Texture Interpolation for Probing Visual Perception

Jonathan Vacher[*]
Albert Einstein College of Medicine
Dept. of Systems and Comp. Biology
10461 Bronx, NY, USA
jonathan.vacher@ens.fr

Aida Davila
Albert Einstein College of Medicine
Dominick P. Purpura Dept. of Neuroscience
10461 Bronx, NY, USA
adavila@mail.einstein.yu.edu

Adam Kohn **Ruben Coen-Cagli**
Albert Einstein College of Medicine
Dept. of Systems and Comp. Biology, and
Dominick P. Purpura Dept. of Neuroscience
10461 Bronx, NY, USA
adam.kohn@einsteinmed.org
ruben.coen-cagli@einsteinmed.org

Abstract

Texture synthesis models are important tools for understanding visual processing. In particular, statistical approaches based on neurally relevant features have been instrumental in understanding aspects of visual perception and of neural coding. New deep learning-based approaches further improve the quality of synthetic textures. Yet, it is still unclear why deep texture synthesis performs so well, and applications of this new framework to probe visual perception are scarce. Here, we show that distributions of deep convolutional neural network (CNN) activations of a texture are well described by elliptical distributions and therefore, following optimal transport theory, constraining their mean and covariance is sufficient to generate new texture samples. Then, we propose the natural geodesics (*i.e.* the shortest path between two points) arising with the optimal transport metric to interpolate between arbitrary textures. Compared to other CNN-based approaches, our interpolation method appears to match more closely the geometry of texture perception, and our mathematical framework is better suited to study its statistical nature. We apply our method by measuring the perceptual scale associated to the interpolation parameter in human observers, and the neural sensitivity of different areas of visual cortex in macaque monkeys.

1 Introduction

Texture synthesis Among existing texture synthesis algorithms [41], few have been inspired by visual neuroscience and visual perception. One theory assumed that there exists a fundamental perceptual feature that composes a texture, termed "texton". Despite being falsified, this led to the formulation of the theory of stationary Gaussian textures, which can be obtained by phase randomization [15, 53]. A complementary view, inspired by the primary visual cortex (V1) [25], is that textures perception only depends on the statistics of the wavelet coefficients of the texture (wavelets can be viewed as a standardized collection of "textons" [26]). An implementation of this hypothesis allows for the synthesis of textures by matching the marginal statistics (histograms) of the wavelet coefficients of a white noise image to those of a texture example [22, 6]. Pursuing this idea, Portilla and Simoncelli (PS) obtained high quality new texture samples by iteratively matching a set of higher-order summary statistics of the wavelet coefficients [39, 54]. The PS approach has also been successful in synthesizing sound textures [33]. These algorithms have been further improved with a proper mathematical framework to ensure convergence [49, 50]. A more recent approach uses deep learning to synthesize high-quality textures [16]. This approach is similar to PS [39], but instead of

[*] JV is now based at Laboratoire des Systèmes Perceptifs (LSP), Département d'études cognitives, École Normale Supérieure, PSL University, 75005 Paris, France

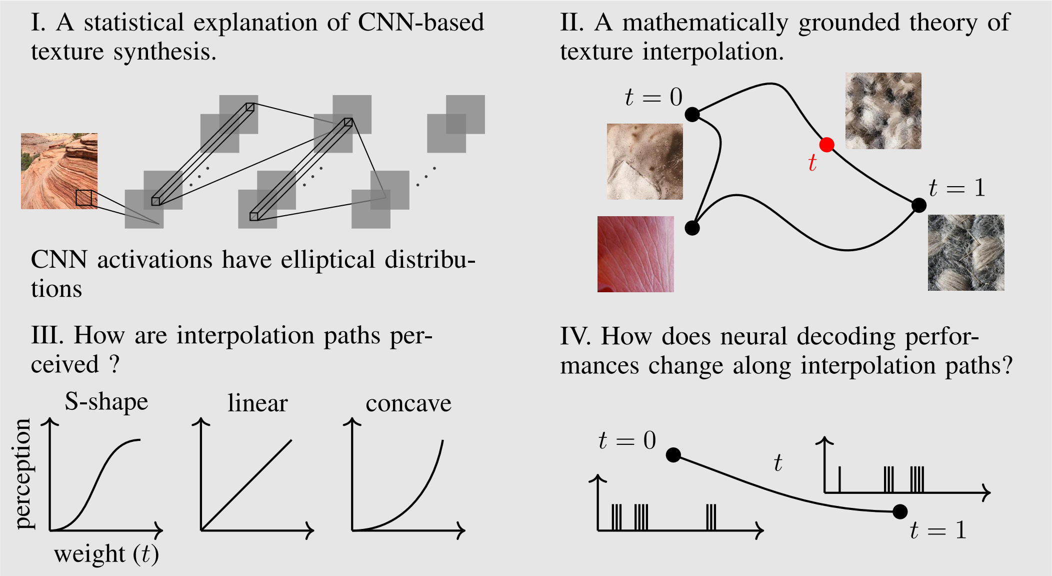
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

wavelet coefficients it consists in matching the statistics of a pre-trained CNN's activations of white noise to those of a texture example. Despite this progress, it remains unclear what computational ingredients are necessary for texture synthesis [52] (e.g., training the CNN weights may not be necessary [21, 52]). Lastly, although our focus here is on texture synthesis approaches that are related to visual perception, there are also multiple approaches that use CNNs, focusing on other aspects such as mathematical methods [31] and computer graphics [51, 62, 64].

