Long Range Constraints for Neural Texture Synthesis Using Sliced Wasserstein Loss

TLDR

- Texture Synthesis: Goal is to generate a realistic, yet different, version of a reference texture.
- We propose a modified Sliced Wasserstein Loss to capture long-range constraints in neural texture synthesis.
- An additional height-dimension loss term/multi-scale approach to improve structure without manual masks or alternative regularization terms.

Texture Synthesis and SW Loss

- Let layer ℓ of an L layer convolutional neural network have N_ℓ channels and M_ℓ pixels in each channel.
- $p^\ell, \hat{p}^\ell$: probability density functions for vectors $\{F^\ell_m\}$ and $\{\hat{F}^\ell_m\}$ associated to images I_1 and I_2.
- We assume that the probability density functions take the form

$$p^\ell(x) = \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} \delta_{F^\ell_m}(x). \tag{1}$$

The Sliced Wasserstein Loss between two images, $\{w_\ell\}$ are weight terms

$$\mathcal{L}_{\text{SW}}(I_1, I_2) = \sum_{\ell=1}^{L} w_\ell \mathcal{L}_{\text{SW},\ell}(p^\ell, \hat{p}^\ell), \tag{2}$$

where the Sliced Wasserstein Distance between two feature distributions is given by

$$\mathcal{L}_{\text{SW},\ell}(p^\ell, \hat{p}^\ell) = \mathbb{E}_V[\mathcal{L}_{\text{SW1D}}(p^\ell_V, \hat{p}^\ell_V)]. \tag{3}$$

Let V be a random direction on the unit sphere of dimension N_ℓ. Here, we define (with corresponding definitions for $\hat{p}^\ell$) $p^\ell_V := \{\langle F^\ell_m, V\rangle\}$ as a set consisting of batched projections of the feature maps F^ℓ_m onto V; define vector P^ℓ_V consisting of the elements of p^ℓ_V and

$$\mathcal{L}_{\text{SW1D}}(p^\ell_V, \hat{p}^\ell_V) = \frac{1}{\text{len}(P^\ell_V)} \left\| \text{sort}(P^\ell_V) - \text{sort}(\hat{P}^\ell_V) \right\|_2^2. \tag{4}$$

- Convolution operators are local, so (1) ignores correlations between distant pixels (e.g. long-range structure).
- Consider a set of feature maps $F^\ell \in \mathbb{R}^{H_\ell \times W_\ell \times N_\ell}$ and a feature vector of shape $W_\ell \times N_\ell$, which we denote by $F^\ell_{H,n} \in \mathbb{R}^{H_\ell}$, where $n \in \{1, \ldots, W_\ell \times N_\ell\}$.
- We have another set of probability distributions to match, which incorporate locality:

$$p^\ell_H(x) = \frac{1}{W_\ell \times N_\ell} \sum_{n=1}^{W_\ell \times N_\ell} \delta_{F^\ell_{H_n}}(x). \tag{5}$$

Consider distributions p^ℓ_H and $\hat{p}^\ell_H$ associated with I_1 and I_2. The corresponding additional loss term is

$$\mathcal{L}_{\text{SW},H}(I_1, I_2) = \sum_{\ell=1}^{L} w_\ell \mathcal{L}_{\text{SW},\ell}\left(p^\ell_H, \hat{p}^\ell_H\right), \tag{6}$$

Texture Synthesis and SW Loss Continued

We minimize

$$\mathcal{L}_{\text{Slicing}}(I_1, I_2) = \mathcal{L}_{\text{SW}}(I_1, I_2) + \mathcal{L}_{\text{SW},H}(I_1, I_2). \tag{7}$$

Algorithm 1: Synthesis Algorithm

Denote the feature map extraction of image I from VGG19 as Extract(I);
$I_{\text{WN}} \leftarrow$ white noise to be updated by optimizer via backpropagation;
$I_{\text{Ref}} \leftarrow$ reference texture;
for $k \leftarrow 1$ **to** M ; // Epochs
do
Calculate Extract(I_{WN});
Calculate Extract(I_{Ref});
Calculate $\mathcal{L}_{\text{Slicing}}(I_{\text{WN}}, I_{\text{Ref}})$;
Backpropagate and update I_{WN};
return I_{WN} as synthesized texture;

Example Results

For more periodic textures (first two rows), the performance varies between algorithms, but proposed algorithm has more consistent performance on nonstationary textures (last two rows).

