IMAGE REFLECTION REMOVAL USING THE WASSERSTEIN GENERATIVE ADVERSARIAL NETWORK

Tingtian Li and Daniel P.K. Lun

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

ABSTRACT

Imaging through a semi-transparent material such as glass often suffers from the reflection problem, which degrades the image quality. Reflection removal is a challenging task since it is severely ill-posed. Traditional methods, while all require long computation time on minimizing different objective functions with huge matrices, do not necessarily give satisfactory performance. In this paper, we propose a novel deep-learning based method to allow fast removal of reflection. Similar to the traditional multiple-image approaches, the proposed algorithm first captures the multi-view images of a scene. Then the images are fed to a convolutional neural network to obtain the depth information along the edges of the image. It is sent to a Wasserstein generative adversarial networks (WGAN) for estimating the edges of the background. Finally, the background edges are used in another WGAN to reconstruct the background image. Experimental results show that the proposed method can achieve state-of-the-art performance, and is significantly faster than the traditional methods due to the use of the deep learning methods.

Index Terms—Reflection removal, Wasserstein generative adversarial network, blind image separation

1. INTRODUCTION

Images with the reflection of an unwanted scene are acquired frequently in daily lives when imaging through semi-transparent material such as glass. It does not only degrade the visibility of the desired background but also affects the subsequent analyses and applications of the images. Various approaches are proposed in the last decades for solving this problem. Mathematically, the reflection scene I_R which is superimposed on the background scene I_B in the captured image I can be modeled as follows:

$$I = I_B + I_R. \tag{1}$$

To recover I_B from I is a typical blind image separation problem. Since there are two variables needed to be solved from only one equation, this problem is severely ill-posed. Various priors such as the gradient sparsity and the assumption that the edges of two layers are seldom overlapped [1-4] are adopted to relieve the problem. However, with only the observed image, this problem is too difficult to solve, as two variables needed to be solved from one equation. Methods using multiple images [2-4] and light field (LF) images [5, 6] were then developed and they showed better performance comparing to the single-image based methods. These approaches allow the depth of the scene to be evaluated to facilitate the identification of the background and reflection, which often locate at different distances from the camera. However, if a pixel is the superimposition of the background and reflection of different depths, the depth of that pixel is ambiguous. Furthermore, it is noticed in many practical situations that some parts of the background and reflection can share the same depth range. Hence just using the depth of a pixel is still insufficient to determine if it belongs to the background or reflection. In [6], we suggested distinguishing the background and reflection through their edges. It is seldom that the edges of the background and reflection overlap in an image, as they are usually uncorrelated. Besides, we suggested excluding the edge points having the depth in the range shared by the background and reflection. Rather, they are regenerated based on the remaining edge points. Although the method in [6] has good performance, it requires a long computation time to carry out a few large-scale optimization processes in the algorithm.

Quite recently, the learning-based approaches, such as deep neural networks (DNN), are also applied to the problem of reflection removal [7, 8]. Since these approaches are single-image based, it is difficult to find an effective cue that can be used to clearly distinguish background and reflection in the image. Hence, only some weak priors that are not generally true in practice are used for training the DNN. For instance, both methods in [7, 8] assume reflection images must be blurry, which is not valid in many practical situations. Two examples are shown in Fig. 5. In fact, the generality of such an assumption is also pointed out in [7]. In this paper, we suggest extending the ideas in [6] but realizing them using the WGAN. As a branch of DNN, generative adversarial networks (GAN) has drawn distinct attention recently. It was successfully applied to solve many inverse problems such as inpainting [9], super-resolution

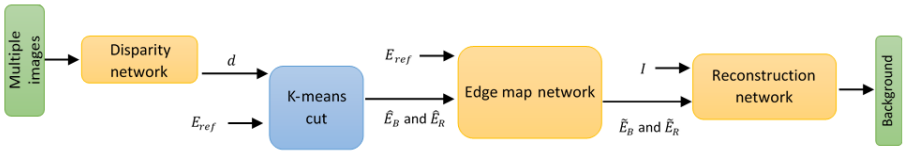
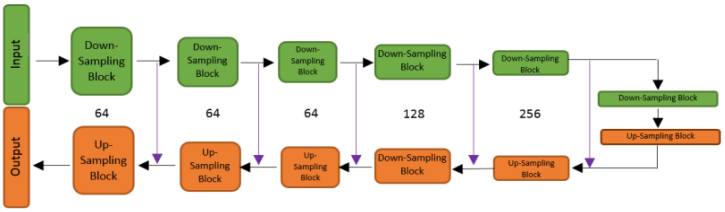


Fig. 1. The flow chart of the proposed algorithm.