Perception and neural encoding of textures Gaussian textures [15] and PS textures [39] have been widely used to study human visual perception (see [56] for a review). Gaussian textures are useful because they can be well parametrized [53] to answer specific questions, *e.g.* related to orientation and spatio-temporal frequency content of images [30, 20]. However, they lack the natural complexity that is captured by PS textures [39]. For this reason, PS textures, often called "naturalistic", have been instrumental to understanding image processing in the visual cortex beyond area V1 [14, 35, 65, 66, 36]. The PS texture model also accounts for various aspects of visual perception including categorization [2], crowding [3] and visual search [44], and has been proposed as a general model of peripheral vision [13, 43] despite some limitations [58, 60, 24]. However, different from simple Gaussian textures, PS textures cannot be easily modeled, and it is difficult to identify perceptually relevant axes in the space of PS summary statistics. Indeed, these statistics consist of a list of heterogeneous features (*i.e.* skewness, kurtosis, magnitude and phase correlations) which are not comparable, and prevent the use of simple closed-form probabilistic descriptions. To circumvent this problem, CNN texture synthesis [16] algorithms are promising because: (i) they can synthesize naturalistic textures that are perceptually indistinguishable from their original version [59]; (ii) they have a simple description based on homogeneous features (deep network activations at each layer) characterized by their mean and covariance, which will allow for more practical modeling.

Texture interpolation and optimal transport Texture interpolation or mixing is a niche in the broader field of texture synthesis. It consists of generating new textures by mixing different texture examples. To our knowledge, PS texture interpolation (and its equivalent for sound textures) has been used in only few neurophysiology and perceptual studies [14, 35, 34], relying on ad-hoc interpolation methods. Instead, texture interpolation is of main interest in computer graphics [4, 45, 5] and to illustrate optimal transport (OT) algorithms [42, 61]. Recent work combines Gaussian models [61] and CNN-based texture synthesis to perform texture interpolation [63]. Other related work further formalizes this, using a statistical description of textures. This is because interpolation between two textures naturally arises as their weighted average, which, could be properly defined in a statistical framework. One approach is to use OT [38] which gives a geometry to the space of probability distributions. Together with the hypotheses that perception of textures is statistical [56] and, that the brain represents probability distributions [40], the OT framework appears as a highly appropriate and unifying framework to study texture perception and its neural correlates. Specifically, it allows for the fine exploration of the space of natural textures by generating and using texture samples along a 1 dimensional geodesic as stimuli.

Contributions Our work provides several contributions that could serve vision studies (Figure 1), which we illustrate using the VGG19 CNN [48]. First, we show that CNN activations of natural textures have elliptical distributions (distribution with elliptical contour lines), which can be described by their mean and covariance even though they are not Gaussian. Leveraging OT theory [38], we show that enforcing these statistics corresponds to matching their distributions. By exploiting the natural geodesics arising with the OT metric, we define the interpolation between arbitrary textures. We compare interpolations obtained with alternative methods, and argue that ours is the most relevant to study the statistical nature of texture perception and its neural basis. Our results also suggest that training the CNN is necessary to generate interpolations that respect perceptual continuity. Lastly, we demonstrate how to use texture interpolation for human psychophysics and monkey neurophysiology experiments. In psychophysics, we use Maximum Likelihood Difference Scaling (MLDS [29, 32]) to measure the perceptual scale of the interpolation weight (*i.e.* the position of the interpolated texture on the geodesic joining two textures). We find that the perceptual scale is reliable for individual participants, and across texture pairs it ranges from linear to threshold non-linear. In neurophysiology, we study the tuning of visual cortical neurons in areas V1 and V4 to interpolation between a naturalistic texture and a spectrally matched Gaussian texture. Using population decoding, we find that adding naturalistic content to the Gaussian texture (*i.e.* moving along the geodesic towards the natural image) does not increase the stimulus-related information in V1, while it increases linearly with the interpolation weight in V4. We provide code to perform texture synthesis and


I. A statistical explanation of CNN-based texture synthesis.

CNN activations have elliptical distributions

II. A mathematically grounded theory of texture interpolation.

III. How are interpolation paths perceived ?

IV. How does neural decoding performances change along interpolation paths?

Figure 1: Outline of our contributions.

interpolation that can be run using a simple command line on a computer with a nvidia GPU or CPUs only[1]. We also provide the code to run the experiments both using psychtoolbox [28] and jspsych [9].

2 Methods

In the following paragraphs, first we define the class of elliptical distributions that we hypothesize are a good description of CNNs activations to natural textures. Then, we propose a method to quantify the elliptical symmetry of a high-dimensional, empirical distribution, to test our hypothesis. Next, we briefly define the OT framework and apply it to elliptical distributions. We then describe our framework for texture synthesis and interpolation.

Elliptical distributions A random vector $X \in \mathbb{R}^D$ ($D \in \mathbb{N}$) is elliptically distributed [19] if its density $\mathbb{P}_X$ can be written as

$$\mathbb{P}_X(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma},g) = c_n|\boldsymbol{\Sigma}|^{-\frac{1}{2}}g((\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is a symmetric positive definite (SPD) matrix, $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean vector, $g : \mathbb{R}_+ \to \mathbb{R}$ is a function such that $\int_0^\infty t^{D/2-1}g(t)dt < \infty$ and

$$c_n = \frac{\Gamma(D/2)}{\pi^{D/2}\int_0^\infty t^{D/2-1}g(t)dt}.$$

where Γ is the Gamma function (*i.e.* an extension of the factorial function). Gaussian Scale Mixtures (GSMs) distributions, which are known to describe the empirical distribution of wavelet coefficients of natural images [57], are a specific case of elliptical distributions [18], including the Gaussian distribution with $g(t) = \exp(-t/2)$, and the Student-t distribution with $g(t) = (1 + t/\kappa)^{-(\kappa+D)/2}$ and $\kappa \in \mathbb{R}_+$. Now, we define the empirical statistics of N data samples $\mathbf{X} \in \mathbb{R}^{D\times N}$. We estimate the mean vector and the SPD matrix by standard empirical estimators, respectively

