Measure the blurry degree of original blurry image using the total variance of the laplacian [12] of an image, this provides a quick and accurate method for scoring how blurry an image is. The value is called BDL_value in the paper. The higher of this value means the less blur. If the value is larger than threshold 2.5, we think the sharpness of the image is acceptable. The BDL_value of original blurry image is written as bd_0. In figure 2, bd_0 is equal to 1.51.

Step 2: Split the original blurry image into several parts using crop ( ) method from PIL. It was splited into 4 parts with 256*256 size in the paper. Measure the blurry degree of each splited part using the same method to step 1. They are written as $bd_{1(0)}$, $bd_{2(0)}$, $bd_{3(0)}$, $bd_{4(0)}$. If bd_i is shorter than threshold 2.5, it need to be deblurred in step 3. If not, it can skip Step 3. In figure 2, $bd_{1(0)}$ is equal to 0.74, $bd_{2(0)}$ is equal to 0.75, $bd_{3(0)}$ is equal to 0.86, $bd_{4(0)}$ is equal to 0.66.

Step 3: Deblur the parts whose *BDL_value* is shorter than 2.5 through trained generator of DeblurGAN [11]. Measure the blurry degree of these deblurred part. They are written as $bd_{i(1)}$. If $bd_{i(1)}$ is shorter than threshold 2.5, it need to be deblurred again until the value is larger than 2.5. After n times of deblurring, the *BDL_value* of the i^{th} part are written as $bd_{i(n)}$. (where i=1, 2, 3, 4. n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9). If the $bd_{i(n)}$ value of some part is still shorter than 2.5 after 5 times of deblurring, we will not do deblurring again and use the 5 times' result. In figure 2, after one time of deblurring, $bd_{1(1)}$ is equal to

TABLE I. THE *BDL_VALUE* FOR 20 IMAGES WITH DIRECT DEBLURRING AND SPLIT-AND-MERGE ALGORITHM

Image	Final 4 part	Bd1	Bd2	Image	Final 4 part	Bd1	Bd2
1	$bd_{1(3)}, bd_{2(5)}, bd_{3(3)}, bd_{4(5)}$.	10.2795	2.2689	11	$bd_{1(2)}, bd_{2(2)}, bd_{3(3)}, bd_{4(4)}$.	13.3241	2.3069
2	$bd_{1(2)}, bd_{2(3)}, bd_{3(4)}, bd_{4(5)}$.	14.0400	2.4243	12	$bd_{1(3)}, bd_{2(3)}, bd_{3(4)}, bd_{4(5)}$.	9.2946	2.3707
3	$bd_{1(1)}, bd_{2(3)}, bd_{3(2)}, bd_{4(5)}$.	8.9018	2.9517	13	$bd_{1(2)}, bd_{2(2)}, bd_{3(4)}, bd_{4(4)}$.	14.1013	2.4127
4	$bd_{1(3)}, bd_{2(3)}, bd_{3(4)}, bd_{4(2)}$.	12.3670	2.3264	14	$bd_{1(3)}, bd_{2(3)}, bd_{3(4)}, bd_{4(4)}$.	11.6384	2.3657
5	$bd_{1(2)}, bd_{2(2)}, bd_{3(2)}, bd_{4(4)}$.	10.8437	2.2798	15	$bd_{1(4)}, bd_{2(4)}, bd_{3(3)}, bd_{4(5)}$.	10.9372	2.2709
6	$bd_{1(2)}, bd_{2(3)}, bd_{3(3)}, bd_{4(4)}$.	10.0928	2.2780	16	$bd_{1(3)}, bd_{2(4)}, bd_{3(4)}, bd_{4(4)}$.	11.2845	2.2706
7	$bd_{1(2)}, bd_{2(2)}, bd_{3(2)}, bd_{4(4)}$.	11.3628	2.2915	17	$bd_{1(5)}, bd_{2(5)}, bd_{3(4)}, bd_{4(4)}$.	10.2525	2.2816
8	$bd_{1(3)}, bd_{2(4)}, bd_{3(3)}, bd_{4(3)}$.	12.3245	2.3362	18	$bd_{1(3)}, bd_{2(4)}, bd_{3(4)}, bd_{4(5)}$.	10.2532	2.3428
9	$bd_{1(3)}, bd_{2(2)}, bd_{3(5)}, bd_{4(4)}$.	10.9821	2.2914	19	$bd_{1(4)}, bd_{2(5)}, bd_{3(4)}, bd_{4(4)}$.	9.2917	2.1837
10	$bd_{1(3)}, bd_{2(2)}, bd_{3(4)}, bd_{4(5)}$.	12.2361	2.2946	20	$bd_{1(4)}, bd_{2(4)}, bd_{3(3)}, bd_{4(3)}$.	13.0128	2.2066

2.33, $bd_{2(1)}$ is equal to 2.48, $bd_{3(1)}$ is equal to 2.57 which is larger than threshold 2.5, $bd_{4(1)}$ is equal to 2.67 which is larger than threshold 2.5. Thereforethe *3rd* and *4th* part have reach the acceptable value and *1st* and *2nd* part need to be deblurred again. Finally, for *1st* part we get $bd_{1(3)}$ = 2.53 and for *2nd* part we get $bd_{2(2)}$ = 2.53.

