Wasserstein Barycenter and its Application to Texture Mixing

Geometric Data Analysis

Antoine Ratouchniak

December 22, 2023

ENS Paris-Saclay

Table of contents

Introduction

Introduction

Texture mixing is a problem that aims to synthesize texture that closely looks like a target texture.

Target texture | Synthesized texture 1 | Synthesized texture 2

Figure 1: Texture synthesis (illustration from Efros et al. [1])

[1] Efros, A. A., & Freeman, W. T. (2023). Image quilting for texture synthesis and transfer. In Seminal Graphics Papers: Pushing the Boundaries, Volume 2 (pp. 571-576).

Definitions

Wasserstein distance

Setup:

Wasserstein distance

Setup:

- $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ where $\mathcal{P}_p$ denotes the set of probability measures with finite p-th moment

Wasserstein distance

Setup:

- $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ where $\mathcal{P}_p$ denotes the set of probability measures with finite p-th moment
- $d(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is a cost function (distance), usually $d(x, y) = \|x - y\|$

Wasserstein distance

Setup:

- $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ where $\mathcal{P}_p$ denotes the set of probability measures with finite p-th moment
- $d(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is a cost function (distance), usually $d(x, y) = \|x - y\|$
- $\Pi(\mu, \nu) \in \mathcal{P}_p(\mathbb{R}^d \times \mathbb{R}^d)$ is the set of all probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ that have marginals μ and ν

Wasserstein distance

Setup:

- $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ where $\mathcal{P}_p$ denotes the set of probability measures with finite p-th moment
- $d(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is a cost function (distance), usually $d(x, y) = \|x - y\|$
- $\Pi(\mu, \nu) \in \mathcal{P}_p(\mathbb{R}^d \times \mathbb{R}^d)$ is the set of all probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ that have marginals μ and ν

Definition: Wasserstein distance of order p

$$\mathrm{W}_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} d(x, y)^p d\pi(x, y) \right)^{1/p}$$

Wasserstein distance in the discrete case

Let $X, Y \subset \mathbb{R}^d$ be two spaces and $(x_i)_{1\leq i\leq N}, (y_j)_{1\leq j\leq M}$ two sequences of points

We have the discrete probability distributions μ and ν defined by

$$\mu = \sum_{i=1}^{N} \alpha_i \delta_{x_i} \quad \nu = \sum_{j=1}^{M} \beta_j \delta_{y_j}$$

Definition: Wasserstein distance of order p in the discrete case

$$\mathrm{W}_p(\mu, \nu) = \left(\min_{P \in \mathbb{R}_+^{N\times M}} \sum_{i=1}^{N} \sum_{j=1}^{M} P_{i,j} d(x_i, y_j)^p \right)^{1/p}$$

such that $(P\mathbf{1}_m)_i = \alpha_i$ and $(P^T\mathbf{1}_n)_j = \beta_j$

Wasserstein distance on point clouds

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i} \quad \nu = \frac{1}{N}\sum_{i=1}^{N}\delta_{y_j}$$

Definition: Wasserstein distance of order p on point clouds

$$\mathrm{W}_p(\mu,\nu) = \mathrm{W}_p(X,Y) = \left(\min_{\sigma\in\mathfrak{S}_N}\sum_{i=1}^{N} d(X_i, Y_{\sigma(i)})^p\right)^{1/p}$$

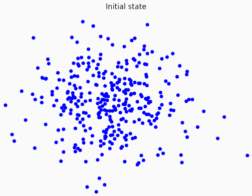


Figure 2: Gaussian point cloud to squirrel point cloud

Problem: computing the discrete Wasserstein distance when the number of samples grows can be complicated (complexity of $\mathcal{O}(n^{2.5})$).

Solution: reduce to the 1D case using the Sliced-Wasserstein distance where the solution is $\sigma^* = \sigma_Y \circ \sigma_X^{-1}$ that can be computed in $\mathcal{O}(n \log n)$.