$$M_N(\mathbf{X}) = \frac{\mathbf{X}\mathbb{1}_N}{N} \quad \text{and} \quad C_N(\mathbf{X}) = G_N(\mathbf{X} - M_N(\mathbf{X})) \quad \text{with} \quad G_N(\mathbf{X}) = \frac{\mathbf{X}\mathbf{X}^{\mathrm{T}}}{N-1}. \qquad (1)$$

where $\mathbb{1}_N = (1,\dots,1)^T \in \mathbb{R}^N$. Thanks to [19] Theorem 3, the function g can be estimated by evaluating the distribution of the norm of the columns of $\bar{\mathbf{X}} = C(\mathbf{X})^{-1/2}(\mathbf{X} - M(\mathbf{X}))$. The density h of these norms verifies

$$h(r) \propto r^{D-1}g(r^2).$$

We will later use this property to measure the empirical convergence of the functional parameter g.

[1] https://github.com/JonathanVacher/texture-interpolation

Wasserstein distance The 2-Wasserstein distance (which we term Wasserstein hereafter) between two probability distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$ can be defined (Remark 2.14 [38]) as

$$W_2(\mathbb{P}_X, \mathbb{P}_Y)^2 = \inf_{X \sim \mathbb{P}_X, Y \sim \mathbb{P}_Y} \mathbb{E}\left(\|X - Y\|^2\right). \tag{2}$$

When the two probability distributions (P_X, P_Y) are elliptical with the same g function and their SPD parameters is equal to their covariance matrices, the Wasserstein distance depends only on their means and covariances

$$W_2(\mathbb{P}_X, \mathbb{P}_Y)^2 = \|\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y\|^2 + \mathcal{B}(\boldsymbol{\Sigma}_X, \boldsymbol{\Sigma}_Y)^2$$

where $\mathcal{B}(\boldsymbol{\Sigma}_X, \boldsymbol{\Sigma}_Y)^2 = \operatorname{Tr}(\boldsymbol{\Sigma}_X + \boldsymbol{\Sigma}_Y - 2(\boldsymbol{\Sigma}_X^{1/2}\boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_X^{1/2})^{1/2})$ is the Bures metric between $\boldsymbol{\Sigma}_X$ and $\boldsymbol{\Sigma}_Y$ [38]. To give some intuition, it is worth noting that when $\boldsymbol{\Sigma}_X$ and $\boldsymbol{\Sigma}_Y$ commute (*i.e.* $\boldsymbol{\Sigma}_X\boldsymbol{\Sigma}_Y = \boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_X$),

$$\mathcal{B}(\boldsymbol{\Sigma}_X, \boldsymbol{\Sigma}_Y) = \left\|\boldsymbol{\Sigma}_X^{1/2} - \boldsymbol{\Sigma}_Y^{1/2}\right\|_f,$$

where $\|\|_f$ is the Frobenius norm between matrices. In 1D, it corresponds to the absolute value of the difference between the standard deviations.

Texture statistics and synthesis algorithm Except for a small subset of textures often called micro-textures [15], natural textures have non-Gaussian statistics. As for natural images, the statistics of their coefficients in a wavelet domain are well modeled by Gaussian Scale Mixtures (GSMs) [57]. Recent work suggests this may also be true when considering their coefficients at the different layers of a CNN [46, 17]. As mentioned above, GSMs are a specific case of elliptical distributions. Therefore, for a given texture, CNN activations at each layer l can be represented by a triplet $(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, g_l)$ which defines the corresponding elliptical distribution. However, this is not sufficient to define a complete generative model of textures, because the triplet characterizes only the marginal distribution and not the joint distribution (*i.e.* spatial dependencies). For this reason, CNN-based texture synthesis methods [16], which consists of imposing target statistics to the feature vectors of a white noise image, exploit the spatial dependencies encoded implicitly in the neural network.

Here we adopt that approach, using the summary statistics of the elliptical distributions. Given the neural network (denoted by $\mathcal{F}$) and the statistics $(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, g_l)_l$ of a texture example u, we achieve synthesis by imposing the statistics to the feature vectors of an input white noise image v. The feature vectors of v at layer l are denoted by $\mathbf{X}_l^v = \mathcal{F}_l(v)$. We considered two different loss functions that when minimized as a function of an input noise v will generate a new texture example. The first loss function, called "Gram" (previously used by Gatys *et al.* [16]), aims at enforcing the Gram matrix of the samples by minimizing the Euclidean norm of the difference with the target Gram matrix

$$L_{\mathrm{G}}(u, v) = \sum_{l=1}^{L} \|G_{N_l}(\mathbf{X}_l^v) - \mathbf{G}_l^u\|_f^2 \tag{3}$$

where $\mathbf{G}_l^u = \boldsymbol{\Sigma}_l^u + \boldsymbol{\mu}_l^u\boldsymbol{\mu}_l^{u\mathrm{T}}$ and G_{N_l} is defined in Equation (1). The second is the "Wasser" loss and aims at minimizing the Wasserstein distance between the input and the target feature vectors

$$L_{\mathrm{W}}(u, v) = \sum_{l=1}^{L} \|M_{N_l}(\mathbf{X}_l^v) - \boldsymbol{\mu}_l^u\|^2 + \mathcal{B}(C_{N_l}(\mathbf{X}_l^v), \boldsymbol{\Sigma}_l^u) \tag{4}$$

where M_{N_l} and C_{N_l} are defined in Equation (1). The Wasserstein distance is only an approximation here because there is no reason why the statistics of the input and the target feature vectors have the same g_l functions at each layer l. Yet, we find in Section 3 that this is empirically enough to match the distribution even without any constraint on g_l. Using these loss functions the synthesis of a texture u is achieved by estimating

$$\bar{v}_{\mathrm{m}} = \operatorname{argmin}_v L_{\mathrm{m}}(u, v)$$

for $\mathrm{m} \in \{\mathrm{G}, \mathrm{W}\}$. Importantly, this problem can be solved by gradient descent (backpropagation) initialized from a white noise image $\bar{v}_0$.

