

GP-GAN: Towards Realistic High-Resolution Image Blending

Huikai Wu

huikai.wu@nlpr.ia.ac.cn
CRISE, CASIA, and UCAS

Junge Zhang

jgzheng@nlpr.ia.ac.cn
CRISE, CASIA, and UCAS

Shuai Zheng

szheng@robots.ox.ac.uk
University of Oxford

Kaiqi Huang*

kaiqi.huang@nlpr.ia.ac.cn
CRISE, CASIA, and UCAS

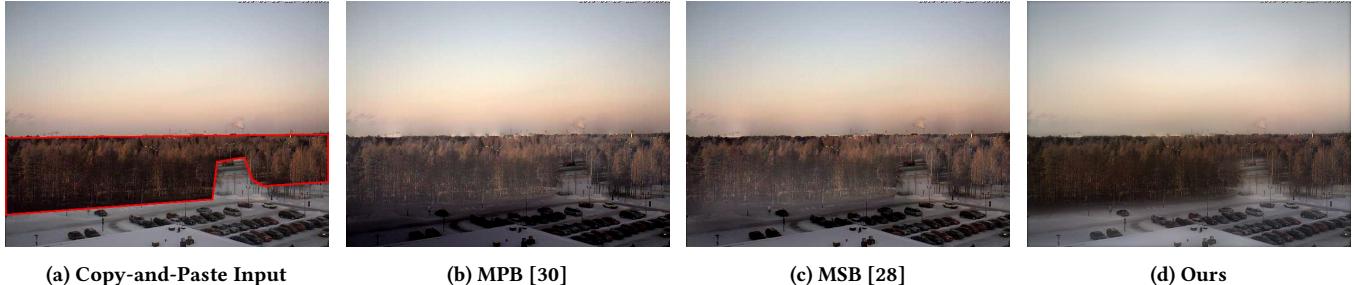


Figure 1: Qualitative illustration of high-resolution image blending. (a) shows the composite copy-and-paste image, where the inserted object is circled out by the red polygon. Our approach (d) produces an image with better quality than those from the alternatives (b) and (c) in terms of illumination, spatial, and color consistencies. Best viewed in color.

ABSTRACT

It is common but challenging to address high-resolution image blending in the automatic photo editing application. In this paper, we would like to focus on solving the problem of high-resolution image blending, where the composite images are provided. We propose a framework called Gaussian-Poisson Generative Adversarial Network (GP-GAN) to leverage the strengths of the classical gradient-based approach and Generative Adversarial Networks. To the best of our knowledge, it's the first work that explores the capability of GANs in high-resolution image blending task. Concretely, we propose Gaussian-Poisson Equation to formulate the high-resolution image blending problem, which is a joint optimization constrained by the gradient and color information. Inspired by the prior works, we obtain gradient information via applying gradient filters. To generate the color information, we propose a Blending GAN to learn the mapping between the composite images and the well-blended ones. Compared to the alternative methods, our approach can deliver high-resolution, realistic images with fewer bleedings and unpleasant artifacts. Experiments confirm that our approach achieves the state-of-the-art performance on Transient Attributes

dataset. A user study on Amazon Mechanical Turk finds that the majority of workers are in favor of the proposed method. The source code is available in <https://github.com/wuhuikai/GP-GAN>, and there's also an online demo in <http://wuhuikai.me/DeepJS>.

CCS CONCEPTS

• Computing methodologies → Image processing; Reconstruction.

KEYWORDS

Image Editing; Image Blending; Image Processing; Generative Adversarial Networks; Poisson Editing

ACM Reference Format:

Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. 2019. GP-GAN: Towards Realistic High-Resolution Image Blending. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3343031.3350944>

1 INTRODUCTION

Technologies such as PhotoShop make it much easier to edit an image than before. However, image editing still requires talents. For example, photos composed by expert users remain far better than the ones from newcomers. As the camera technologies improve, the high-resolution image makes photo editing becomes even more challenging. We want to bridge the talent gap between expert users and beginners on image editing. Mainly, we aim at addressing the problem of high-resolution image blending, which focuses on generating realistic high-resolution images given the composite ones. As shown in Figure 1, users insert an object in the background

*Also with CAS Center for Excellence in Brain Science and Intelligence Technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '19, October 21–25, 2019, Nice, France

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-6889-6/19/10...\$15.00
<https://doi.org/10.1145/3343031.3350944>

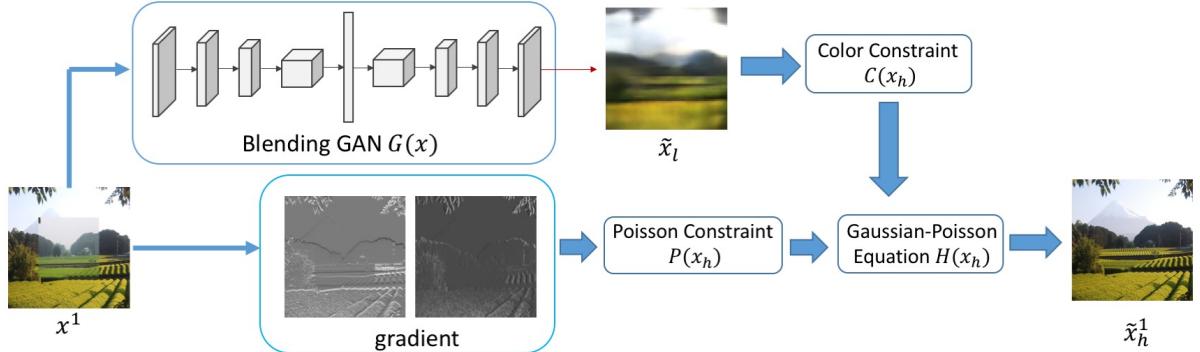


Figure 2: Framework Overview of GP-GAN. Given a composite image x , the low-resolution realistic image \tilde{x}_l is first generated by Blending GAN $G(x)$ with x^1 as the input, where x^1 is the coarsest scale in the Laplacian pyramid of x . Then we optimize the Gaussian-Poisson Equation $H(x_h)$ constrained by $C(x_h)$ and $P(x_h)$ with the closed-form solution, resulting in \tilde{x}_h^1 that contains rich details. We then upsample \tilde{x}_h^1 as the next \tilde{x}_l and optimize the Gaussian-Poisson Equation at a finer scale in the pyramid of x . Best viewed in color.

image (Figure 1a) and want to make it more realistic. Most users would often have high expectation on the quality of the generated images. If the algorithm produces images like Figure 1b or 1c, users will give up the solution after their first few tries.