(a) Generator network:



(b) Discriminator network:

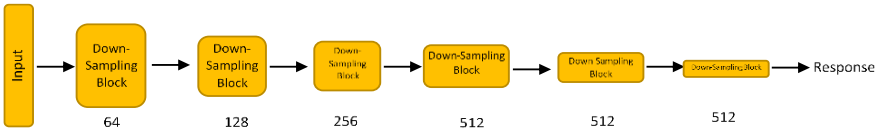


Fig. 2. The network architectures of WGAN. (a) The generator network. Each down-sampling block represents: 4x4 conv.(stride 2, (stride 4 for the last one)) +BN+ReLU+5x5 conv.(stride 1)+BN+ReLU+5x5 conv.(stride 1)+BN+ReLU. Each down-sampling block represents: 4x4 transposed conv.(stride 2, (stride 4 for the first one))+BN+ReLU+5x5conv.(stride 1)+BN+ReLU+5x5 conv.(stride 1)+BN+ReLU. The purple arrow denotes the concatenation operation. (b) The discriminator network. Each down-sampling block represents: 4x4 conv.(stride 2)+BN+leakyReLU. The number near each block represents the channel number of that block.

[10], etc. However, the training of GAN is a minimax process, which can be unstable and difficult to converge. By adopting the Wasserstein distance in the loss function, it is shown in [11] that only slight changes to the discriminator and loss function are needed to achieve stable and fast convergence. The resulting GAN structure is called WGAN. The algorithm proposed in this paper follows closely with that in [6], but replaces the three most time-consuming processes in [6] by an edge disparity network, edge map network, and edge reconstruction network (as shown in Fig. 1) implemented using a convolutional neural network (CNN) and two WGANs. Similar to [6], the algorithm makes use of the depth of the image edges to identify the background. It is a much stronger prior than those in [7, 8], hence the robustness of the algorithm is improved. The edge points having the depth in the range shared by both the background and reflection should be excluded. They are regenerated using a WGAN. The strong ability of WGAN in synthesizing new data following the ground truth's distribution lets it particularly suitable to the edge regeneration task. Experimental results show that the proposed algorithm achieves a state-of-the-art performance and greatly improve the computation speed.

2. EDGE DISPARITY NETWORK

For most existing multiple-image reflection removal methods, the image depth is first computed based on the disparity derived from the multiple views of the target scene. To obtain multi-view images, we use an LF camera for convenience. By means of a lens array, each set of LF images contains multiple views of a scene at slightly different angles. Estimating the depth information from LF images can be time-consuming [12]. Here, we use a CNN to estimate the edge depth from LF images. The proposed edge disparity network contains 8 convolutional layers. Except for the last one, each convolutional layer is followed by a batch normalization layer and ReLU. The filter sizes of the first, second and last convolutional layers are 25x5x5x256, 256x5x5x128 and 128x5x5x1, respectively. The convolutional layers in the middle are all 128x5x5x128. The cost function is based on the reconstruction cost but not the depth ground truth, which is hard to obtain. We train the CNN by minimizing the following cost function,

$$\mathcal{L}_d = \sum_{u,x} \left\| M_u(x) L(x,u) - M_u(x) L\big(x + (u - u_{ref}) d(x), u_{ref}\big) \right\|^2, \quad (2)$$

where d is the disparity; L is the 4-dimensional LF image; x and u represent the spatial and angular coordinates of LF respectively; u_{ref} is the reference view we choose for reconstructing the background, and M_u is the gradient magnitude map of view u. It is used to emphasize the edges in the cost function. In the testing phase, the input LF images are fed to the proposed CNN and generate the depth values along the edges in the image (we call it the edge depth). Similar to [6], we use the edge depth to generate the initial edges of the background and reflection, respectively. However, not all background and reflection edge points are included as mentioned above. Those edge points with the depth values in the range shared by both the background and reflection are prone to error, which are excluded from the initial background and reflection edges; and regenerate them based on the remaining ones by using a WGAN.

