Supplementary Material to Neural Texture Synthesis with Guided Correspondence

Yang Zhou Kaijian Chen Rongjun Xiao Hui Huang*
Visual Computing Research Center, Shenzhen University
{zhouyangvcc, szchenkaijian, xiaorongjun000, hhzhiyan}@gmail.com

This supplementary material contains the appendix of our paper and a part of additional results. For more supplemental results in high resolution, please check our web-page at https://github.com/EliotChenKJ/Guided-Correspondence-Loss.

A. Difference to the Contextual loss

Mechrez *et al*. [6] proposed the Contextual loss for image transformation with non-aligned data. Heitz *et al*. [5] proved it unsuitable for texture synthesis but didn't analyze why. In this section, we will try to figure out the reason and make clear the difference between the Contextual loss and our Guided Correspondence loss (besides the difference in features involved, since we considers the matching diversity and additional guidance in the patch distance).

The Contextual loss measures image similarity based on semantic features. The motivation behind its design is to search for correspondence pairs that are significant enough between all source and target samples. Briefly, it first considers the similarity of a target sample with respect to the context of the entire source image. Then all the target samples are involved as the context for each source sample, and the overall similarity is defined therefrom.

To be more specific, as illustrated in Figure 1, the first step of Contextual loss is to find the nearest neighbor for each target/output sample t_i. Then the similarity of t_i to all source/reference samples in S is normalized by the similarity of t_i to its nearest neighbor. After the normalization, the second step is to choose the correspondence with the maximum similarity among all target samples for each source sample s_j. Finally, these similarities are summed up as the contextual similarity between the entire S and T.

The so-designed Contextual loss may result in ill matches for some target samples: 1) There might be more than one match for some target samples, and 2) There might be no match found for some target samples. Figure 1 shows an illustration of this situation (see also Figure 3 in [6]). These configurations are considered reasonable for image transformation since the style reference could be very different from the (natural) image to be optimized. If the correspondence pair is not significant enough, we keep the target sample unchanged.

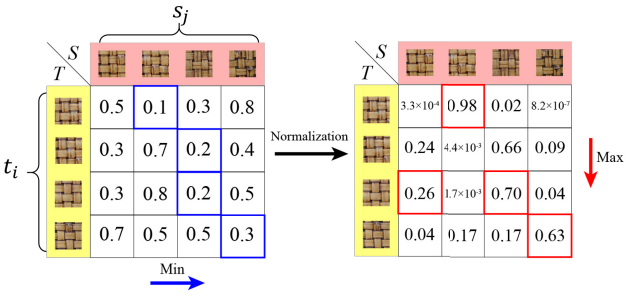

Figure 1. The computation of Contextual loss. For illustration, we visualize the similarity/distance matrix between all the target and source samples. The Contextual loss first searches the nearest neighbor and normalizes the similarity matrix along the horizontal direction. Then it finds the maximum similarity along the vertical path. This design, however, may leave some target samples unmatched or matched to more than one source sample, leading to under-optimization or inconsistent optimization in the output. We only adopt the first step of Contextual loss (search and normalize horizontally) to build our Guided Correspondence loss.

However, these ill matches may cause severe artifacts in texture optimization. As shown in Figure 2, we can see many pixels in the results produced by using the Contextual loss look like noise, which means they are either under-optimized or over-optimized (i.e., optimized by different source samples).

Our Guided Correspondence loss, in contrast, is built upon the MRF criterion. We aim to optimize every target patch to be similar to a particular source patch. Since we also hope the target sample is significantly closer to its nearest neighbor than to all other source samples, we adopt the first step of the Contextual loss, which corresponds to the search and normalization along the horizontal direction of the similarity matrix.

*Corresponding author.