Texture interpolation The Wasserstein distance makes the set of probability measures a metric space (Proposition 2.3 [38]). Therefore, it offers a proper framework to perform texture interpolation because it allows to define the barycenter of K probability measures $(\mathbb{P}_{X_k})_k$ with weights (λ_k) by

$$\bar{\mathbb{P}} = \operatorname*{argmin}_{\mathbb{P}} \sum_{k=1}^{K} \lambda_i W_2(\mathbb{P}, \mathbb{P}_{X_k})^2 \quad \text{where} \quad \sum_{i=1}^{K} \lambda_i = 1.$$

Note that there is an alternative method when using the Wasserstein distance, see supplementary Section 2. Unlike L_{W}, the Gram loss function L_{G} is not derived from a proper metric over the space of probability distributions (*e.g.* Gaussians distributions parametrized by (Σ, μ) and $(\Sigma - \mu\mu^{\mathrm{T}}, 2\mu)$ are different while being associated to the same Gram matrix). However, we can apply a similar heuristic to define interpolation between multiple textures $(u_k)_k$ weighted by (λ_k)

$$\bar{v}_{\mathrm{m}} = \operatorname*{argmin}_{v} \sum_{k=1}^{K} \lambda_k L_{\mathrm{m}}(u_k, v) \quad \text{where} \quad \sum_{i=1}^{K} \lambda_i = 1$$

and for m $\in \{\mathrm{G}, \mathrm{W}\}$. This problem is also solved by gradient descent (backpropagation) initialized from a white noise image $\bar{v}_0$. The use of different loss functions defines different geometries over the space of probability distributions (even if it does not make it a proper metric space). As a consequence, these loss functions will lead to the synthesis of different texture mixtures because the barycenters will lie on different geodesics. In practice, we interpolate between $K = 2$ textures, then $\lambda_1 = t$ and $\lambda_2 = 1 - t$ with $t \in [0, 1]$.

3 Results

First, we present our results on natural texture statistics and the Wasserstein framework. Then, we compare qualitatively our interpolation results to the PS algorithm [39], as well as different CNN architectures and the previously used Gram loss. Finally, we demonstrate with psychophysics and neurophysiology experiments how our texture interpolation algorithm can be used to probe biological vision. We used 32 natural textures from the dataset of [7] and 32 natural images from BSD [1].

Statistics of natural textures Figure 2, top, shows a successful example of texture synthesis by matching the mean and covariance matrix of the CNN activations (eq. (4)). In the Wasserstein framework, if the CNN activations of the example texture are well described by elliptical distributions then matching their mean vectors and covariance matrices is sufficient to match the full distributions. Using the method described in supplementary Section 1, we found that textures are more elliptically distributed than natural images. Specifically, first, the distribution ν_{tex} of CNN activations (Figure 2, third row, orange) was as concentrated as that of wavelet coefficients (bottom row, orange) which are known to be approximately elliptically distributed (*i.e.* as GSMs [57, 8, 47]), although both were less concentrated than Gaussian random vectors (green). Second, the activations of natural textures were more concentrated than those of natural images (Figure 2, blue histograms), and synthesis failed for the example image (Figure 2, top). Similar to wavelets [8, 47], CNN activations of natural images may also be better described by mixture distributions with some components corresponding to textures [55]. We suggest that the higher ellipticity of CNN activations of natural textures compared to those of natural images can explain the success of deep texture synthesis methods like [16].

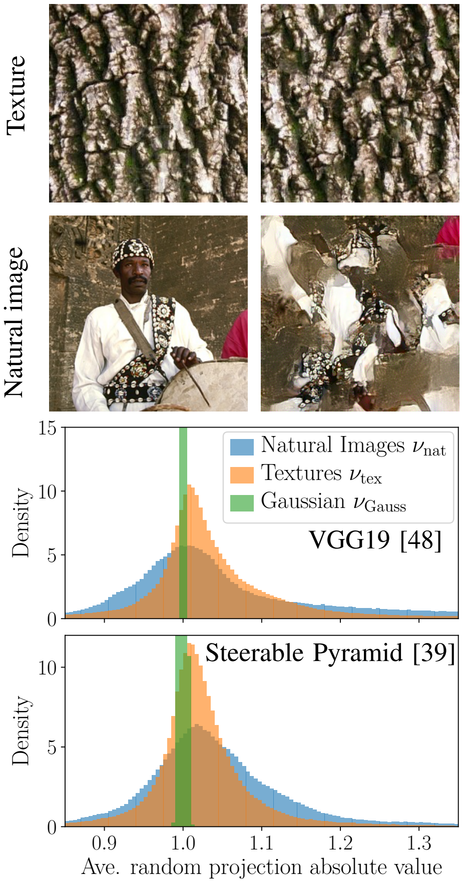


Figure 2: Top two rows: examples of synthesis using our method (eq. (4)). Bottom two rows: quantification of the elliptical symmetry of natural images' and textures' CNN/wavelet activations at a single layer/scale. The narrower the more elliptical. Differences in the histograms for Gaussians are due to the different dimension D.