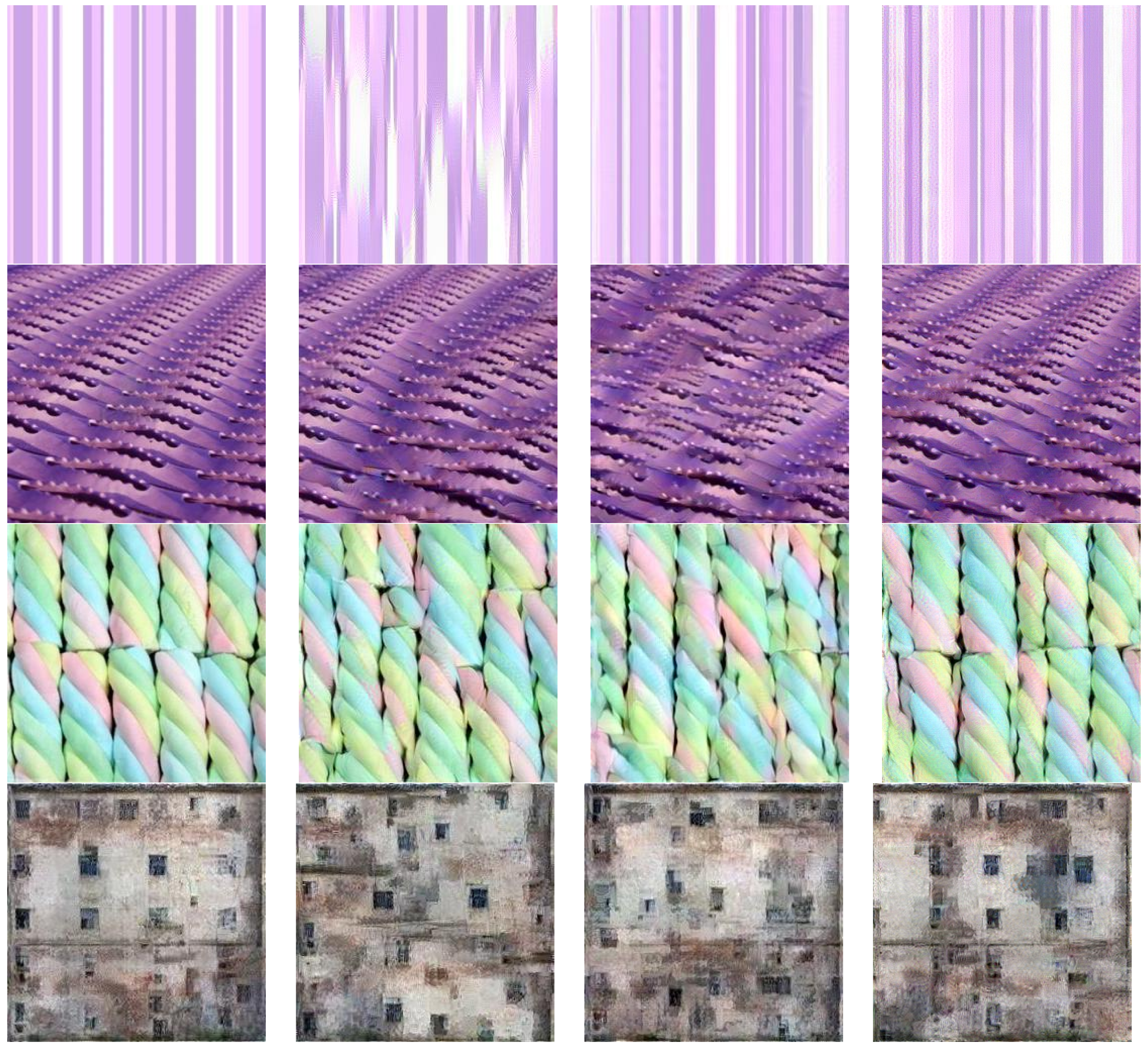

Fig. 1: Comparison of results for textures. **Left:** Reference. **Mid Left:** SW Loss. **Mid Right:** Spectrum. **Right:** Using new loss (Ours).

Multi-scale Algorithm

To improve the results of our algorithm, we incorporate a multi-scale approach previously seen in [1].