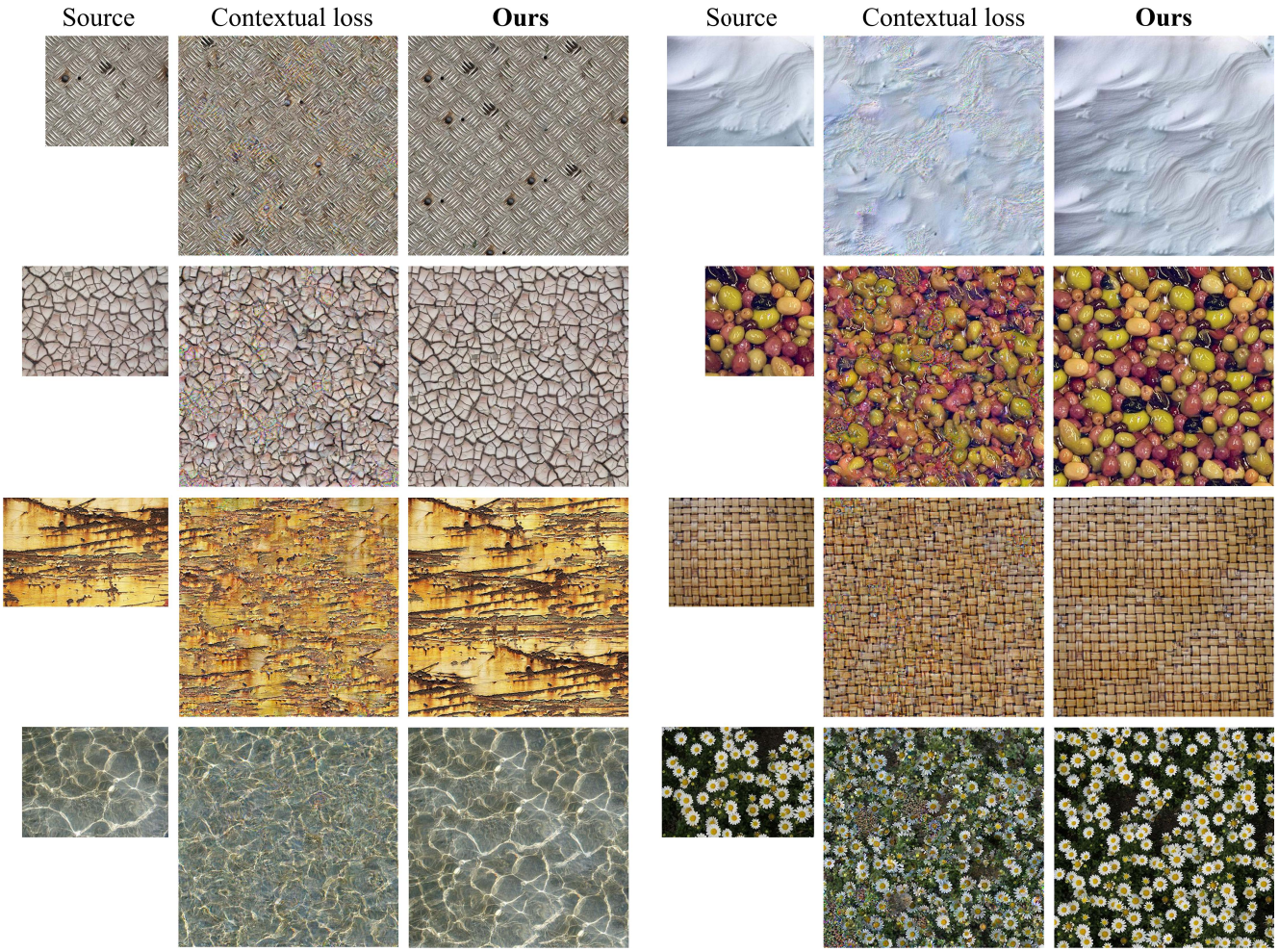

Figure 2. Comparison with the Contextual loss [6] on uncontrolled texture synthesis. We use pretrained VGG-19 as the feature extractor, based on which we compute the Contextual loss of the output texture to the input example. Then we optimize the output pixels via back-propagation. Severe artifacts can be seen in their results due to under-optimization or inconsistent optimization of target patches.

B. Details of Training SPADE for Real-time Controlled Synthesis

SPADE [7] is a conditional generative model proposed recently. Briefly, SPADE takes a semantic label map as the condition to modulate the feature activations in the generator to make the output image follow the given semantic layout. We adopt it for real-time synthesis with progression or orientation control maps as the condition. Take progression control as an example. We directly replace the label maps with the progression maps. The challenge here is that the network must have a strong generalization power, given that the source progression is probably very different from the user-specified target. To address that, we separate the training into two stages.

As depicted in Figure 3, the first stage is reconstruction training. We train 20k iterations to let the model learn how to reconstruct the source textures from the source progression maps. Specifically, in each iteration, we randomly crop a block from the source and its corresponding guidance (256×256 pixels), and train SPADE without modification (including the discriminator and the losses).

The second stage is training for random synthesis. For that, we generate a massive number of random target progression maps from 1) source guidance perturbed by Perlin noise [8], 2) source guidance stretched or rotated randomly, and 3) pure Perlin noise. We also run 20k iterations for the random synthesis. The proportions of the three types of augmented target maps are 10%, 70%, and 20%, respectively. Since now the generated image has no "ground truth" for the random synthesis, the Guided Correspondence loss is necessary as proved in the paper as a complementary to the conditional-GAN loss. More results are included in the folder of "Supplementary".