Empirical convergence of the radial function The Wasserstein loss function (eq. (4)) corresponds to the Wasserstein distance (eq. (2)) only when both distributions are from the same elliptical family *i.e.* when they have the same norm density h. In practice this is not the case because we initialize our optimization with a white noise texture, for which CNN activations are from a different elliptical

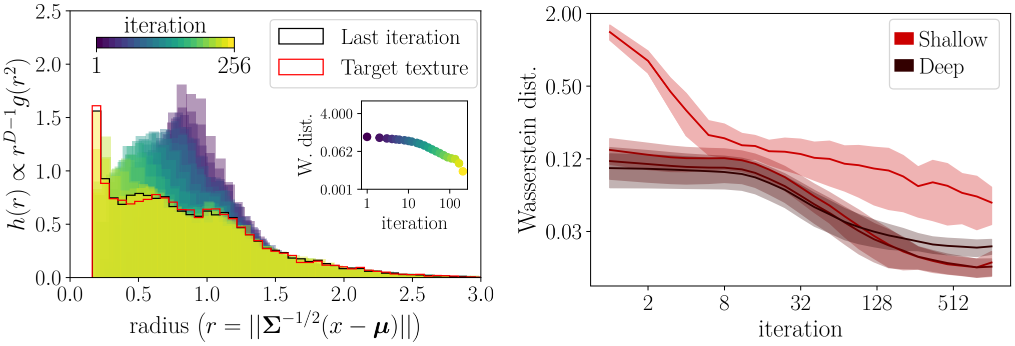


Figure 3: Empirical convergence of the radial distribution h (and therefore g). Left: example for a single texture at a single layer. The histograms represent the distribution of h at each iteration (color coded from blue to yellow). The inset shows the 1D Wasserstein distance between the histogram at current iteration and the histogram corresponding to the target texture. Right: average 1D Wasserstein distance, across 32 textures at different layers. Shaded areas: 95% c.i.

family. Yet, we find empirically that during training the distribution h converges towards that of the target texture. This is illustrated for one example texture in Figure 3, left, and quantified across different textures and CNN layers in Figure 3, right. This result also holds when the network has random weights (not shown), suggesting that the network architecture and the mean and covariance of its activations encode the information corresponding to distribution h.

Comparison of interpolation methods The Wasserstein loss (eq. (4)) offers a natural way to perform texture interpolation using the geodesics defined by the corresponding Wasserstein metric (Section 2, §6). We compare the interpolation obtained with the Wasserstein loss using VGG19 with trained and untrained weights, and a single layer multiscale architecture [52]. We also compare to the interpolation obtained with the Gram loss using VGG19 trained weights [16] and to a variant of the PS algorithm [39] (extension to interpolate color textures [54]). Figure 4 illustrates our main qualitative observation (see also supplementary Figures 7, 8): both the Wasserstein loss interpolation and the PS algorithm generate a perceptually homogeneous mixture of the target textures, whereas the Gram loss interpolation generates textures that are composed of discrete patches of the target textures. Interestingly, both architectures with untrained weights generate more patchy textures, similar to the Gram loss.

Paths of texture interpolation We argue that the interpolation results in Figure 4 for the PS algorithm and the Wasserstein loss with trained weights are more perceptually meaningful, because the interpolated textures preserve stationarity, complying with the hypothesis that texture perception is statistical. As our goal is to study visual perception, we compared the paths of interpolation of the Wasserstein loss and the PS algorithm (which has been used in previous experimental studies [14,

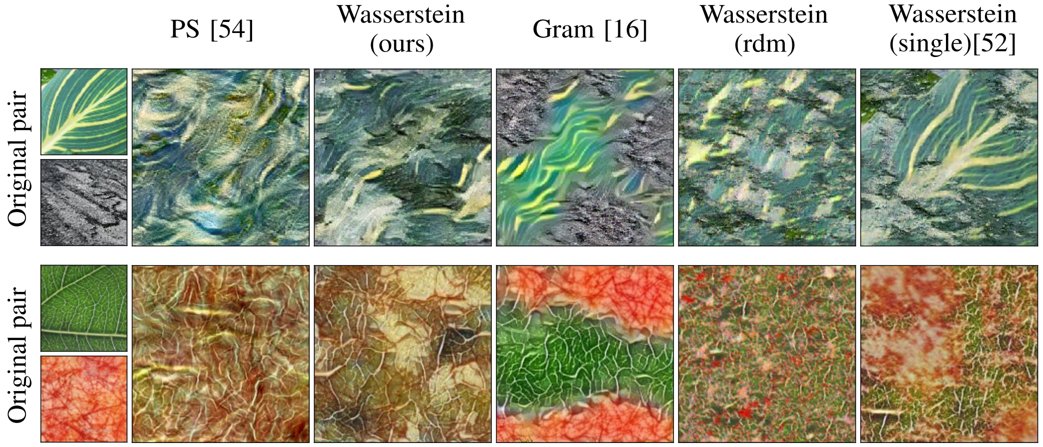


Figure 4: Comparison of textures interpolation methods. PS: the PS algorithm [39] extended to color texture interpolation [54]. Wasserstein (ours), (rdm) and (single): our method eq. (4), using respectively VGG19 pretrained, VGG19 with random weights, and a single layer multiscale architecture [52]. Gram: as in [16] eq. (3). Interpolation weight $t = 0.5$.