To generate well-blended images, Perez *et al.*, Tanaka *et al.*, and Szeliski *et al.* [23, 28, 30] propose the classic gradient-based methods, which enable a smooth transition and reduce the color/illumination differences between foreground and background. Among these solutions, Poisson image editing [23] is the most widely used method, which firstly produces a gradient vector field based on the gradients of the composite image and then recovers the blended image from this gradient vector field by addressing a Poisson equation. Such methods are good at generating high-resolution results with rich details and textures. However, the generated images tend to be unrealistic, which contain various kinds of artifacts. Because the traditional gradient-based methods usually have strong assumptions about the distribution of realistic images based on human priors.

Recent researches have achieved significant progress in modeling the distribution of realistic images with the rise of Generative Adversarial Networks (GANs) [2, 5, 9, 24]. Concretely, GANs provide a framework for estimating the distribution of natural images via simultaneously training a generator and a discriminator in a zero-sum game. The generator can produce natural images after training. Mirza *et al.* [13, 21] generalize the idea to a condition setting, which expands the usage of GANs into image-to-image applications like image inpainting [22]. Inspired by the success of GANs in generating realistic images, we propose to employ GANs for overcoming the disadvantages of gradient-based image blending algorithms. Compared to these methods, GANs are much better at modeling the distribution of realistic images. However, it usually takes lots of computation and memory resources to generate high-resolution images with rich details and textures.

We develop a novel framework named GP-GAN to combine the strength of GANs and gradient-based image blending methods, as

shown in Figure 2, which consists of two phases. In phase one, a low-resolution realistic image is generated based on the proposed Blending GAN. In phase two, we solve the proposed Gaussian-Poisson Equation based on the gradient vector field and the generated image in phase one fashioned by the Laplacian pyramid. This framework allows us to achieve high-resolution and realistic images, as shown in Figure 1d, which outperforms all the baseline methods. Our main contributions are four folds, which are summarized as follows:

- We develop a framework GP-GAN for high-resolution image blending that takes advantages of both GANs and gradient-based image blending methods. To the best of our knowledge, it is the first work that explores the capability of GANs in high-resolution image blending task.
- We propose a network called Blending GAN for generating low-resolution realistic images.
- We propose the Gaussian-Poisson Equation for combining gradient information and color information.
- We also conduct a systematic evolution of the proposed approach based on both benchmark experiments and user studies on Amazon Mechanic Turk, which shows that our method outperforms all the baselines and achieves the state-of-the-art performance.

2 RELATED WORK

We briefly review the relevant works from the classical image blending approaches to generative adversarial networks and conditional generative adversarial networks. We also discuss the difference between our work and the others.

2.1 Image Blending

The goal of classical image blending approaches is to improve the spatial and color consistencies between the source and target images. One way [10] is to apply the dense image matching approach to copy and paste the corresponding pixels. However, this method would not work when there are significant differences between the source and target images. The other way is to make the transition

as smooth as possible for hiding artifacts in the composite images. Alpha blending [33] is the simplest and fastest method, but it blurs the fine details when there are some registration errors between the source and target images. Alternatively, [1, 7, 14, 16, 19, 29, 33] address this problem in the gradient domain. Our work is different from these gradient-based approaches in that we introduce GANs to generate a low-resolution realistic image as the color constraint, resulting in a more natural composite image. [32, 35, 39] also address a similar task to ours. However, they focus on adjusting the color and illumination of the inserted object, requiring an accurate segmentation mask. Differently, our method aims at making a smooth transition around the edges of the source and target images as well as reducing the color and illumination differences. Thus, a well-blended image can be generated by our method, although the segmentation mask of the inserted object is coarse.

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [9] are first introduced to address the problem of generating realistic images. The main idea of GANs is a zero-sum game between learning a generator and a discriminator. The generator tries to produce more realistic images from random noises, while the discriminator aims to distinguish generated images from the real ones. Although the original method works for creating digital images from MNIST dataset, some generated images are noisy and incomprehensible. Denton *et al.* [5] improve the quality of the generated images by expanding GANs with a Laplacian pyramid implementation, but it does not work well for the images containing objects looking wobbly. Gregor *et al.* [11] and Dosovitskiy *et al.* [6] achieve successes in generating natural images; however, they do not leverage the generators for supervised learning. Radfor *et al.* [24] achieve further improvement with deeper convolutional network architecture, while Zhang *et al.* [37] stack two generators to progressively render more realistic images. InfoGAN [4] learns a more interpretable latent representation. Salimans *et al.* [27] reveal several tricks in training GANs. Arjovsky *et al.* [2] introduce an alternative training method Wasserstein GAN, which relaxes the GAN training requirement of balancing the discriminator and generator. However, existing GANs still do not work well for the image editing applications in that the generated results are not high-resolution and realistic yet.

2.3 Conditional GANs

Our work is also related to conditional GANs [21], which aims to apply GANs in a conditional setting. There are several works along this research direction. Previous works apply conditional GANs to discrete labels [21], text [25], image inpainting [22], image prediction from a normal map [34], image manipulation guided by user constraints [40], product photo generation [36], style transfer [20], and image-to-image translation [13]. Different from previous works, we use an improved adversarial loss and discriminator for training the proposed Blending GAN. We also propose the Gaussian-Poisson Equation to produce high-resolution images.

3 THE APPROACH

In this section, we first introduce the task of image blending formally. We then present the framework of our Gaussian-Poisson Generative Adversarial Network (GP-GAN).

3.1 Image Blending

Given a source image x_{src} , a destination (target) image x_{dst} and a mask image x_{mask} , the composite (copy-and-paste) image x can be obtained by Equation 1,

$$x = x_{src} * x_{mask} + x_{dst} * (1 - x_{mask}), \quad (1)$$

where $*$ is element-wise multiplication operator. The goal of image blending is to generate a well-blended image \tilde{x} that is semantically similar to the composite image x but looks more realistic and natural with the resolution unchanged. x is usually a high-resolution image.

3.2 Framework Overview

Generating high-resolution well-blended images is hard. To tackle this problem, we propose GP-GAN, a framework for generating high-resolution and realistic images, as shown in Figure 2. This is the first time that GANs are used for realistic high-resolution image blending to the best of our knowledge.

GP-GAN seeks a well-blended high-resolution image \tilde{x}_h by optimizing a loss function consisting of a color constraint and a gradient constraint. The color constraint tries to make the generated image more realistic and natural while the gradient constraint captures the high-resolution details such as textures and edges.

The color constraint is constructed with a low-resolution realistic image \tilde{x}_l . To generate \tilde{x}_l , we propose Blending GAN $G(x)$ that learns to blend a copy-and-paste image and generate a realistic one semantically similar to the input. Once $G(x)$ is trained, we can use it to generate \tilde{x}_l functioning as the color constraint.