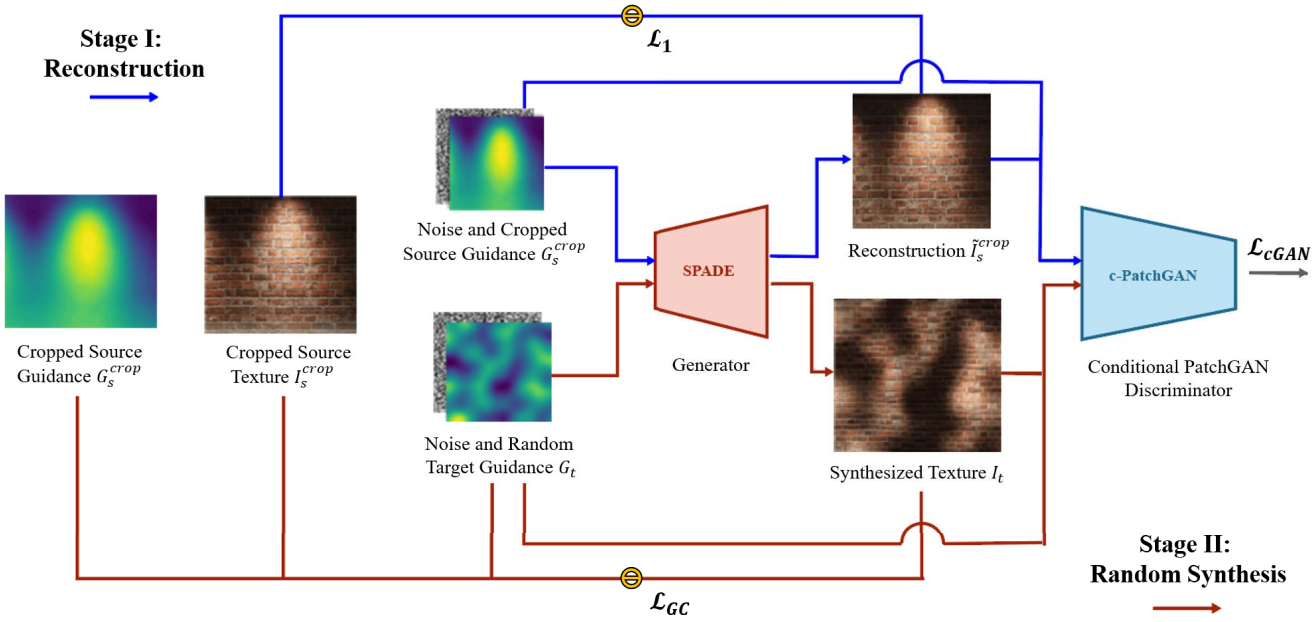

Figure 3. Utilizing the Guided Correspondence loss in training SPADE [7] for real-time controlled synthesis.

C. Details of Inversion-based Single-image Editing

Recent single-image models [2, 3, 9] can be applied for editing tasks but struggle to synthesize images of non-repetitive objects [11]. To address that, Wang *et al*. [11] proposed IMAGINE, a classifier-based inversion model to regularize the image semantics of the synthesized image. Specifically, they use ResNet50 [4] pre-trained on ImageNet [1] to compute classification errors between the input and output image. Through error back-propagation, the semantics of the synthesized image is optimized to be object-like. Then, to synthesize fine-level image details, IMAGINE involves a patch-based discriminator same as SinGAN does, which is trained adversarially during the optimization (IMAGINE optimizes the output and the discriminator in an iterative turn). However, we found this ad-hoc training of PatchGAN discriminator cannot guarantee fine textures for the output. Considering the patch-based MRF property of our Guided Correspondence loss, we remove the discriminator, and compute the Guided Correspondence loss to enforce the patch coherence across the synthesized image; see Figure 4 for the illustration of our inversion-based single-image editing.

Figure 5 show more results of single-image editing using the combination of classification loss ($\mathcal{L}_{Class}$) and the proposed Guided Correspondence loss ($\mathcal{L}_{GC}$). Actually, there's another very recent work, MAGIC [10], proposed by Rouhsedaghat *et al*. on this task. They introduce a pre-trained *quasi-robust* classifier to provide stronger semantic regularization, and an additional pre-trained segmentation

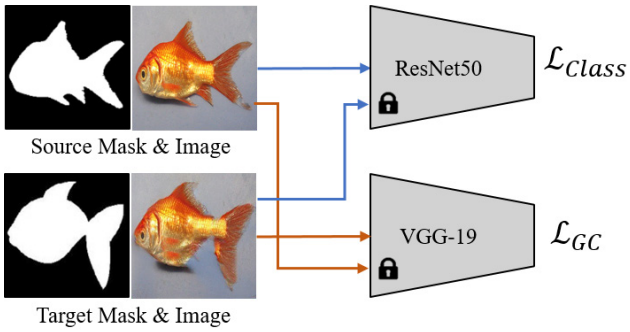

Figure 4. Based on the classifier-based inversion model of [11], we propose to combine classification loss and the Guided Correspondence loss to optimize the output image

network to enforce the synthesis following the mask guidance. We compare our editing to this "enhanced" inversion model, and found our results seemly have fewer artifacts; see Figure 6.

D. Additional Results and Comparisons

Figure 7 shows uncontrolled synthesis results and comparison with state-of-the-art methods in the high resolution. Figures 8, 9 and 10 show results of controlled synthesis with different guidance channels, as well as the comparisons with Zhou *et al*. [13].