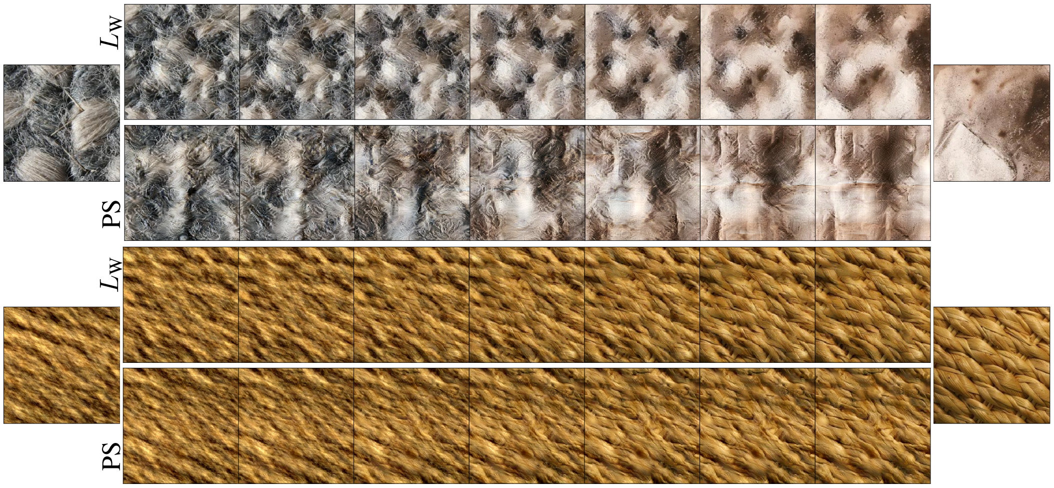


Figure 5: Examples of texture interpolation paths for the Wasserstein loss (L_W) using VGG19 with trained weights, and the PS algorithm extended to interpolation of color textures [54]

35]). Interpolation results are shown in Figure 5 (and also in supplementary Figures 9–12). Overall, these interpolations are perceptually smooth for both algorithms, *i.e.* textures synthesized with close weights appear almost indistinguishable. The interpolations are also quite similar despite the edge artifacts present in PS results (which are due to the periodic boundary implicitly assumed by the Fourier transform used in the complex steerable pyramid [39]). In addition to interpolating between different textures, Figure 5 shows an example of interpolation between a natural texture and a spectrally matched Gaussian texture. This could be useful for vision research studies, because neurons in V1 are thought to be mainly sensitive to spectral cues, whereas higher areas V2/V4 are sensitive to higher-order statistics of these spectral cues [14, 35]. Our qualitative comparison suggests that both our method and PS interpolation may be suitable to probe vision (no patches, smooth interpolation). Yet, our method has the advantage that it relies on simple statistics (mean and covariance) of CNN activations combined with OT, and thus offers a well-grounded mathematical framework for further modeling of perception and neural activity.

Perception of interpolated textures To validate the intuition that our interpolation produces perceptually meaningful results, we measured the perceptual scale associated to the interpolation weight using the MLDS protocol [32, 29].

Protocol The experiment consists of 2-alternate forced choice trials. Participants are presented with 3 stimuli with parameters $t_1 < t_2 < t_3$ and are required to choose which of the two pairs with parameters (t_1, t_2) and (t_2, t_3) is the most similar. We used two sets of three textures (see supplementary Figure 6): (i) a first set where we interpolate between the stationary Gaussian synthesis ($t = 0$) and the naturalistic texture ($t = 1$, as in Figure 5 bottom); (ii) a second set where we interpolate between two arbitrary textures (as in Figure 5 top). All stimuli had an average luminance of 128 (range $[0, 255]$) and an RMS contrast of 39.7. For each texture pair, we use 11 equally spaced ($\delta_t = 0.1$) interpolation weights. To ensure that stimulus comparisons are above the discrimination threshold we only use triplets such that $|t_i - t_j| \geqslant 2\delta_t$. For each texture set (i) and (ii), a groups of 8 naive participants performed the experiment. Participants were recruited through the platform prolific[2] and performed the experiments online. The protocol was approved by the Internal Review Board of the Albert Einstein College of Medicine. Monitor gamma was corrected to 1 assuming the standard value of 2.2.

MLDS model The MLDS model assumes that the observer uses a perceptual scale that is an increasing function $t \mapsto f(t) \in [0, 1]$ to perform the task. At each trial, the observer makes a stochastic judgement whether or not $f(t_1) - 2f(t_2) + f(t_3) + \varepsilon < 0$, where ε is the observer noise modeled as a zero-mean Gaussian variable with standard deviation $\sigma > 0$.

Results For each texture, we fitted the MLDS model on participants' data pooled together using the standard method [29] (Figure 6). We also fitted the model to individual participant data (supplementary Figure 4 and 5). First, we found that the measured perceptual scale is meaningful, because confidence intervals for human participants (Figure 6, colored shaded areas) are tight compared to

[2] https://www.prolific.co

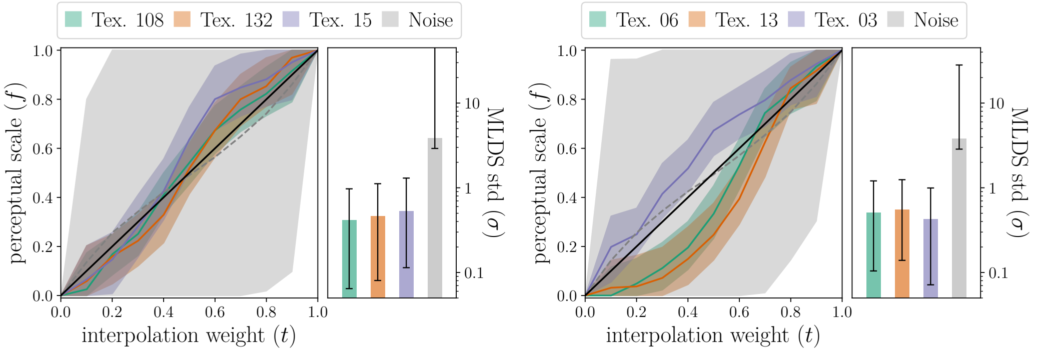


Figure 6: Perceptual scale of the interpolation weights and the corresponding observer model standard deviation. Error bars are 99.5% bootstrapped c.i.. Left: Stationary Gaussian to naturalistic interpolation. Right: Interpolation between arbitrary texture pairs.