The goal of gradient constraint is to generate the high-resolution details, including textures and edges given the composite image x . Their gradients directly capture textures and edges of an image. We propose Gaussian-Poisson Equation to force \tilde{x}_h to have a similar gradient to x while approximating the color of \tilde{x}_l .

GP-GAN can naturally generate realistic images in arbitrary resolution. Given a composite image x , we first obtain \tilde{x}_l by feeding x^1 to $G(x)$, where x^1 is the coarsest scale in the Laplacian pyramid of x . Then we update \tilde{x}_h^1 by optimizing Gaussian-Poisson Equation with the closed-form solution. \tilde{x}_h^1 is upsampled and serves as \tilde{x}_l at the finer scale in the Laplacian pyramid of x . The final realistic image \tilde{x}_h with the same resolution as x is obtained at the finest scale of the pyramid.

In Section 3.3, we will describe the details of our Blending GAN $G(x)$. The details of GP-GAN and Gaussian-Poisson Equation will be described in Section 3.4.

3.3 Blending GAN

We seek a low-resolution well-blended image \tilde{x}_l that is visually realistic and semantically similar to the input image. A straightforward way is to train a conditional GAN and use the generator to produce realistic images. Since we have both the input image and the corresponding ground truth x_g , we aim to train a generator in a supervised way. To achieve this goal, we propose Blending GAN

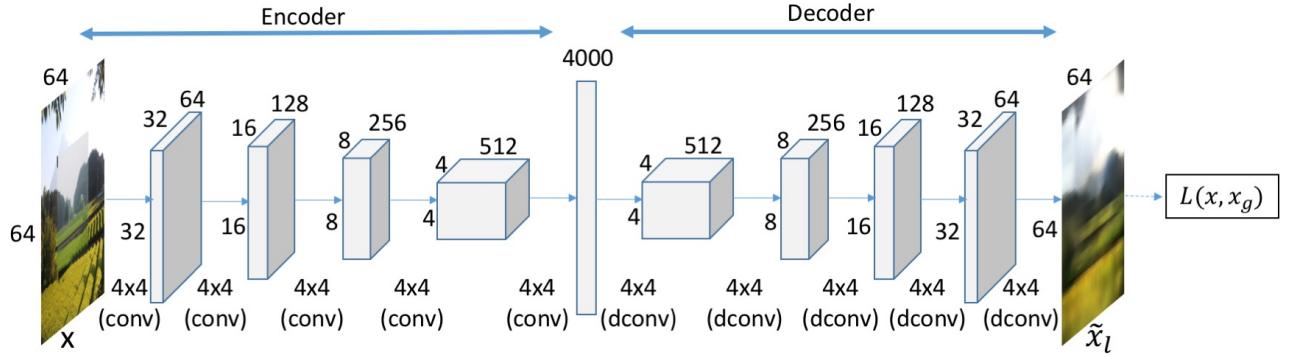


Figure 3: Network architecture of Blending GAN $G(x)$.

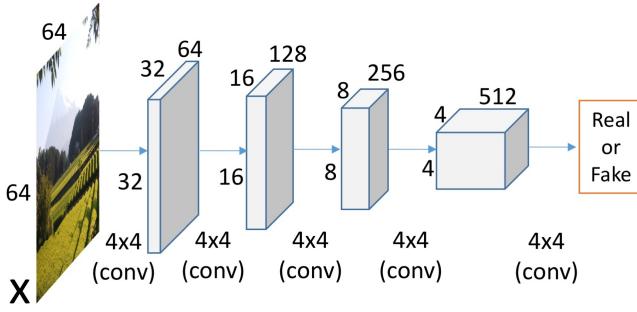


Figure 4: The architecture for the discriminator of Blending GAN.

$G(x)$, which leverages the unsupervised Wasserstein GAN [2] for supervised learning tasks. The proposed Blending GAN is different from Wasserstein GAN in that it has a proper constructed auxiliary loss and dedicated designed architecture.

Recent works discuss various loss functions for image processing tasks, for instance, l_1 loss [38], l_2 loss, and perceptual loss [15]. l_1 and l_2 loss can accelerate the training process but tend to produce blurry images. The perceptual loss is good at generating high-quality images but is time and memory consuming. We employ l_2 loss in this paper because it could accelerate the training process and generate sharp and realistic images when combined with GANs [13]. The combined loss function is defined as follows:

$$L(x, x_g) = \lambda L_{l_2}(x, x_g) + (1 - \lambda) L_{adv}(x, x_g), \quad (2)$$

where λ is 0.999 in our experiment. L_{l_2} is defined as follows:

$$L_{l_2}(x, x_g) = \|G(x) - x_g\|_2^2, \quad (3)$$

and L_{adv} is defined as follows:

$$L_{adv}(x, x_g) = \max_D E_{x \in \chi} [D(x_g) - D(G(x))]. \quad (4)$$

The architecture of Blending GAN $G(x)$ is shown in Figure 3, which is motivated by [22]. We find that a network with only convolutional layers could not learn to blend composite images for the lack of global information across the whole image. Thus we replace the channel-wise fully connected layer used in [22] with standard fully connected layers.



Figure 5: Image blending results generated by $G(x)$. The experiment is conducted on the Transient Attributes Database [18]. x is the copy-and-paste image composed by x_{src} and x_{dst} with a central-squared patch as the mask. \tilde{x}_l is the output of $G(x)$ with size 64×64 . x_g is the ground truth image used for training $G(x)$, which is the same as x_{dst} . Best viewed in color.

The architecture of the discriminator $D(x)$ is shown in Figure 4. We apply the batch normalization [12] and leaky ReLU after each convolution except for the first layer and the last layer. The first layer employs convolution and leaky ReLU, while the last layer contains convolution only.

Training such a network needs massive data. The copy-and-paste images are easy to collect, but the ground truth images x_g could only be obtained by expert users with image editing software, which is time-consuming. Alternatively, we use x_{dst} to approximate x_g , since x_{src} and x_{dst} in our experiment are photos of the same scene under different conditions, e.g. season, weather, time of day, see Section 4.1 for details. Through this way, we obtain massive

composite images and the corresponding ground truth, as shown in Figure 5.

3.4 Gaussian-Poisson Equation

The proposed Blending GAN can only generate low-resolution images, as shown in Figure 5. Even for slightly larger images, the results tend to be blurry with unpleasant artifacts, which is unsuitable for image blending task. Since the task usually needs to combine several high-resolution images and blend them into one realistic image with the resolution unchanged. To make use of the realistic images generated by Blending GAN, we propose Gaussian-Poisson Equation fashioned by the well-known Laplacian pyramid [3] for generating high-resolution and realistic images.