Figure 11 shows two examples of very challenging non-stationary textures from TexExp [14]. Our method cannot handle well these challenging textures with global struc-

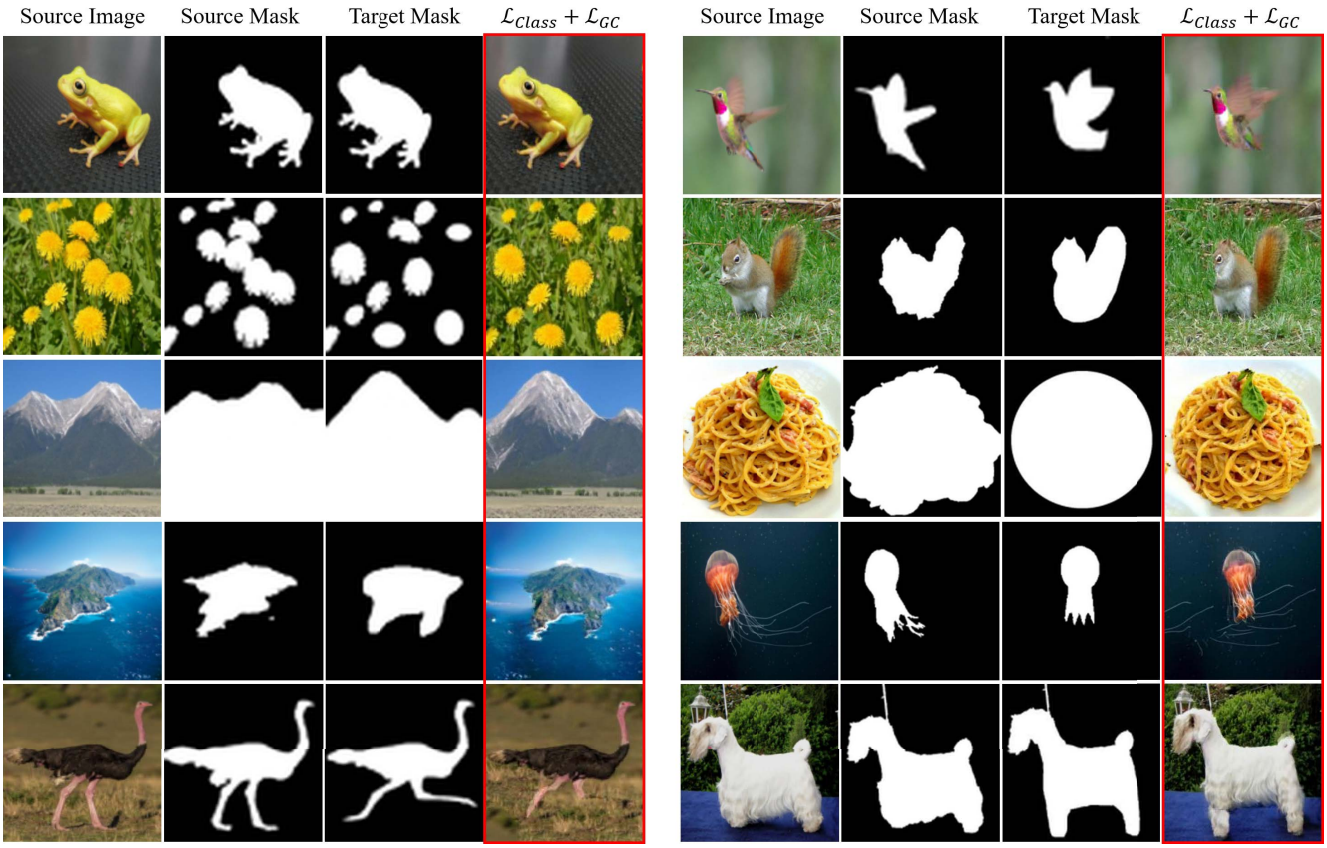

Figure 5. More results of utilizing our Guided Correspondence loss for single-image editing with mask control. All the binary masks are cropped from the preprint work [10].

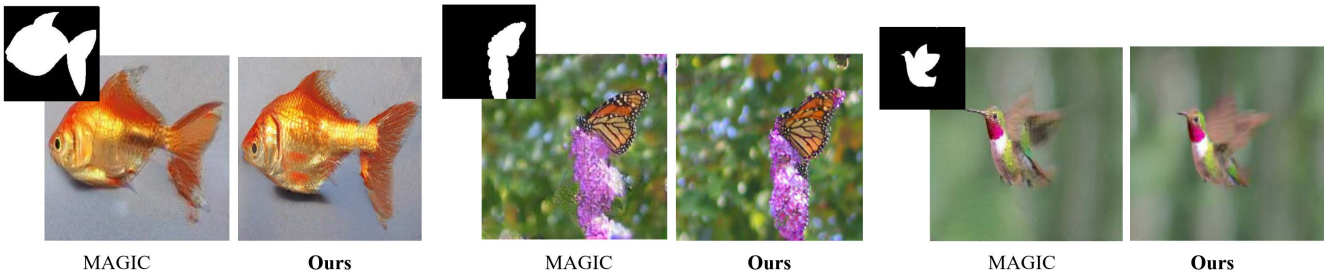

Figure 6. Comparison with a preprint work, MAGIC [10] on mask-guided single-image editing. Note the results of MAGIC are directly cropped from their paper as the code is currently unavailable. Many artifacts can still be seen in their results if we check carefully.

tures in uncontrolled scenarios. TexExp succeeds since its encoder extracts a structural guidance for its decoder. Nevertheless, when guidance channels are provided explicitly, our method can also produce plausible results.