Algorithm 2: Multi-scale Algorithm

Initialize $I_{\text{Synthesis}}$ as white noise, equal to the reference texture downsampled by 2^K;
Let $I_{\text{ref},i}$ be the reference texture downsampled by 2^i;
Let `SWSynthesis` be synthesis using Algorithm 1;
for $i \leftarrow 0$ **to** K **do**
$I_{\text{Synthesis}} \leftarrow \texttt{SWSynthesis}(I_{\text{Synthesis}}, I_{\text{ref},K-i})$;
$I_{\text{Synthesis}} \leftarrow 2 \times \texttt{Upsample}(I_{\text{Synthesis}})$;
return $I_{\text{Synthesis}}$ as the synthesized texture;

More scales, means better synthesis, but possible repetition.

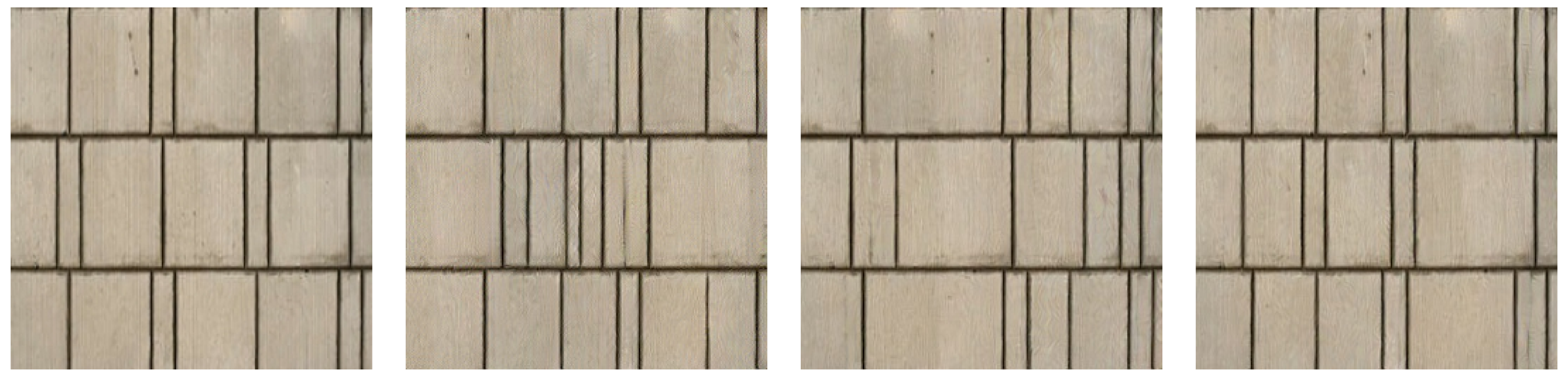

Fig. 2: Multi-scale procedure at different scales. **Left:** Reference. **Mid Left:** $K = 0$. **Mid Right:** $K = 1$. **Right:** $K = 2$.

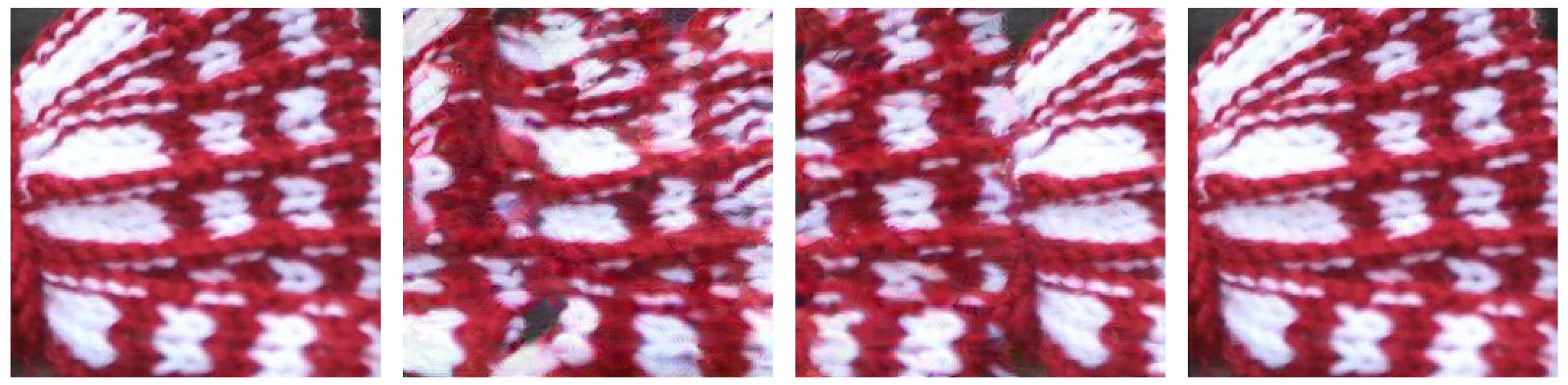

Fig. 3: Progression of synthesis that lead to repetitions. **Left:** Reference Texture. **Middle Left:** $K = 0$. **Middle Right:** $K = 1$. **Right:** $K = 2$.