We observe that although our Blending GAN cannot produce high-resolution images, the generated image \tilde{x}_l is natural and realistic as a low-resolution image. So we can seek a high-resolution and realistic image \tilde{x}_h by approximating the color of \tilde{x}_l while capturing rich details like textures and edges in the original high-resolution image x . Such requirements are formulated into two constraints: one is the color constraint, while the other is the gradient constraint. The color constraint forces \tilde{x}_h to have a similar color to \tilde{x}_l , which can be achieved by generating an image with the same low-frequency signals as \tilde{x}_l . The simplest way to extract the low-frequency signals is using a Gaussian filter. The gradient constraint tries to restore the high-resolution details, which is the same as forcing \tilde{x}_h and x to have the same high-frequency signals. This step could be implemented by using the divergence operator.

Formally, we need to optimize the objective function defined as follows:

$$H(x_h) = P(x_h) + \beta C(x_h). \quad (5)$$

$P(x_h)$ is inspired by the well-known Poisson Equation [23] and is defined as follows:

$$P(x_h) = \int_T \|\mathbf{div} v - \Delta x_h\|_2^2 dt, \quad (6)$$

$C(x_h)$ is defined as follows:

$$C(x_h) = \int_T \|g(x_h) - \tilde{x}_l\|_2^2 dt, \quad (7)$$

and β represents the color preserving parameter. We set β to 1 in our experiment. In Equation 6, T represents the whole image region, \mathbf{div} represents the divergence operator and Δ represents the Laplacian operator. v is defined as follows:

$$v^i = \begin{cases} \nabla x_{src}^i & \text{if } x_{mask}^i = 1 \\ \nabla x_{dst}^i & \text{if } x_{mask}^i = 0 \end{cases}, \quad (8)$$

where ∇ is the gradient operator. Gaussian filter is used in Equation 7 and is denoted as $g(x_h)$. The discretized version of Equation 5 is defined as follows:

$$H(x_h) = \|u - LX_h\|_2^2 + \lambda \|GX_h - \tilde{X}_l\|_2^2, \quad (9)$$

where u is the discretized divergence of v , L is the matrix of the Laplacian operator, and G represents the Gaussian filter. X_h and \tilde{X}_l are the vector representation of x_h and \tilde{x}_l . The closed-form solution for minimizing the cost function of Equation 9 could be obtained in the same manner as [8].

We integrate the closed-form solution for optimizing Equation 9 and the Laplacian pyramid into our final high-resolution image

blending algorithm, which is described by Algorithm 1. Given a high-resolution input image x_{src} , x_{dst} and x_{mask} , we first generate the low-resolution realistic image \tilde{x}_l using Blending GAN $G(x)$. Then we generate Laplacian pyramids $x_{src}^s, x_{dst}^s, x_{mask}^s, s = 1, 2, \dots, S$, where S is the number of scales. $s = 1$ is the coarsest scale and $s = S$ is the original resolution. We update x_h^s by optimizing Equation 9 at each scale and set \tilde{x}_l to be upsampled x_h^s . The final realistic image \tilde{x}_h with the unchanged resolution is set to be x_h^S .

Algorithm 1: High-Resolution Image Blending Framework GP-GAN

```

Input : Source image  $x_{src}$ , destination image  $x_{dst}$ , mask
        image  $x_{mask}$  and trained Blending GAN  $G(x)$ 
1 Compute Laplacian Pyramid for  $x_{src}$ ,  $x_{dst}$  and  $x_{mask}$ 
2 Compute  $\tilde{x}_l$  using  $G(x)$ 
3 for  $s \in [1, 2, \dots, S]$  do
4   Updating  $x_h^s$  by optimizing Equation 9 using the closed
      form solution given  $x_{src}^s, x_{dst}^s, x_{mask}^s$  and  $\tilde{x}_l$ 
5   Set  $\tilde{x}_l$  to be upsampled  $x_h^s$ 
6 end
7 Return  $x_h^S$ 

```

4 EXPERIMENTS

In this section, the datasets for the experiments are introduced firstly. Then the training configurations and experimental settings are described. Finally, the effectiveness of our method are shown quantitatively and visually by comparing with other methods.

4.1 Dataset

Transient Attributes Database [18] contains 8,571 images from 101 webcams. In each webcam, there are well-aligned 60-120 images with severe appearance changes caused by weather, time of day, and season, as shown in Figure 6a and Figure 6b.

For training $G(x)$, we randomly select 2 images from the same camera as x_{src} (Figure 6a) and x_{dst} (Figure 6b). As for the ground truth x_g , we use x_{dst} to approximate it since images under the same webcam is perfect-aligned. x_{mask} is a binary image with a central-squared patch, as shown in Figure 6c. The composite copy-and-paste image is then obtained by Equation 1, as shown in Figure 6d. Although $G(x)$ is trained with the central-squared patch as the mask, our experiments show that it is still able to generate well-blended images for inputs with arbitrary masks.

To evaluate our method with arbitrary masks, we first manually annotate object-level masks for Transient Attributes Database with the LabelMe [26] annotation tool. Then we use the object-level masks to composite the copy-and-paste images, which are used to evaluate different image blending methods. The annotated mask and corresponding composite image are shown in Figure 6e and Figure 6f.

4.2 Implementation Details

Our method is implemented with Chainer [31]. To train Blending GAN, we employ ADAM [17] for optimization, where α is set to

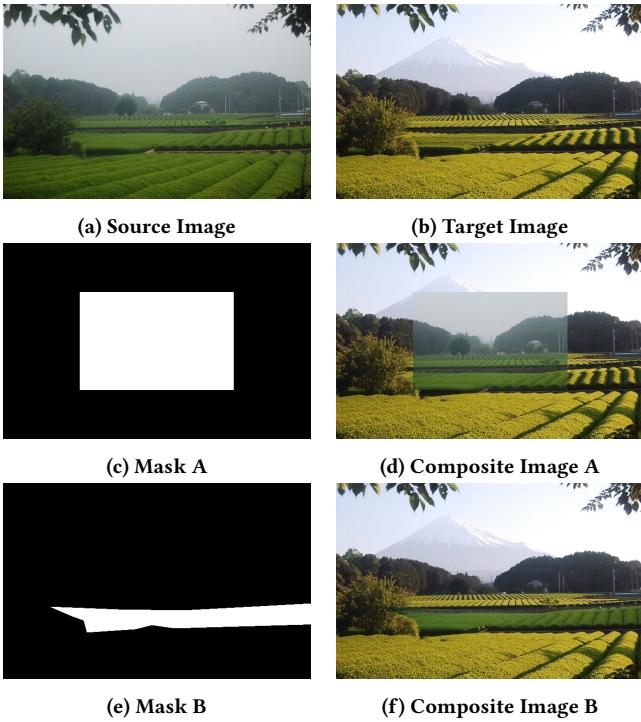


Figure 6: Transient Attributes Database. (a) x_{src} and (b) x_{dst} are from the same webcam but in different seasons. (c) is the central-squared mask and (d) is the corresponding composite image. (e) is the object-level mask annotated with LabelMe and (f) is the corresponding composite image. Best viewed in color.