E. LPIPS score

Shortly speaking, LPIPS [12] calculates the L_2 distance between the activations of images from pre-trained VGG so that measures perceptual similarity. It is demonstrated to match human perception well. Given the semantic characteristic of deep features, LPIPS is content invariant and robust to local distortions and color shifts. In texture synthesis, however, our goal is to synthesize a new texture that is usually larger than and, more importantly, different from the exemplar in content (such as layout or patch distribution). It is unsuitable to directly compute the LPIPS score between the source texture and the synthesized output. We thus compute it as we did for color distance. We randomly crop 1000 patches from both the target and the source texture.

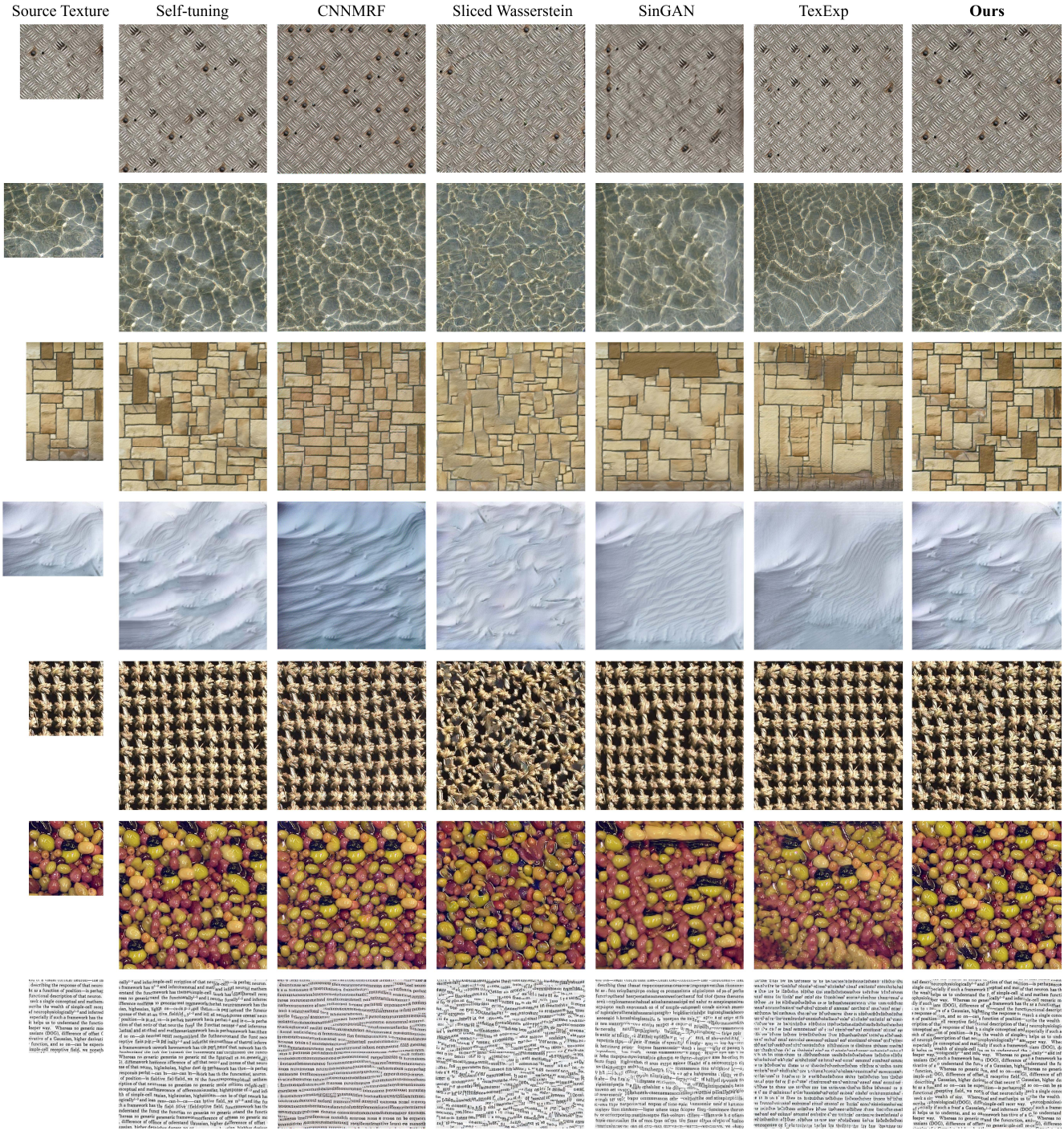

Figure 7. Uncontrolled synthesis of stationary textures and comparison with state-of-the-art approaches.

Then for each target patch, we calculate its minimum LPIPS score among the 1000 source patches. The average score of all target patches is regarded as the final distance between the output and its source texture. As listed in Table 1, our method is totally comparable to Self-tuning (closer than using the color distance) in LPIPS metric. Comparing to other deep approaches, our results possess higher perceptual realism, demonstrating again the synthesis quality of our method.