an observer that would answer randomly (gray shade) and despite the additional inter-participant variability. This is also confirmed by the estimated standard deviations of the MLDS model (Figure 6, bar plots) that are one order of magnitude lower for the participants than for the random observer. For the first set of textures (i), perceptual scales have roughly an S-shape. Individual fits (supplementary Figure 4)) reveal that there are in fact 3 underlying behaviors: (a) linear for participants 1, 2 and 4; (b) S-shape for participants 3,7 and 8 and (c) top-asymmetric concave for participants 5 an 6. In the second set of textures (ii), perceptual scales have two different behaviors: (d) symmetric concave for texture pair 03 and (e) bottom-asymmetric convexe for texture pairs 06 and 13. Individual fits (supplementary Figure 5)) could be characterized by these behaviors. Taken together these results show an insight of the diversity of behaviors. More participants and textures will be required for a complete characterization and to further understand the implications for perception.

Neural coding of interpolated textures To illustrate how our interpolation method can be used to probe neural coding, we analyzed the spiking activity of simultaneously recorded V1 (N=6) and V4 (N=5) neurons in macaque monkeys, in response to 3 distinct sets of interpolated textures.

Protocol Textures were interpolated between synthesized naturalistic textures ($t = 1$) and their spectrally matched Gaussian counterpart ($t = 0$) at 5 different weights ($t = 0.0, 0.3, 0.5, 0.7, 1.0$). All stimuli had their luminance normalized as in the MLDS experiment and were presented at 5 different sizes (2°, 4°, 6°, 8°, 10°) on a CRT monitor. Recordings were conducted in an awake, fixating adult male macaque monkey (*Macaca fascicularis*) implanted with "Utah" arrays in V1 and V4 [27]. A successful trial consisted of the subject maintaining fixation over a central 1.4° window for 1.3 seconds. During this time we presented a sequence of 3 textures displayed for 300-ms each and immediately followed by a 100-ms blank screen.

Data analysis We decoded the interpolation weight from both spike trains and spike counts by Linear Discriminant Analysis using scikit-learn [37]. Specifically, we considered three pairs comprising one texture and the corresponding spectrally matched Gaussian texture, and we decoded the following pairwise weight conditions $(t_1, t_2) = (0.0, 0.3), (0.0, 0.5), (0.0, 0.7), (0.0, 1.0)$ for each texture pair. For spike trains, we first reduced the dimensionality using Principal Component Analysis, and chose the number of components that maximized the averaged cross-validated (5 folds) classification performance.

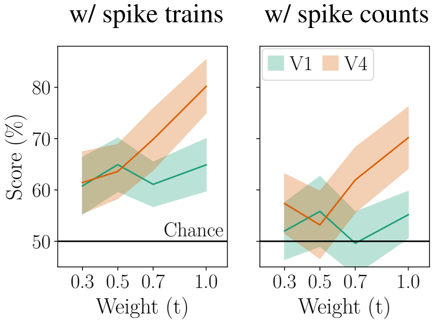


Figure 7: Decoding of weights 0 *and* $t = 0.3, 0.5, 0.7$ *and* 1.0*, averaged over 3 textures, from V1 and V4 neural activity. Error bars are 95% c.i.. Left: using full spike trains+smoothing+PCA. Right: using raw spike counts.*

Results We asked if neurons are sensitive to the interpolation parameter, and if sensitivity is different across areas. The results in Figure 7 suggest that adding naturalistic content to a stationary Gaussian texture, while preserving its power spectrum, does not increase the stimulus-related information in V1, whereas it increases approximately linearly with the weight in V4. This is true both when using the spike count during the stimulus presentation (Figure 7 left) or the full spike train (Figure 7 right).

This decoding performance corroborate the fact that neurons are tuned to the interpolation weight in V4 and not in V1 (see supplementary Figure 3). The decoding performance is overall higher, and the trend more linear for V4, when using the full spike train which indicates that relevant information is encoded dynamically. These results thus offer a basis to further link the measured perceptual scale to neural activity in the visual cortex.

4 Discussion

We studied texture statistics across layers of CNNs, and found that they are more elliptical than natural images. Based on that finding, we showed that texture synthesis can be cast as a minimization of the Wasserstein distance to the distribution of CNN activations of a target texture. The proposed framework offers a geometric interpretation of the space of textures, and affords a precise definition of distance and interpolation between textures based on optimal transport. In particular, it allows one to define a neighborhood (ε-environment) that is compatible with simple summary statistics (see [52]). Our empirical analysis of synthesis and interpolation suggests that CNN weights and architecture both have an effect on the interpolation paths of textures. When the CNN is trained the paths are perceptually smoother and consist of stationary homogeneous textures. This might be because a trained network provides a smoother approximation of the manifold of textures. To our knowledge, there is no comparison of the Gram *vs* Wasserstein loss for texture synthesis. The reason is that, differences are not visible at first sight on the synthesized textures without sampling interpolation of textures. Such differences may be crucial for visual perception studies but less for computer graphics. We also found that the linear interpolation in the inhomogeneous space of PS summary statistics generates homogeneous textures. However, the combination of CNN and optimal transport has the practical advantage of a homogeneous feature space and simpler distributions, thus offering a well-grounded mathematical framework for characterizing and modeling biological visual processing. Yet, our work is limited to textures while a full understanding of natural image space is crucial to further understand visual perception [23, 10]. Also, we didn't explore the possibility of improving the network architecture to produce similar quality textures using less parameters like feature variances and means or feature means only [12, 11]. We demonstrated the applicability with perceptual and neurophysiology experiments, and found preliminary evidence that our interpolation based on probability distributions influence both perception and neural activity. Previous work [14, 58] focused on the perceptual sensitivity of "naturalness" (when the weight t goes from 0 to 1) and showed that it is partly predicted by neuronal responses in V2. In comparison, the MLDS protocol will allow for the measurement of a full perceptual scale, not just perceptual sensitivity. We expect to further relate the perceptual scale to recordings in the visual cortex. In addition, we do not limit our study to the perception of "naturalness" and we propose to measure the perception of interpolation path between arbitrary texture pairs. Such interpolation paths should provide more fine-grained information about the perception of the texture space geometry than previous massive data collection [35].