0.002, and β_1 is set to 0.5. We randomly generate 150K images from Transient Attributes Database using the central-squared patch as the mask. Then the network is trained for 25 epochs with batch size 64.

4.3 Quantitative Comparisons

Our method is compared with three classic image blending approaches. Poisson Image Editing (PB) [23] and its improved version Modified Poisson Image Editing (MPB) [30] are selected as baselines because both of them employ Poisson Equation in their solutions as our method does. We also compare with multi-splines blending (MSB) [28] for its effectiveness and extensive usage.

We first show the quantitative results of our method with realism score as the metric. Realism score is produced by RealismCNN [39], which predicts the visual realism of an image regarding color, lighting, and texture compatibility.

Our method is evaluated on 500 images that are randomly sampled from Transient Attributes Database with the annotated masks. The average realism scores for our method and the baselines are shown in Table 1, where our method outperforms all the baselines. We attribute this to the nature of our method because it can learn what contributes to a realistic and natural image through adversarial learning on large datasets. The average scores are negative for all

evaluated methods, which shows that many blended images are still not realistic. This suggests that there are still many improvements to be made for image blending algorithms.

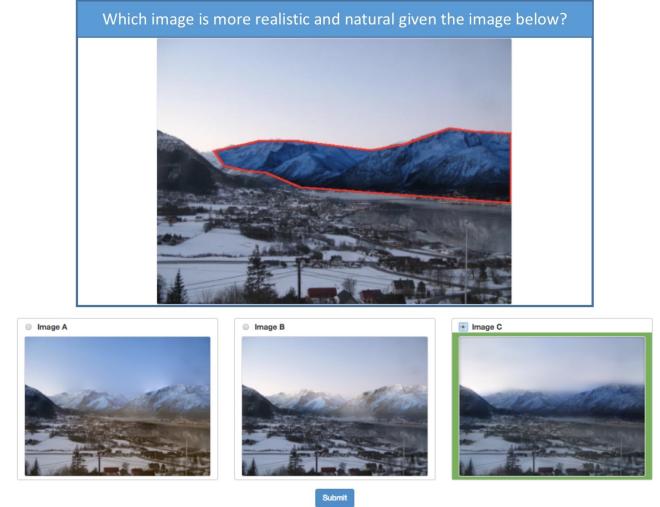


Figure 7: The user interface for user study on Amazon Mechanical Turk. Followed by the composite image with x_{src} circled out, three blended results generated by different algorithms are shown to subjects, and the most realistic one is picked out. Best viewed in color.

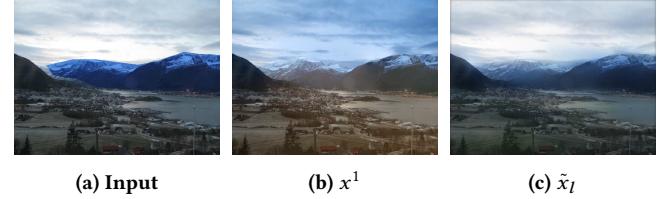


Figure 8: Role of Blending GAN. (a) is a copy-and-paste image. (b) employs the down-sampled x^1 as the color constraint. (c) uses the output of Blending GAN \tilde{x}_l as the color constraint. Best viewed in color.

4.4 User Study

Realism scores show the effectiveness of our method. Since image blending is a user-oriented task, it is essential to conduct user study for evaluation. We employ Amazon Mechanical Turk to collect user assessments. Each time, a composite image x is shown to the subjects followed by three blended images produced by three different algorithms. The subjects are told to pick the most realistic image among these three blended images, as shown in Figure 7. The statistical result of user study is reported in Table 2. GP-GAN is preferred by the majority of users, which is consistent with the result of realism scores in Table 1.