3. EDGE MAP NETWORK

3.1 Edge completion using WGAN

The learning process of WGAN can be described as follows:

$$\min_G \max_D \mathbb{E}_{x \in \chi}[D(x)] - \mathbb{E}_{z \in Z}\big[D\big(G(z)\big)\big], \quad (3)$$

where $\mathbb{E}$ is the expectation operator, G and D are the generator and discriminator, respectively. In (3), $G(z)$ tries to map the input data z following the distribution Z to the data x following the distribution χ. The task of the discriminator D is to distinguish the data $G(z)$ from the real data x. The target is to find a generator G that can produce fake data that the discriminator D cannot distinguish, which means that $G(z)$ must be very close to the data in χ. The network architectures of the WGAN used in this study are shown in Fig. 2. The generator is constructed using a U-net like structure. It is because the encoder-decoder structure of U-net, which first shrinks and then interpolates the lost part, is suitable to deal with inverse like problems [9]. The discriminator contains several convolutional layers but

discards the sigmoid operation as suggested in [11]. The input z contains E_{ref}, $\hat{E}_B$, and $\hat{E}_R$, where E_{ref} is the edges obtained directly from the image; $\hat{E}_B$ and $\hat{E}_R$ are the initial background and reflection edges.

3.2 Training of WGAN for edge estimation

For training the proposed WGAN, we prepared a number of LF images with known background and reflection ground truths. More details of the training samples can be found in Section 5. Using the background and reflection ground truths, we can also obtain their edges ground truths. Then we train the initial generator by minimizing the following L2 norm loss function,

$$\mathcal{L}_{rec}^{E} = \|G^{E}(z) - E_B\|_2^2, \tag{4}$$

where G^E is the output of the generator and E_B is the ground truth background edges. Then, we define two adversarial losses which correspond to the discriminators D_1^E and D_2^E for discriminating the background and reflection edges given by the generator, respectively. The two adversarial loss functions are,

$$\mathcal{L}_{adv_1}^{E} = D_1^E(E_B) - D_1^E\big(G^E(z)\big); \tag{5}$$

$$\mathcal{L}_{adv_2}^{E} = D_2^E(E_R) - D_2^E\left(E_{ref} - G^E(z)\right), \tag{6}$$

where E_B and E_R are the ground truth background and reflection edges, respectively. We train the discriminators D_1^E and D_2^E by maximizing $\mathcal{L}_{adv_1}^{E}$ and $\mathcal{L}_{adv_2}^{E}$. After D_1^E and D_2^E are trained, $\mathcal{L}_{adv_1}^{E}$ and $\mathcal{L}_{adv_2}^{E}$ are used to form an overall loss function as,

$$\mathcal{L}^{E} = \mathcal{L}_{rec}^{E} + \lambda_1\big(\mathcal{L}_{adv_1}^{E} + \mathcal{L}_{adv_2}^{E}\big), \tag{7}$$

where λ_1 is the Lagrange multiplier which balances the loss terms. Then, we re-train the generator G^E by minimizing the overall loss function $\mathcal{L}^E$. The process iterates until converged. In the testing phase, E_{ref}, $\hat{E}_B$, and $\hat{E}_R$ are fed to the generator to estimate the background edges. A background edge mask is also generated by thresholding the edges with a threshold 0.05. Fig. 3(c) shows an example where the background edges are largely recovered from the initial edges (Fig. 3(b)).

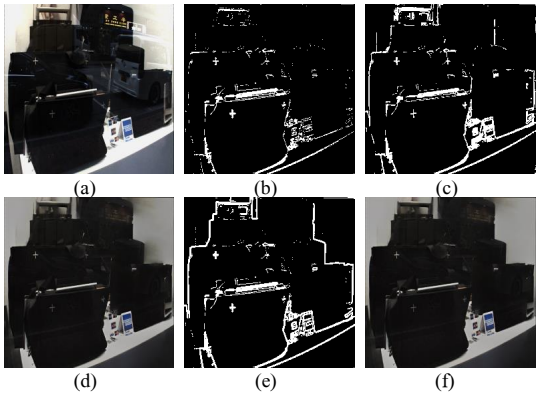


Fig. 3. (a) The original image; (b) the initial incomplete background edge map; (c) the refined background edge map; (d) the reconstructed background using the edges indicated by (c); (e) the enhanced background edge map by selecting the strong edges of (d); (f) further enhanced background using the edges indicated by (e).