Definition: Sliced-Wasserstein distance of order p on point clouds

Let $\mathbb{S}^{d-1} = \{\theta \in \mathbb{R}^d \mid \|\theta\|_2 = 1\}$ be the unit sphere in $\mathbb{R}^d$

$$\mathrm{SW}_p(X, Y) = \int_{\theta \in \mathbb{S}^{d-1}} \mathrm{W}_p(\langle X, \theta \rangle, \langle Y, \theta \rangle) d\theta \qquad \theta \sim \mathcal{U}(\mathbb{S}^{d-1})$$

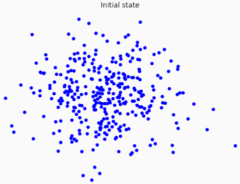


Figure 3: Gaussian point cloud to squirrel point cloud using the Sliced-Wasserstein distance

Wasserstein barycenter on point clouds

Definition: Wasserstein barycenter on point clouds (Agueh et al. [2])

Let $\{X_i \in \mathbb{R}^d\}_{i=1}^n$ be a sequence of point clouds and $(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_+$ such that $\sum_{i=1}^{n} \lambda_i = 1$

$$\mathrm{Bar}(\lambda_i, X_i)_{1 \le i \le n} = \underset{X}{argmin} \sum_{i=1}^{n} \lambda_i \mathrm{W}_p(X, X_i)$$

We define its sliced version by $\tilde{\mathrm{Bar}}(\lambda_i, X_i)_{1 \le i \le n} = \underset{X}{argmin} \sum_{i=1}^{n} \lambda_i \mathrm{SW}_p(X, X_i)$.

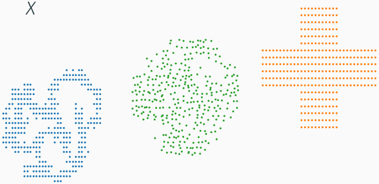


Figure 4: Sliced barycenter between two point clouds

[2] Agueh, M., Carlier, G. (2011). Barycenters in the Wasserstein space. SIAM Journal on Mathematical Analysis, 43(2), 904-924.

Wasserstein barycenter on point clouds

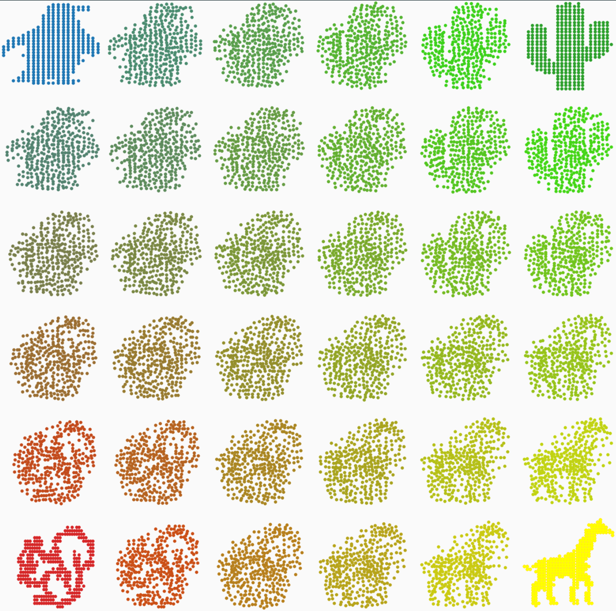


Figure 5: Wasserstein barycenter on point clouds

Texture Mixing

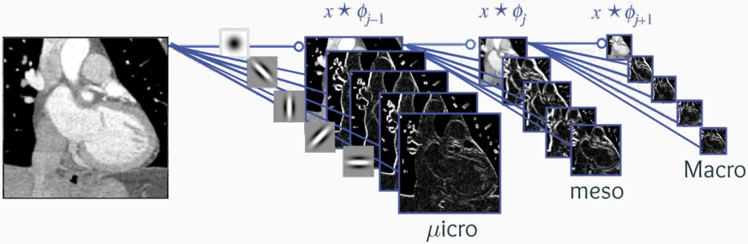
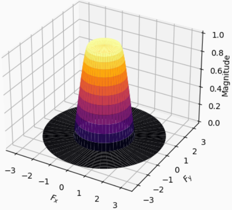


Figure 6: Wavelet transform (illustration from Feydy [3])

[3] Feydy, J. (2020). Geometric data analysis, beyond convolutions. Applied Mathematics.