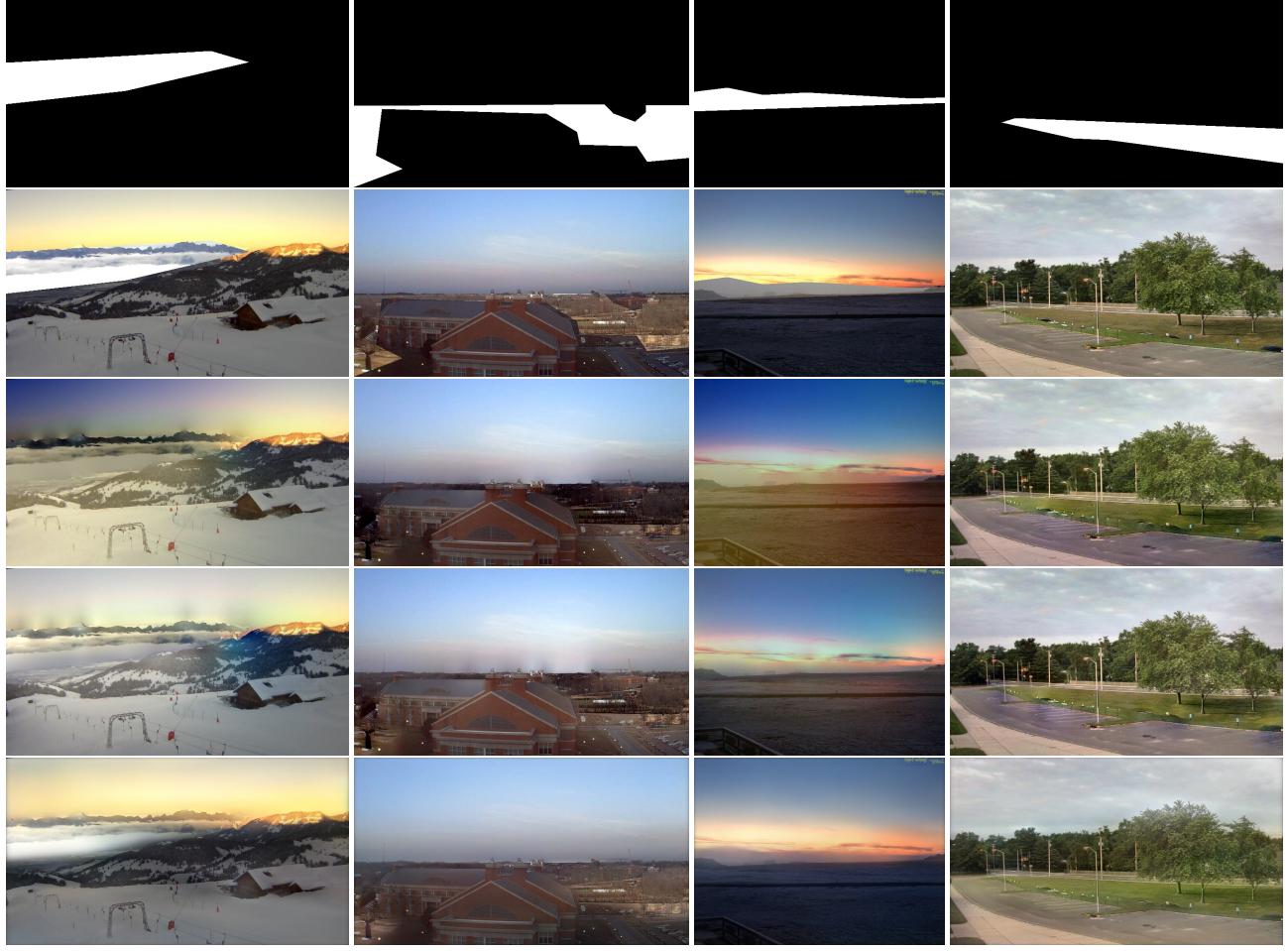


Figure 9: Results of our high-resolution blending algorithm compared with baseline methods. From top to bottom: annotated object-level mask, composite copy-and-paste image, MPB, MSB, and GP-GAN(ours). Results of baseline methods have severe bleedings, illumination inconsistencies, or other artifacts, while GP-GAN produces pleasant, realistic images. Best viewed in color.

Table 1: Realism scores for our method and the baselines (higher is better). GP-GAN outperforms all the baselines.

Method	Input	PB[23]	MPB[30]	MSB[28]	Ours
Score	-0.696	-0.192	-0.151	-0.140	-0.069

4.5 Role of Blending GAN

The output of Blending GAN serves as the color constraint. In this section, we demonstrate the role of Blending GAN by replacing \tilde{x}_l with the down-sampled composite image x^1 . The blended results with either \tilde{x}_l or x^1 as the color constraint are compared. As shown in Figure 8, the blended image tends to have more bleedings and illumination inconsistencies if \tilde{x}_l is replaced by x^1 , which shows the usefulness of low-resolution natural images in our method.

Table 2: User study result. 4 image blending algorithms are compared on Amazon Mechanical Turk. Our method GP-GAN obtains most votes, which is consistent with the result of the realism scores.

Method	Total votes	Average votes	Std.
PB[23]	527	1.054	1.065
MPB[30]	735	1.470	1.173
MSB[28]	770	1.540	1.271
GP-GAN	947	1.894	1.311

4.6 Qualitative Comparisons

Finally, we demonstrate the results of our high-resolution image blending algorithm visually by comparing with MPB and MSB. As shown in Figure 9, our method tends to generate realistic results

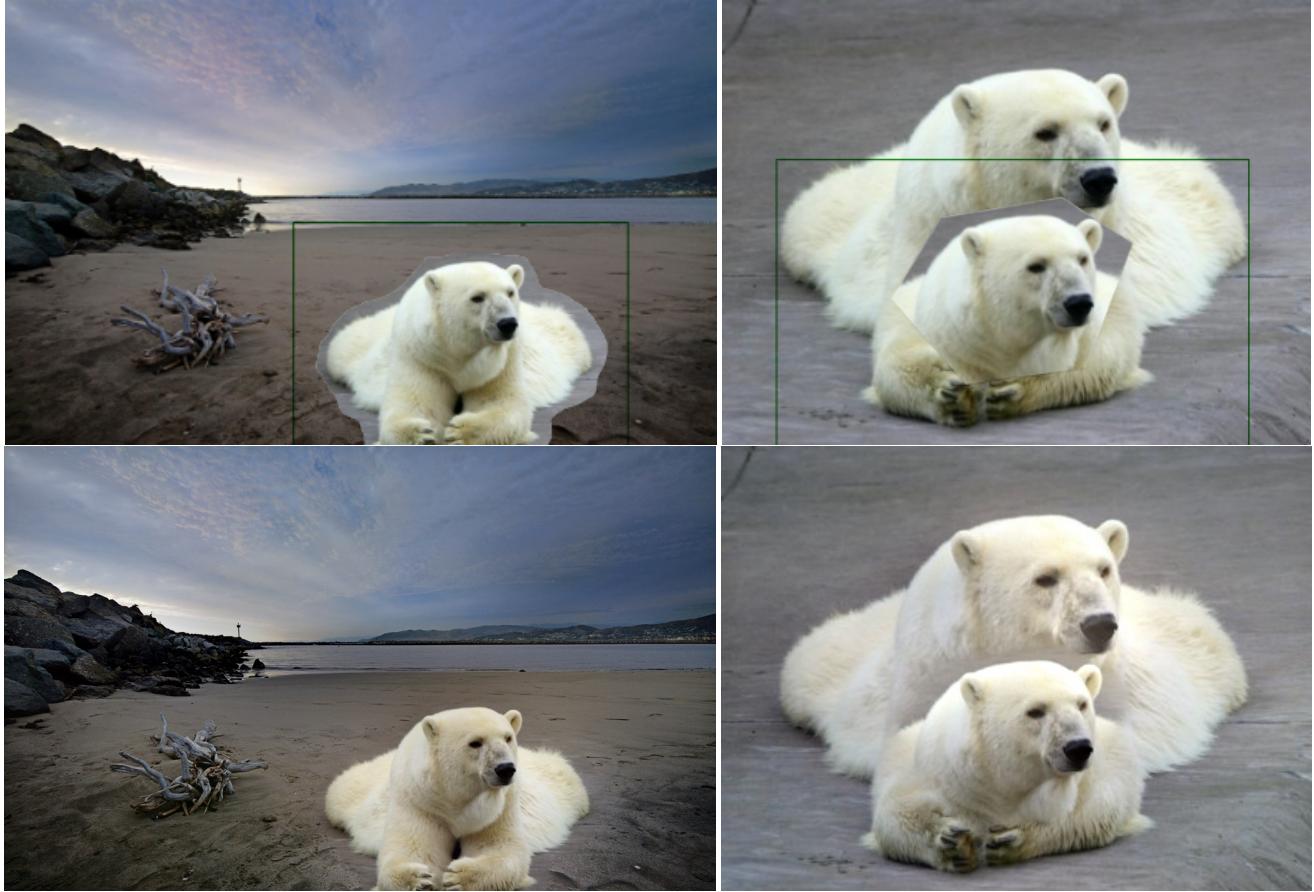


Figure 10: Results of GP-GAN on real images. The top is the copy-and-paste images and the bottom is the blended images. Best viewed in color.

while preserving the appearance of both x_{src} and x_{dst} . Compared to the baseline methods, there are nearly no bleedings or illumination inconsistencies in our results while all the baseline methods have more or fewer bleedings and artifacts.