4. BACKGROUND RECONSTRUCTION NETWORK

As indicated in [6], we can reconstruct the background image based on the refined background edges. However, the complex optimization method used in [6] is extremely time-consuming. In this paper, we suggest using another WGAN to reconstruct the background image to improve the computation speed. The WGAN used here has the same architecture as in Fig. 2. The inputs are the original image (with reflection), the background edges and reflection edges obtained from the edge map network. The loss functions used are also similar to (4) to (7), except that we further enhance the perceptual similarity by adding a perceptual loss. The overall loss function is now defined as,

$$\mathcal{L}^{B} = \mathcal{L}_{rec}^{B} + \lambda_2\mathcal{L}_{p}^{B} + \lambda_3\big(\mathcal{L}_{adv_1}^{B} + \mathcal{L}_{adv_2}^{B}\big), \tag{8}$$

$$\mathcal{L}_{rec}^{B} = \|G^{B}(z) - I_B\|_2^2, \tag{9}$$

$$\mathcal{L}_{p}^{B} = \big\|V\big(G^{B}(z)\big) - V(I_B)\big\|_2^2; \tag{10}$$

$$\mathcal{L}_{adv_1}^{B} = D_1^B(I_B) - D_1^B\big(G^B(z)\big); \tag{11}$$

$$\mathcal{L}_{adv_2}^{B} = D_2^B(I_R) - D_2^B\big(I - G^B(z)\big). \tag{12}$$

G^B, D_1^B, D_2^B are the generator of the background image, the discriminators for the background and reflection images respectively. I, I_B and I_R are the original, ground truth background and reflection images, respectively. V represents the perceptual loss which is the response of the first 14 layers of the VGG-16 model. As pointed out in [14], adding the intermediate response of a pre-trained model to the loss function can improve the perceptual similarity between the reconstructed image and the ground truth. A reconstruction example is shown in Fig. 3(d). We can see that the background image is well reconstructed according to the edge mask (Fig. 3(c)). The result can be further enhanced by inputting the strong edges of the reconstructed background to the networks again. Fig. 3(e) and (f) show how the edge map and reconstructed background image are further enhanced. All the edge maps in Fig. 3 are obtained by thresholding the edge magnitudes with a constant 0.05.

5. EXPERIMENTS AND EVALUATION

5.1 Dataset and training details

For training the different DNNs used in the proposed algorithm, we need many multi-view images with reflection and their background ground truth. To simplify the experiment, we make use of LF images since every LF image contains multiple views of a scene. Since it is difficult to obtain the background ground truths of real LF images (with reflection), we synthesize the required images by superimposing two sets of LF images with different weightings. We capture 318 LF images from different scenes using a Lytro Illum LF camera and resize them to 256x256 pixels. We divide them into two sets and randomly superimpose them such that 112,225 different training samples are obtained. The training image samples can be further augmented using different superimposition coefficients and flipping. We train the edge disparity

network using ADAM [15] with learning rate 2×10^{-5}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For the edge map network and background reconstruction network, we use RMSprop [16] instead of ADAM as suggested in WGAN [11]. The learning rates of the generator and discriminators are 2×10^{-4} and 2×10^{-5} respectively. The networks are trained sequentially for avoiding overfitting. The parameters λ_1, λ_2, and λ_3, in the loss functions are set as 2.5×10^{-3}, 1.25 and 4.4×10^{-3} respectively. We train the networks on a desktop computer using a GTX 1080 Ti.

5.2 Performance and computation speed

To evaluate the performance of the proposed approach, we compare it with four recent methods, LS-LFGS [6], LS-DS [5], LS-SIFTF [3], CEILNet [8] and PLNet [7]. LS-LFGS is our previously proposed method. LS-SIFTF and LS-DS are two traditional methods using multiple images and LF images respectively. CEILNet and PLNet are DNN based methods. We quantitatively evaluate the performance of these methods by using 10 images. Each image is the superimposition of two images, which mimic the background and reflection. Since the ground truth of each image is known, we can evaluate the PSNR of the recovered background. The average results are shown in Table I, which shows that the proposed algorithm significantly outperforms the other approaches. When comparing with other DNN based methods, the proposed algorithm can reconstruct the background edges by exploiting the edge depth information based on multiple-image. For qualitative evaluation, we show the results of two real cases in Fig. 5. It can be seen that the proposed algorithm significantly outperforms LS-SIFTF, LS-DS, CEILNet and PLNet. Table I also shows the average running times for processing five 256x256 real-life images. The proposed method is faster than the other non-DNN approaches, including [6], by an order of magnitude.

Fig. 4. The images used in the quantitative evaluation.

Method	Background results	Ave. Time
Original images	13.09	NA
LS-LFGS [6]	21.71	69.51s
LS-SIFTF [3]	18.91	130.59s
LS-DS [5]	18.85	17.01s
CEILNet [8]	17.71	0.82s
PLNet [7]	19.09	1.15s
Proposed	**24.22**	**1.08s**

Table I. The average execution times and PSNR values of the resulting images generated by different methods with respect to their ground truths. The mean values of all the results are adjusted to that of the ground truth for the ease of comparison, as different biases exist in these approaches.