Low-pass filter

Input image

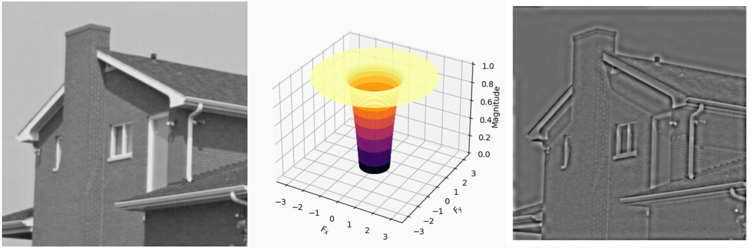


Low-pass filter in the Fourier domain

Low-pass filter applied on the image

Figure 7: Low-pass filter

High-pass filter



Input image

High-pass filter in the Fourier domain

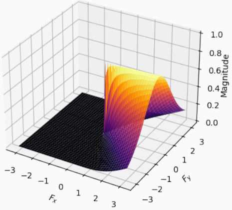
High-pass filter applied on the image

Figure 8: High-pass filter

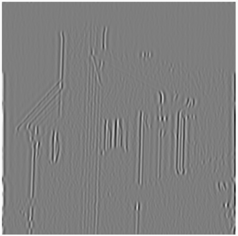
Band-pass oriented filter

Band-pass oriented filter

Input image



Band-pass oriented filter



Band-pass oriented filter applied on the image

Figure 9: Band-pass oriented filter

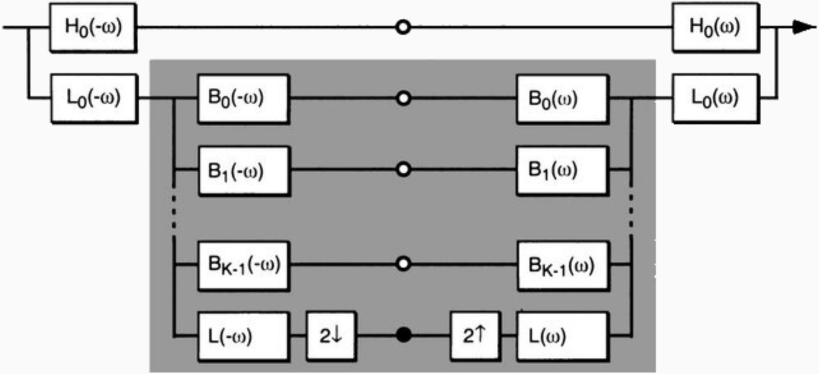


Figure 10: Steerable pyramid (illustration from Heeger & Bergen [4])

[4] Heeger, D. J., & Bergen, J. R. (1995, September). Pyramid-based texture analysis/synthesis. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 229-238).

U-Net (digression)

Nowadays, modern deep learning architectures find the weights and biases of their filters automatically with gradient descent.

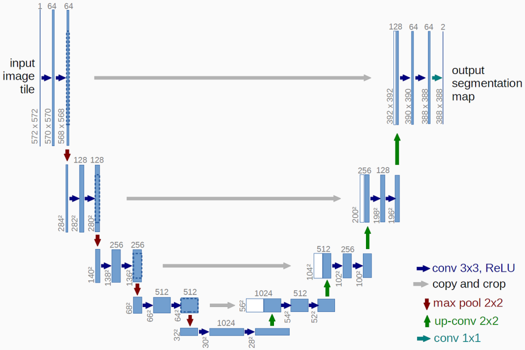


Figure 11: U-Net architecture (illustration from Ronneberger et al. [5])

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

Texture mixing: algorithm

Algorithm 1: Texture mixing

Input: $\{T_i\}_{i=1}^n$ texture exemplars, $\{\lambda_i\}_{i=1}^n$ weights, S number of scales, Q number of orientations, N_{iter} number of iterations

Output: A synthesized texture

Data: Texture $\sim \mathcal{N}(0, I)$ // the input follows a Gaussian noise

for $i = 1, \ldots, n$ **do**

  Pyramids[i] = BuildPyramid(T_i) // for each texture, we build its steerable pyramid

Barycenter = $\widetilde{Bar}(\lambda_i, T_i)_{1 \le i \le n}$ // we compute barycenter of the exemplars, that will be used for color transfer

for $i = 1, \ldots, S \times Q + 2$ **do**

  BarPyramids[i] = $\widetilde{Bar}(\lambda_i, Pyramids[i])_{1 \le i \le n}$ // we compute the barycenter at each scale and orientation

for $i = 1, \ldots, N_{iter}$ **do**

  TexturePyramid = BuildPyramid(Texture) // we build the pyramid for the synthesized texture

  for $j = 1, \ldots, S \times Q + 2$ **do**

    TexturePyramid[j] = $\mathcal{P}_{\text{SW}_2^2}$(TexturePyramid[j], BarPyramids[j]) // for each scale and orientation of the steerable pyramid, we project the synthesized texture on the barycenter of the input textures at its corresponding scale and orientation