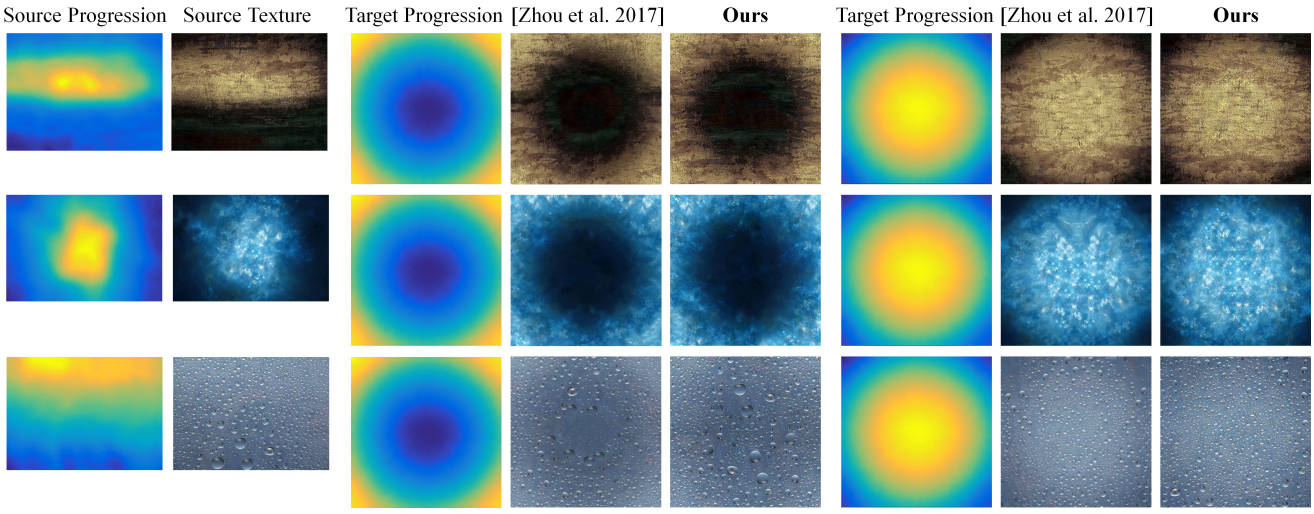

Figure 8. Progression control and comparison with Zhou *et al*. [13].

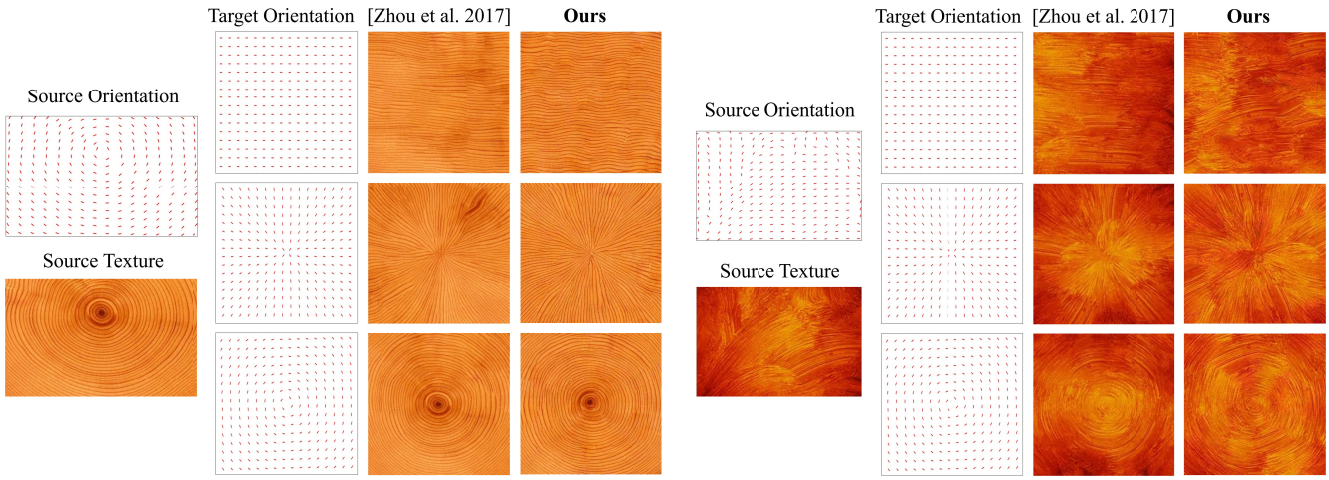

Figure 9. Orientation control and comparison with Zhou *et al*. [13].