6. CONCLUSION

In this paper, a novel learning-based method for solving the depth-based reflection removal problem is proposed. The new approach uses a CNN to estimate the depth of the image edges, and two WGANs to recover the background edges and reconstruct the background image. Experimental results show that the proposed algorithm can give state-of-the-art performance. Furthermore, it takes advantage of the massively parallel structure of the different deep neural networks used in the algorithm such that much faster speed is achieved when computing with GPU.

ACKNOWLEDGMENT

This work is fully supported by the Hong Kong Polytechnic University under research grant RU9P.

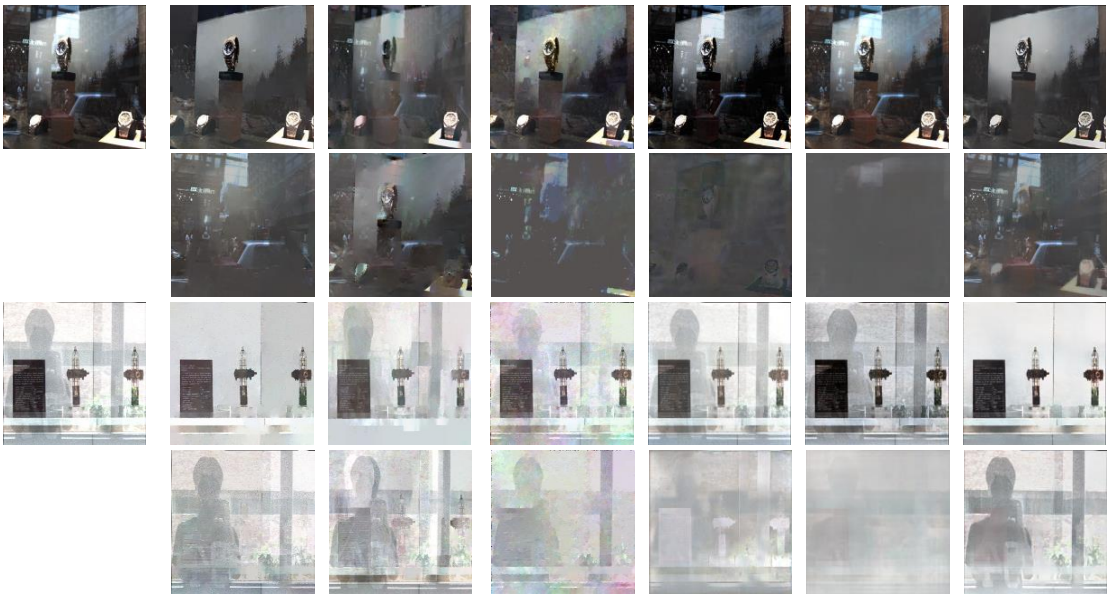


Fig. 5. A comparison of the background (upper row) and reflection (lower row) images generated by different methods for two real image cases.

7. REFERENCES

[1] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 29, 2007.

[2] X. Guo, X. Cao, and Y. Ma, "Robust separation of reflection from multiple images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2187-2194.

[3] Y. Li and M. S. Brown, "Exploiting reflection change for automatic reflection removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2432-2439.

[4] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman, "A computational approach for obstruction-free photography," *ACM Trans. Graph.,* vol. 34, p. 79, 2015.

[5] Y. Ni, J. Chen, and L.-P. Chau, "Reflection Removal Based on Single Light Field Capture," in *Proc. IEEE Int. Sympos. Circuits Syst.*, 2017, pp. 1-4.

[6] T. Li and D. P. K. Lun, "A novel reflection removal algorithm using the light field camera," in *Proc. IEEE Int. Sympos. Circuits Syst.*, 2018.

[7] X. Zhang, R. Ng, and Q. Chen, "Single Image Reflection Separation with Perceptual Losses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[8] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, p. 4.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536-2544.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta*, et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4681-4690.

[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214-223.

[12] C. Hazirbas, S. G. Soyer, M. C. Staab, L. Leal-Taixé, and D. Cremers, "Deep Depth From Focus," in *in Proc. Asian Conf. Comput. Vis.*, 2018.

[13] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph.,* vol. 35, p. 193, 2016.

[14] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Europ. Conf. Comp. Vis.*, 2016, pp. 694-711.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Rep.*, 2015.

[16] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning,* vol. 4, pp. 26-31, 2012.