  Texture = ReconstructPyramid(TexturePyramid) // we reconstruct the image for the synthesized texture

  Texture = $\mathcal{P}_{\text{SW}_2^2}$(Texture, Barycenter) // we apply color transfer on the synthesized texture

Figure 12: Algorithm for texture mixing

Figure 13: Mixing between 2 textures

Texture mixing: results

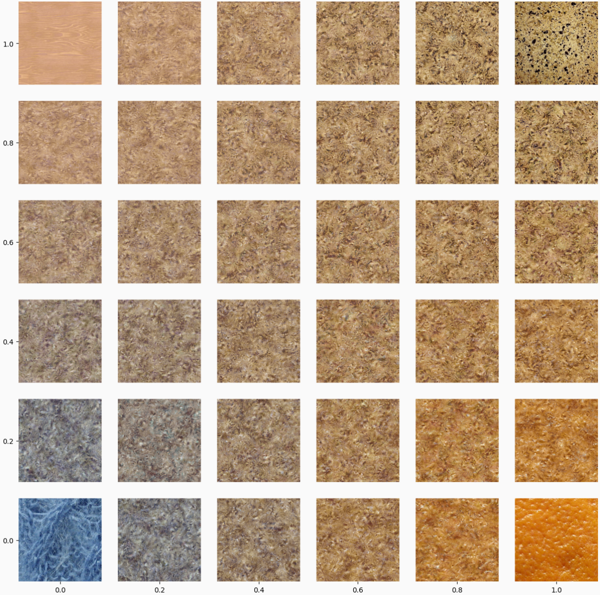
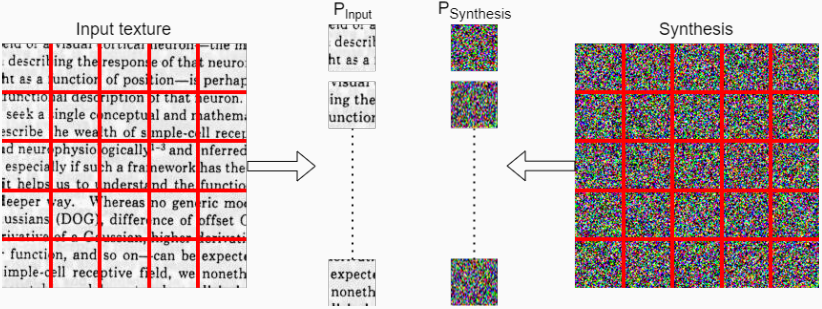


Figure 14: Mixing between 4 textures

Texture mixing: patches

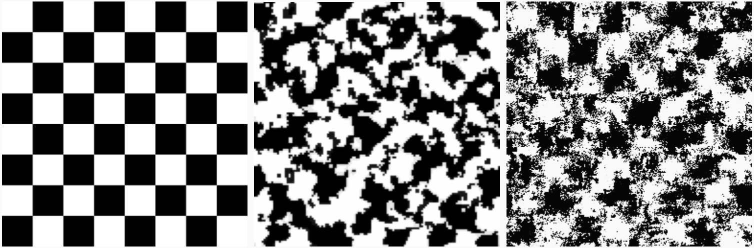
Problem: the algorithm may not capture the details, and patterns of the textures.
Solution: use patches to impose a correlation.



$$P_{Synthesis} = \mathcal{P}_{\mathbf{SW}_2^2}(P_{Synthesis}, P_{Input})$$

Figure 15: Sliced-Wasserstein in the patch domain. Patches are extracted from the images and we project the patches of the synthesis onto the patches of the input texture. This is done at each scale and orientation.