Step 4: After deblurring process of Step 3. We get the most higher $bd_{i(n)}$ value for each part: $bd_{1(3)}$ = 2.53, $bd_{2(2)}$ = 2.53, $bd_{3(1)}$ = 2.57, $bd_{4(1)}$ = 2.67. Then we merge these 4 parts together to get a clearer image with 512*512 size and get the BDL_value of the clear image. In figure 2, the BDL_value of the final clear image is equal to 10.28.

IV. EXPERIMENT

A. Experimental method

The dataset is the GOPRO dataset [7]. There are a light version (9GB) and a complete version (35GB). It consists of 2103 pairs of artificially blurred and real sharp images in 720p (1280*720 size image) quality, taken from multiple street views. The dataset is decomposed in subfolders by scenes. The light version dataset is choosed to train and test the DeblurGAN [11]. DeblurGAN is an approach based on conditional generative networks and a multi-component loss function. 50 blurred images are choosen from the dataset to do experiments according to the proposed idea. The detailed process about how to deal with one image according to the proposed method is as follows.

Initially, for the whole original blurred image without splitting process, the *BDL_value* of direct deblurring image is written as Bd2. For the image, the Bd2 value for the 2.2689.

Step 1: Reshape the given blurred image into 512*512 size and 256*256 size and measure their blurry degree. They are showed in Figure 3. The *BDL_value* of 512*512 size image is 1.5129. The *BDL_value* of 256*256 size image is 0.9931.

Step 2: Split the 512*512 size image into 4 parts with 256*256 size and measure the blurry degree for each part. They are showed in Figure 3. $bd_{1(0)}$ = 0.7382 (top-left part), $bd_{2(0)}$ = 0.7532 (top-right part), $bd_{3(0)}$ = 0.8564 (bottom-left part), $bd_{4(0)}$ = 0.6619 (bottom-right part).

Figure 3. Splitted 4 parts from original blurred image

Step 3: Deblur these 4 parts beacause $bd_{i(0)}$(i = 1, 2, 3, 4) are all shorter than threshold 2.5. After deblurring, they are showed in Figure 4. $bd_{1(1)}$ = 2.3251 (top-left part), $bd_{2(1)}$ = 1.9806 (top-right part), $bd_{3(1)}$ = 2.3693 (bottom-left part), $bd_{4(1)}$ = 2.1674 (bottom-right part). According to the method in Step 3, deblurring need to be continued beacause $bd_{i(1)}$(i = 1, 2, 3, 4) are all shorter than threshold 2.5.

Step 4: Finally we get $bd_{1(3)}$ = 2.5043, $bd_{2(5)}$ = 2.4654, $bd_{3(3)}$ = 2.5041, $bd_{4(5)}$ = 2.4716.

Figure 4. Splitted 4 parts after first deblurring

B. Experimental results

The final deblurring results for each part, where $bd_{1(3)}$ = 2.5043, $bd_{2(5)}$ = 2.4654, $bd_{3(3)}$ = 2.5041, $bd_{4(5)}$ = 2.4716, are merged into one image: *deblur_merge_final.png*. The

BDL_value of *deblur_merge_final.png* is written as Bd1. The *Bd1* for the image is 10.2795. Table 1 list the *Bd1* value and *Bd2* value for 20 images with direct deblurring and Split-and-Merge algorithm. We can see from the table that the *Bd1* value is larger than *Bd2* value for all the 20 images. The values comparison demonstrates that compared to direct deblurring for the whole blurry image, images exactly have less blur after Split-and-Merge algorithm.

C. Discussion

For comparing the detailed difference of deblurred images from direct deblurring and Split-and-Merge algorithm. In figure 5, the original blurry image is cut into 4 pieces, which are in the left part. The deblurred image from direct deblurring is cut into 4 pieces, which are on the middle part. The right part is are the 4 images which compose of *deblur_merge_final.png* in the preceding Experimental results part of the paper.