Our method can also be applied to real images in high resolution, as shown in Figure 10.

5 CONCLUSION

We advanced the state-of-the-art in image blending by combining the ideas from the generative model GANs and gradient-based approaches. Our insight is, on the one hand, GANs are good at generating natural images from a particular distribution but weak in capturing the high-frequency image details like textures and edges. On the other hand, the gradient-based methods perform well at generating high-resolution images with local consistency, although the generated images tend to be unnatural and have many artifacts. GANs and gradient-based methods should be integrated. Hence, this integration would result in an image blending system that overcomes the drawbacks of both approaches. Our system can also be useful for image-to-image translation task. Despite the effectiveness, our algorithm fails to generate realistic images when

the composite images are far away from the distribution of the training dataset. We aim to address this issue in future work.

ACKNOWLEDGMENTS

This work is funded by the National Natural Science Foundation of China (Grand No. 61876181, 61721004, 61403383) and the Projects of Chinese Academy of Sciences (Grand QYZDB-SSW-JSC006 and Grand 173211KYSB20160008).

REFERENCES

- [1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. 2004. Interactive digital photomontage. *ACM Transactions on graphics (TOG)* 23, 3 (2004), 294–302.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*. 214–223.
- [3] Peter Burt and Edward Adelson. 1983. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications* 31, 4 (1983), 532–540.
- [4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2172–2180.
- [5] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in Neural Information Processing Systems (NIPS)*. 1486–1494.

- [6] Alexey Dosovitskiy and Thomas Brox. 2016. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*. 658–666.
- [7] Raanan Fattal, Dani Lischinski, and Michael Werman. 2002. Gradient domain high dynamic range compression. *ACM Transactions on graphics (TOG)* 21, 3 (2002), 249–256.
- [8] Robert T. Frankot and Rama Chellappa. 1988. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 10, 4 (1988), 439–451.
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2672–2680.
- [10] Nuno Gracias, Mohammad Mahoor, Shahriar Negahdaripour, and Arthur Gleason. 2009. Fast image blending using watersheds and graph cuts. *Image and Vision Computing* 27, 5 (2009), 597–607.
- [11] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. In *International Conference on Machine Learning (ICML)*. 1462–1471.
- [12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- [13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976.
- [14] J. Jia, J. Sun, C.K. Tang, and H.Y. Shum. 2006. Drag-and-drop pasting. *ACM Transactions on graphics (TOG)* 25, 3 (2006), 631–637.
- [15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *The European Conference on Computer Vision (ECCV)*. 694–711.
- [16] M. Kazhdan and H. Hoppe. 2008. Streaming multigrid for gradient-domain operations on large images. *ACM Transactions on graphics (TOG)* 27, 3 (2008), 1–10.
- [17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [18] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. 2014. Transient Attributes for High-Level Understanding and Editing of Outdoor Scenes. *ACM Transactions on graphics (TOG)* 33, 4 (2014), 149.
- [19] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. 2004. Seamless image stitching in the gradient domain. In *The European Conference on Computer Vision (ECCV)*. 377–389.
- [20] Chuang Li and Michael Wand. 2016. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. In *The European Conference on Computer Vision (ECCV)*. 702–716.
- [21] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
- [22] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2536–2544.
- [23] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Transactions on graphics (TOG)* 22, 3 (2003), 313–318.
- [24] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
- [25] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative Adversarial Text to Image Synthesis. In *International Conference on Machine Learning (ICML)*. 1060–1069.
- [26] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision* 77, 1-3 (2008), 157–173.
- [27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi. Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NIPS)*. 2234–2242.
- [28] Richard Szeliski, Matthew Uyttendaele, and Drew Steedly. 2011. Fast poisson blending using multi-splines. In *The IEEE International Conference on Computational Photography (ICCP)*. 1–8.
- [29] Richard Szeliski, Matthew Uyttendaele, and Drew Steedly. 2011. Fast poisson blending using multi-splines. In *The IEEE International Conference on Computational Photography (ICCP)*.
- [30] Masayuki Tanaka, Ryo Kamio, and Masatoshi Okutomi. 2012. Seamless image cloning by a closed form solution of a modified poisson problem. In *SIGGRAPH Asia*. 15.
- [31] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems in NIPS*. 1–6.
- [32] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. 2017. Deep image harmonization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [33] Matthew Uyttendaele, Ashley Eden, and Richard Szeliski. 2001. Eliminating ghosting and exposure artifacts in image mosaics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2.
- [34] Xiaolong Wang and Abhinav Gupta. 2016. Generative image modeling using style and structure adversarial networks. In *The European Conference on Computer Vision (ECCV)*. 318–335.
- [35] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. 2012. Understanding and improving the realism of image composites. *ACM Transactions on graphics (TOG)* 31, 4 (2012), 84.
- [36] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S. Paek, and In So Kweon. 2016. Pixellevel domain transfer. In *The European Conference on Computer Vision (ECCV)*. 517–532.
- [37] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242* (2017).
- [38] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. 2017. Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging* 3, 1 (2017), 47–57.
- [39] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2015. Learning a discriminative model for the perception of realism in composite images. In *The IEEE International Conference on Computer Vision (ICCV)*. 3943–3951.
- [40] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *The European Conference on Computer Vision (ECCV)*. 597–613.

A. VISUAL RESULTS

More visual results are shown in Figure 11 and Figure 12.