Texture mixing: results



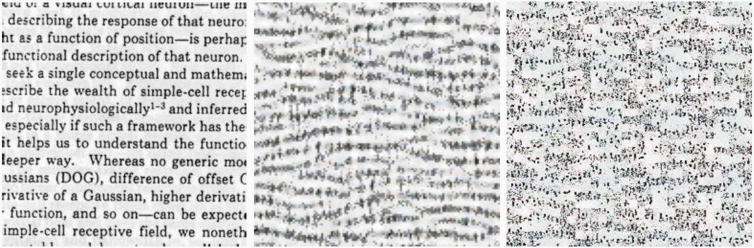
Target image

Texture synthesis without patch constraint

Texture synthesis with patch constraint

Figure 16: Texture synthesis w/o patch constraint

Texture mixing: results


describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathem
escribe the wealth of simple-cell recep
ld neurophysiologically[1-3] and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset C
rivative of a Gaussian, higher derivati
function, and so on—can be expecte
imple-cell receptive field, we noneth

Target image

Texture synthesis without patch constraint

Texture synthesis with patch constraint

Figure 17: Texture synthesis w/o patch constraint

Texture mixing: spectrum

Proposition: Spectrum constraint (Tartavel et al [6])

Let $u_{\text{ref}} \in \mathbb{R}^{N\times\times N\times 3}$ be a reference image and

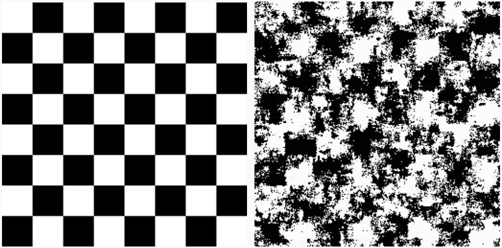
$\mathcal{S} = \{u \in \mathbb{R}^{N\times N\times 3} \mid \exists \phi \in \mathbb{R}^2$ such that $\mathcal{F}_{(k,l)}\{u\} = e^{i\varphi^T \binom{k}{l}} \mathcal{F}_{(k,l)}\{u_{\text{ref}}\}\}$ where $\mathcal{F}$ is the Fourier transform. $\mathcal{S}$ is the set of images whose spectrum is the same as u_{ref}.

We can minimize $\|u_{(k,l)} - u_{(k,l)_{\text{ref}}}\|_2^2$ and keep the same amplitude for u by setting:

$$\mathcal{F}_{(k,l)}\{u_{\text{new}}\} = \frac{\mathcal{F}_{(k,l)}\{u\} \cdot \mathcal{F}_{(k,l)}\{u_{\text{ref}}\}}{|\mathcal{F}_{(k,l)}\{u\} \cdot \mathcal{F}_{(k,l)}\{u_{\text{ref}}\}|} \quad \forall (k,l) \in \{\frac{-N}{2}, \dots, \frac{N}{2} - 1\}^2$$

[6] Tartavel, G., Gousseau, Y., & Peyré, G. (2015). Variational texture synthesis with sparsity and spectrum constraints. Journal of Mathematical Imaging and Vision, 52, 124-144.

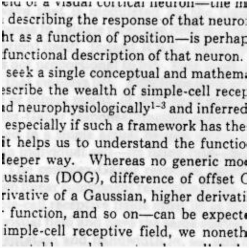
Texture mixing: results



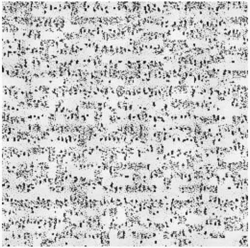
Target image

Texture synthesis with patch and spectrum constraints

Figure 18: Texture synthesis with spectrum constraint


describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathem
scribe the wealth of simple-cell recep
d neurophysiologically[1-3] and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset C
rivative of a Gaussian, higher derivati
function, and so on—can be expecte
imple-cell receptive field, we noneth

Target image



Texture synthesis with patch and spectrum constraints

Figure 19: Texture synthesis with spectrum constraint

Texture mixing: Gromov-Wasserstein optimal plan

Definition: Gromov-Wasserstein distance of order p (Mémoli [7])

Let Σ_1, Σ_2 be two geometric domains and $C_X : \Sigma_1 \times \Sigma_1 \to \mathbb{R}_+, C_Y : \Sigma_2 \times \Sigma_2 \to \mathbb{R}_+$ pairwise distances between two points in spaces X and Y respectively.