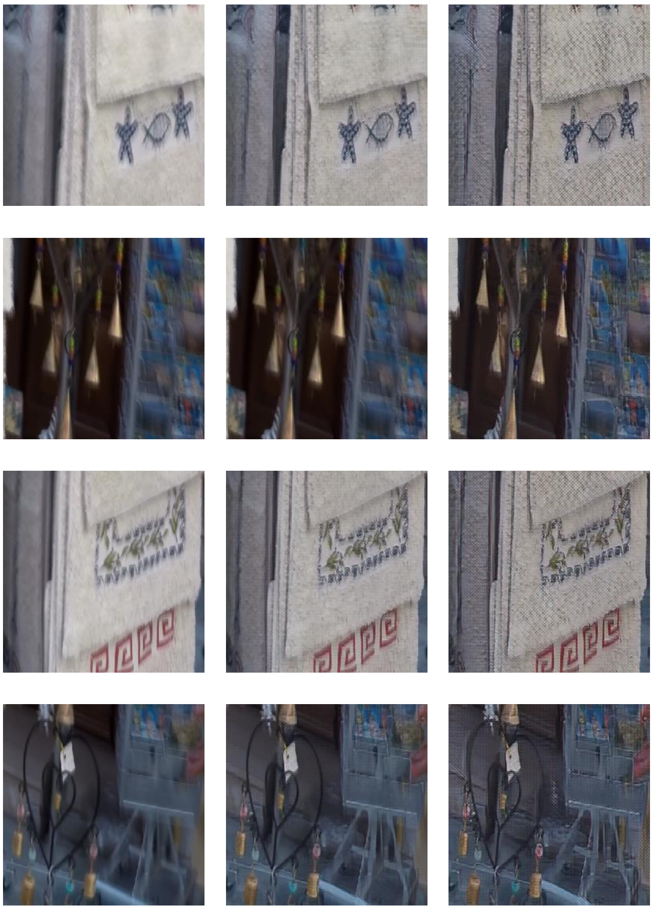

Figure 5. Details comparison of images from the left original blurry image, image after direct deblurring (middle, *BDL_value*: 2,2689) And image after Split-and-Merge algorithm (right, *BDL_value*: 10.2795).

It is easy to find that the middle and the right one are both have clear details than the left original blurry image. It shows that Split-and-Merge algorithm and convolutional DeblurGAN can realize motion deblurring. From the top to the bottom, in the first images' comparison, the outfit of stars and fishes in the right image is apparently better than the middle one. In the second images' comparison, the outfit of yellow triangle decorations in the right image is apparently clearer and blue cabinet's details is not therefore mottled like the middle image. In the third images' comparison, the outfit of green leaves and red symbols in the right image apparently have higher resolution than the middle one. In the fourth images' comparison, the pink decorations in the black heart-shaped appliance are more tridimensional than the middle one.

V. Conclusion

In this paper, we have proposed a Split-and-Merge algorithm for motion deblurring based on DeblurGAN. After splitting a blurred image into several parts, The algorithm focuses on deblurring for different times according to different blurred degree until all parts reach the acceptable degree. Finally the algorithm merge all parts into one image. Details comparison of images from direct deblurring and Split-and-Merge algorithm show this algorithm can help enhance sharpness of motion blur picture than original DeblurGAN.

VI. Acknowledement

The work of this paper is under the guide of my supervisor, Professor Yuji Sato. It is his scientific methods of work and tirelessly guidance that gives me a great deal of help. Therefore I really would like to say thanks for his guidance enlightening and detailed inspection.

References

[1] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. ArXiv e-prints, Sept. 2016. 1, 3, 4

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. June 2014. 1, 3

[3] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal. 2015. 3, 5, 7, 8

[4] A. Chakrabarti. A neural approach to blind motion deblurring. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016. 3, 5

[5] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From Motion Blur to Motion Flow: a Deep Learning Solution for Removing Heterogeneous Motion Blur. 2016. 3

[6] M. Noroozi, P. Chandramouli, and P. Favaro. Motion Deblurring in the Wild. 2017. 3, 5, 8

[7] S. Nah, T. Hyun, K. Kyoung, and M. Lee. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. 2016. 1, 2, 3, 4, 5, 6, 7, 8

[8] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman. Deep generative filter for motion deblurring. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 2993–3000, 2017. 3

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. arxiv, 2016. 1, 3, 4, 5, 7

[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017. 3

[11] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J. (2017). DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. arXiv preprint arXiv:1711.07064.

[12] Bansal, R., Raj, G., & Choudhury, T. (2016, November). Blur image detection using Laplacian operator and Open-CV. In System Modeling & Advancement in Research Trends (SMART), International Conference (pp. 63-67). IEEE.