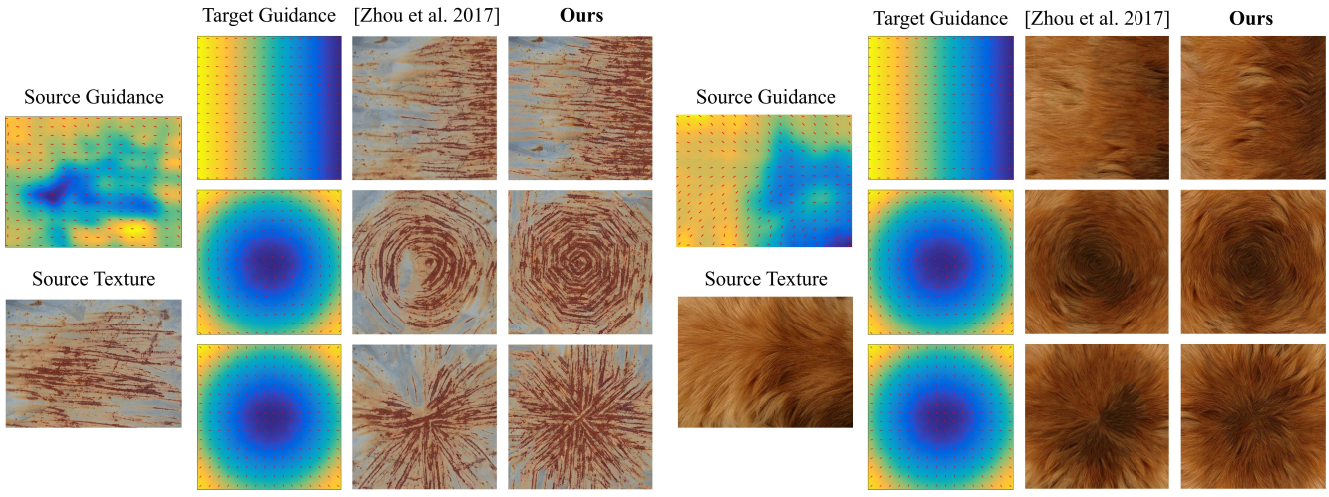

Figure 10. Two controls (progression and orientation control simultaneously) and comparison with Zhou *et al*. [13].

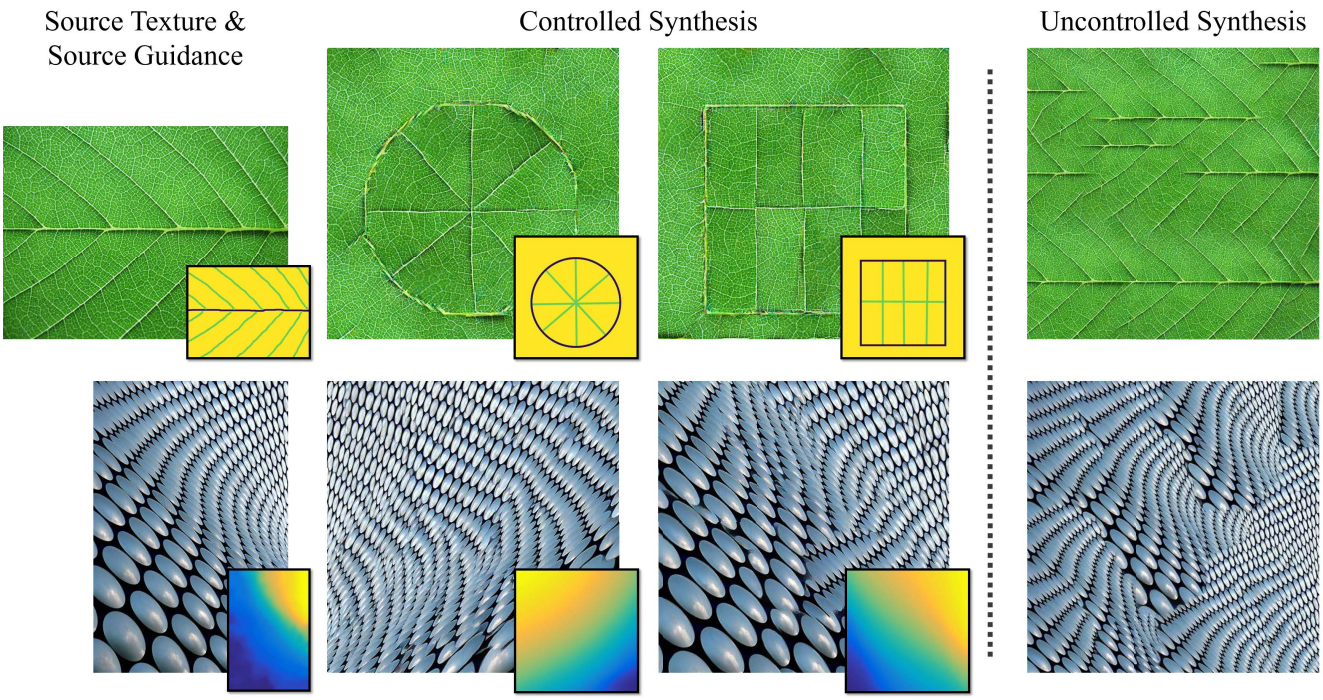

Figure 11. Our results on very challenging non-stationary textures with global structures from TexExp [14]. Although our method cannot handle them well in uncontrolled scenarios (right), when guidance channels are provided explicitly, our approach can also produce plausible results following user-specified structures (left).