$\mathrm{GW}_p((\mu, C_X), (\nu, C_Y)) =$

$$\left(\inf_{\pi \in \Pi(\mu,\nu)} \iint_{\Sigma_1,\Sigma_2} (C_X(x,y) - C_Y(x',y'))^p d\pi(x,y) d\pi(x',y') \right)^{1/p}$$

Definition: Gromov-Wasserstein distance of order p on point clouds

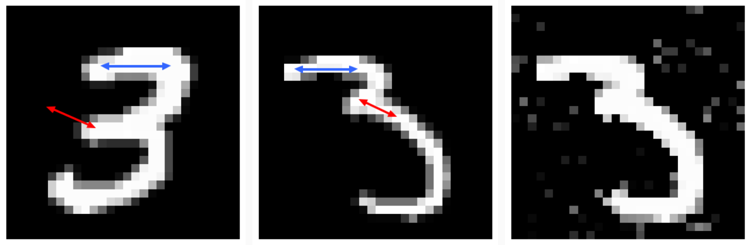
$\mathrm{GW}_p((\mu, C_X), (\nu, C_Y)) = \mathrm{GW}_p(C_X, C_Y) =$

$$\left(\min_{\sigma \in \mathfrak{S}_N} \sum_{(i,j) \in \{1,\ldots,N^2\}} (C_X(x_i, y_j) - C_Y(x_{\sigma(i)}, y_{\sigma(j)}))^p \right)^{1/p}$$

[7] Mémoli, F. (2011). Gromov–Wasserstein distances and the metric approach to object matching. Foundations of mathematics.

Texture mixing: Gromov-Wasserstein optimal plan

Idea: Inspired by the cross-correlation [8], we can minimize the Gromov-Wasserstein distance at the coarsest scale between the orientations. In practice, we designate one orientation as the target and the others as sources and apply the transportation patch-wise.

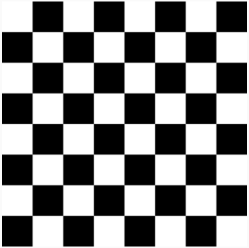


Source image — Target image — Source image after transportation by Gromov-Wasserstein

Figure 20: Gromov-Wasserstein transportation from a source image to a target image. Blue arrows indicate a low distance and red arrows a high one.

[8] Portilla, J., Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. International journal of computer vision, 40, 49-70.

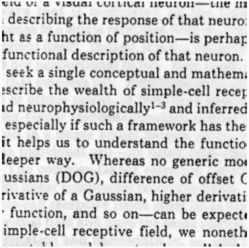
Texture mixing: results



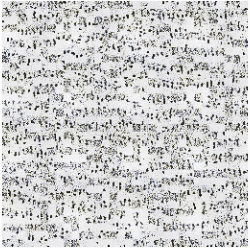
Target image

Texture synthesis with patch and Gromow-Wasserstein

Figure 21: Texture synthesis with Gromow-Wasserstein transportation plan at coarsest scale


describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathem
scribe the wealth of simple-cell recep
d neurophysiologically[1–3] and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset C
rivative of a Gaussian, higher derivati
function, and so on—can be expecte
imple-cell receptive field, we noneth

Target image



Texture synthesis with patch and Gromow-Wasserstein

Figure 22: Texture synthesis with Gromow-Wasserstein transportation plan at coarsest scale

Conclusion

Conclusion & related works

The Wasserstein distance is a powerful tool but can suffer from high dimensionality.

Nonetheless, many studies have used this distance, for example for architectures like Wasserstein GAN and its dual [9], generative AI, texture synthesis [10] [11]

[9] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[10] Delon, J., Desolneux, A., Facq, L., & Leclaire, A. (2023, May). Optimal Transport Between GMM for Multiscale Texture Synthesis. In International Conference on Scale Space and Variational Methods in Computer Vision (pp. 627-638). Cham: Springer International Publishing.

[11] Vacher, J., Davila, A., Kohn, A., & Coen-Cagli, R. (2020). Texture interpolation for probing visual perception. Advances in neural information processing systems, 33, 22146-22157.

M. Agueh and G. Carlier.
Barycenters in the wasserstein space.
SIAM Journal on Mathematical Analysis, 43(2):904–924, 2011.

J. Feydy.
Geometric data analysis, beyond convolutions.
Applied Mathematics, 2020.

F. Mémoli.
Gromov–wasserstein distances and the metric approach to object matching.
Foundations of computational mathematics, 11:417–487, 2011.

J. Rabin, G. Peyré, J. Delon, and M. Bernot.
Wasserstein barycenter and its application to texture mixing.
In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer, 2012.

O. Ronneberger, P. Fischer, and T. Brox.
U-net: Convolutional networks for biomedical image segmentation.
In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

G. Tartavel, Y. Gousseau, and G. Peyré.
Variational texture synthesis with sparsity and spectrum constraints.
Journal of Mathematical Imaging and Vision, 52:124–144, 2015.

References

[4, 1, 2, 5, 6, 3]

Questions?