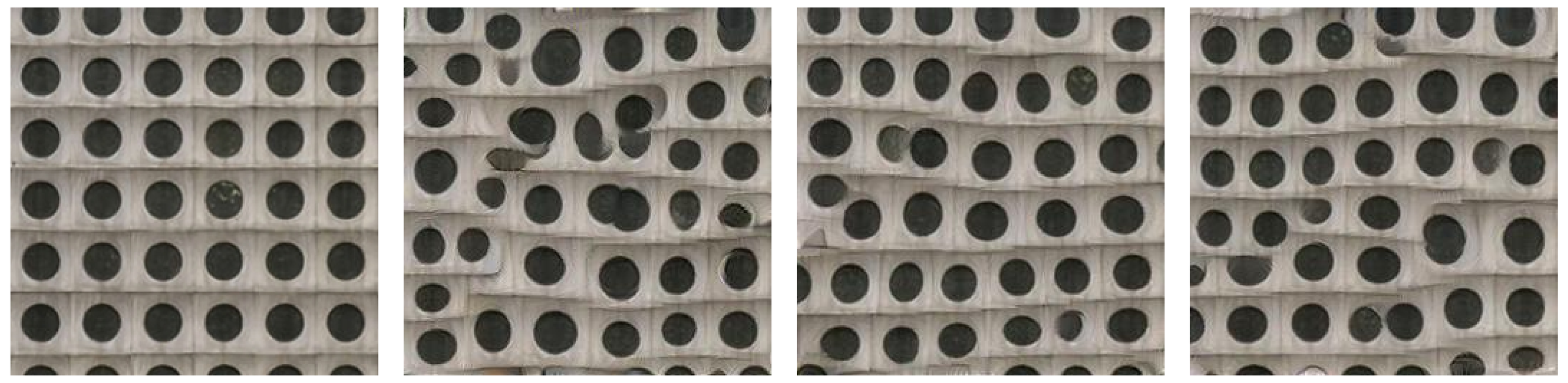

Fig. 4: Ablation study: results with only (2) **Left:** Reference. **Mid Left:** $K = 0$. **Mid Right:** $K = 1$. **Right:** $K = 2$.

- Quant comparison with other methods using set of 34 textures.
- $K = 1$ shows the best results for 256×256 images.

Method	LPIPS	FID	c-FID	KID	c-KID
$K = 0$	$.44$	107.2	72.3 ± 0.34	$-.014$	$.073 \pm 0.0008$
SW	$.45$	101.8	78.7 ± 0.48	$-.016$	$.084 \pm 0.001$
Spec.	$.45$	99.6	78.3 ± 0.57	$-.016$	$.085 \pm 0.001$
Gonthier	$.42$	77.6	68.3 ± 0.35	$-.018$	$.069 \pm 0.0009$
$K = 1$	$.38$	67.1	53.5 ± 0.32	$-.018$	$.043 \pm 0.0006$
$K = 2$	$.25$	38.3	40.5 ± 0.6	$-.022$	0.028 ± 0.0009

Tab. 1: Perceptual Metric Comparison, (C-FID/C-KID scores [2] have SE for runs)

References

[1] Nicolas Gonthier, Yann Gousseau, and Saïd Ladjal. "High resolution neural texture synthesis with long range constraints," in *Journal of Mathematical Imaging and Vision 64:478–492*, 2022.

[2] Guilin Liu, Rohan Taori, Ting-Chun Wang, Zhiding Yu, Shiqiu Liu, Fitsum A. Reda, Karan Sapra, Andrew Tao, and Bryan Catanzaro "Transposer: Universal Texture Synthesis Using Feature Maps as Transposed Convolution Filter," in *arXiv:2007.07243 [cs.CV]*, July 2020.