Table 1. Comparison on LPIPS metric with state-of-the-art methods for uncontrolled synthesis (smaller is better).

PatchSize	Self-tuning	CNNMRF	SWD	SinGAN	TexExp	Ours
64	**0.389**	0.420	0.462	0.456	0.447	0.391
128	0.535	0.560	0.597	0.595	0.581	**0.534**

Table 2. Average running time of different methods for uncontrolled synthesis. We randomly pick 10 images from the dataset, use the default settings for all methods, and record the running time (in seconds). Note for the two GAN-based methods (SinGAN & TexExp), we count their total time for training.

Self-tuning	CNNMRF	SWD	SinGAN	TexExp	Ours
226.18	210.18	484.60	(5160.71)	(6180.24)	313.28

F. Complexity and Limitation

The complexity of our loss is $O(N^2)$, where N is the number of patches that we sampled on the target image. The complexity is at the same level as Self-tuning, CNNMRF and the SWD loss, as reflected from the average running time shown in Table 2. While for the training efficiency of real-time synthesis, it takes about 40 mins (running 3k iterations) using our loss to train TextureNets and about 10 hours (running 40k iterations) to train SPADE.

The $O(N^2)$ complexity, however, brings the main limitation to our method. Unlike traditional PatchMatch, which only needs to store a few paras about the corresponding patch, we have to save all the patch distances for the computation of contextual similarity. We can therefore have a limited number of augmentation copies to enlarge the search space. For those textures containing diversely oriented patches, such as the wood ring and dog fur, our method can synthesize with acceptable smoothness and continuity in orientation; see results shown in Figures 9 and 10. When an exemplar only has monotonous oriented patches, such as the bamboo and the rusty scratches (Figure 10 (left)), the orientation change would be abrupt in the results.

There are two possible solutions to overcome this limitation. One is to constrain the computation within a local context. The other solution is to manually rotate the searched source patch to follow the target orientation field after nearest neighbor search.

This limitation also introduces artifacts to results of image editing in some cases. As shown in Figure 12, the right leg of Ostrich is missed due to the limited data augmentation. Then we can see the leg is coming back after we turn on the rotation augmentation (8 copies as we did for orientation control). But since this image is not a texture image, the rotated patches also affected the background near the foreground boundaries. We leave this issue for future work.

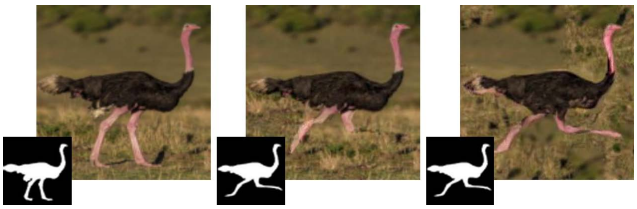

Figure 12. Left: input image; Middle: artifacts can be seen on the right leg of the Ostrich in single image editing if we use the default setting (without rotation augmentation); Right: after turning on the rotation augmentation, the leg is coming back (right).

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, pages 248–255. IEEE Computer Society, 2009. 3

[2] Ariel Elnekave and Yair Weiss. Generating natural images with direct patch distributions matching. In *Proc. of Euro. Conf. on Computer Vision*, volume 13677, pages 544–560. Springer, 2022. 3

[3] Niv Granot, Assaf Shocher, Ben Feinstein, Shai Bagon, and Michal Irani. Drop the GAN: in defense of patches nearest neighbors as single image generative models. *CoRR*, abs/2103.15545, 2021. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, pages 770–778. IEEE Computer Society, 2016. 3

[5] Eric Heitz, Kenneth Vanhoey, Thomas Chambon, and Laurent Belcour. A sliced wasserstein loss for neural texture synthesis. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, June 2021. 1

[6] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proc. of Euro. Conf. on Computer Vision*, 2018. 1, 2

[7] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, 2019. 2, 3

[8] Ken Perlin. An image synthesizer. In *Proc. of SIGGRAPH*, page 287–296, 1985. 2

[9] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proc. of Int. Conf. on Computer Vision*, 2019. 3

[10] Mozhdeh Rouhsedaghat, Masoud Monajatipoor, Kai-Wei Chang, C.-C. Jay Kuo, and Iacopo Masi. MAGIC: mask-guided image synthesis by inverting a quasi-robust classifier. *CoRR*, abs/2209.11549, 2022. 3, 4

[11] Pei Wang, Yijun Li, Krishna Kumar Singh, Jingwan Lu, and Nuno Vasconcelos. IMAGINE: image synthesis by image-guided model inversion. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3681–3690. Computer Vision Foundation / IEEE, 2021. 3

[12] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, 2018. 4

[13] Yang Zhou, Huajie Shi, Dani Lischinski, Minglun Gong, Johannes Kopf, and Hui Huang. Analysis and controlled synthesis of inhomogeneous textures. *Computer Graphics Forum (Proc. of Eurographics)*, 36(2):199–212, 2017. 3, 6

[14] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 37(4):49:1–49:13, 2018. 3, 7