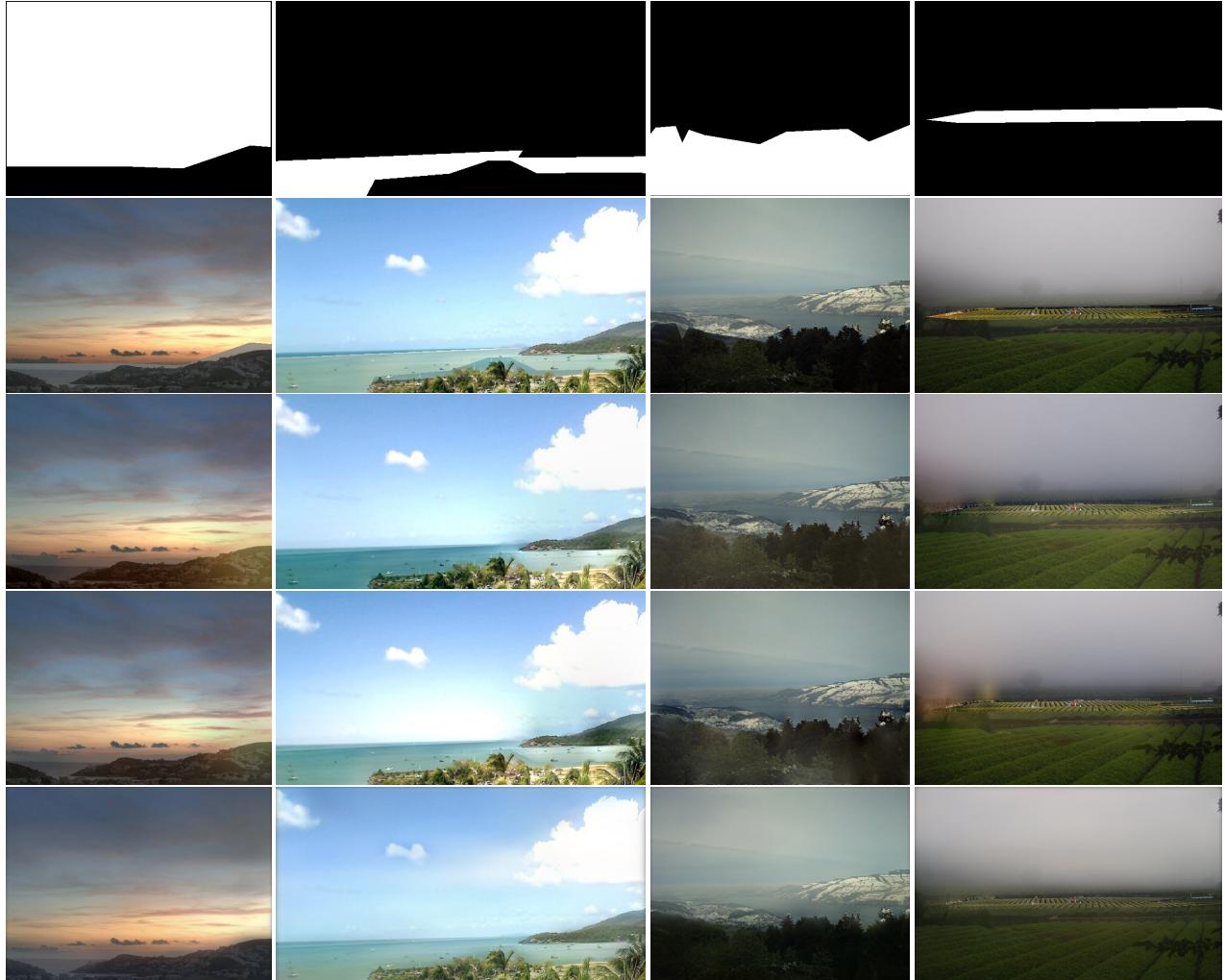


Figure 11: Results of our high-resolution image blending algorithm compared with baseline methods. From top to bottom: annotated object-level mask, composite copy-and-paste image, MPB, MSB, and GP-GAN.

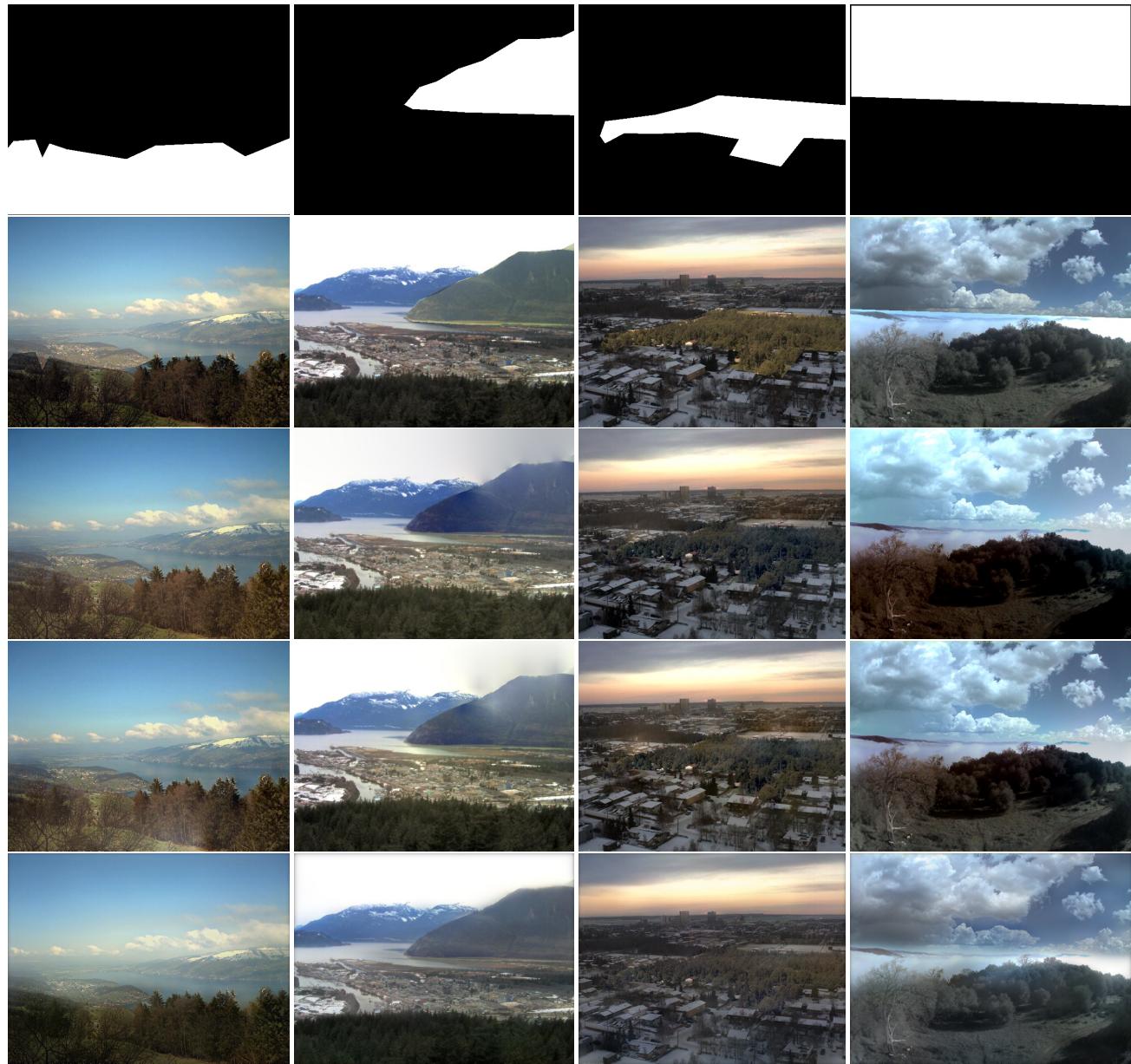


Figure 12: Results of our high-resolution image blending algorithm compared with baseline methods. From top to bottom: annotated object-level mask, composite copy-and-paste image, MPB, MSB, and GP-GAN.