Broader Impact

Any protocol that quantifies perception could potentially be turned into a diagnostic tool. Apart from that, our work does not present any foreseeable societal consequence.

Acknowledgements

We thank Anna Jasper for help with neuronal recordings and Pascal Mamassian for suggesting the use of MLDS and for fruitful discussions. JV and RCC are supported by NIH (NIH/CRCNS EY031166).

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[2] B. Balas. Attentive texture similarity as a categorization task: comparing texture synthesis models. *Pattern recognition*, 41(3):972–982, 2008. `https://dx.doi.org/10.1016/j.patcog.2007.08.007`.

[3] B. Balas, L. Nakano, and R. Rosenholtz. A summary-statistic representation in peripheral vision explains visual crowding. *Journal of vision*, 9(12):13–13, 2009. `https://dx.doi.org/10.1167/9.12.13`.

[4] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on visualization and computer graphics*, 7(2):120–135, 2001.

[5] U. Bergmann, N. Jetchev, and R. Vollgraf. Learning texture manifolds with the periodic spatial GAN. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 469–477. PMLR, 2017.

[6] T. Briand, J. Vacher, B. Galerne, and J. Rabin. The heeger-bergen pyramid-based texture synthesis algorithm. *Image Processing On Line*, 4:2014–11, 2014.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[8] R. Coen-Cagli, P. Dayan, and O. Schwartz. Statistical models of linear and nonlinear contextual interactions in early visual processing. In *Advances in neural information processing systems*, pages 369–377, 2009.

[9] J. R. De Leeuw. Jspsych: a javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, 47(1):1–12, 2015.

[10] A. Deza, A. Jonnalagadda, and M. P. Eckstein. Towards metamerism via foveated style transfer. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[11] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli. Image quality assessment: unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2020.

[12] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style, 2017.

[13] J. Freeman and E. P. Simoncelli. Metamers of the ventral stream. *Nature neuroscience*, 14(9):1195, 2011. `https://doi.org/10.1038/nn.2889`.

[14] J. Freeman, C. M. Ziemba, D. J. Heeger, E. P. Simoncelli, and A. J. Movshon. A functional and perceptual signature of the second visual area in primates. *Nature neuroscience*, 16(7):974, 2013. `https://doi.org/10.1038/nn.3402`.

[15] B. Galerne, Y. Gousseau, and J.-M. Morel. Random phase textures: theory and synthesis. *IEEE Transactions on image processing*, 20(1):257–267, 2011. `https://doi.org/10.1109/TIP.2010.2052822`.

[16] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[17] L. G. S. Giraldo and O. Schwartz. Integrating flexible normalization into midlevel representations of deep convolutional neural networks. *Neural computation*, 31(11):2138–2176, 2019.

[18] E. Gómez-Sánchez-Manzano, M. Gómez-Villegas, and J. Marín. Sequences of elliptical distributions and mixtures of normal distributions. *Journal of multivariate analysis*, 97(2):295–310, 2006.

[19] E. Gómez, M. A. Gómez-Villegas, and J. M. Marín. A survey on continuous elliptical vector distributions. *Revista matemática complutense*, 16(1):345–361, 2003.

[20] R. L. Goris, E. P. Simoncelli, and J. A. Movshon. Origin and function of tuning diversity in macaque visual cortex. *Neuron*, 88(4):819–831, 2015.

[21] K. He, Y. Wang, and J. Hopcroft. A powerful generative model using random weights for the deep image representation. In *Advances in Neural Information Processing Systems*, pages 631–639, 2016.

[22] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM, 1995. `http://dx.doi.org/10.1145/218380.218446`.

[23] O. J. Hénaff and E. P. Simoncelli. Geodesics of learned representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[24] D. Herrera-Esposito, R. Coen-Cagli, and L. Gomez-Sena. Flexible contextual modulation of naturalistic texture perception in peripheral vision. *bioRxiv*, 2020.

[25] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[26] B. Julész. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91, 1981. `https://doi.org/10.1038/290091a0`.

[27] R. C. Kelly, M. A. Smith, J. M. Samonds, A. Kohn, A. Bonds, J. A. Movshon, and T. S. Lee. Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex. *Journal of Neuroscience*, 27(2):261–264, 2007.

[28] M. Kleiner, D. Brainard, D. Pelli, A. Ingling, R. Murray, and C. Broussard. What's new in psychtoolbox-3. *Perception*, 36(14):1–16, 2007.

[29] K. Knoblauch, L. T. Maloney, et al. Mlds: maximum likelihood difference scaling in r. *Journal of Statistical Software*, 25(2):1–26, 2008.

[30] M. S. Landy and J. R. Bergen. Texture segregation and orientation gradient. *Vision research*, 31(4):679–691, 1991.

[31] A. Leclaire and J. Rabin. A multi-layer approach to semi-discrete optimal transport with applications to texture synthesis and style transfer. *preprint*, 2019.

[32] L. T. Maloney and J. N. Yang. Maximum likelihood difference scaling. *Journal of Vision*, 3(8):5–5, 2003.

[33] J. H. McDermott, A. J. Oxenham, and E. P. Simoncelli. Sound texture synthesis via filter statistics. In *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 297–300. IEEE, 2009.

[34] R. McWalter and J. H. McDermott. Adaptive and selective time averaging of auditory scenes. *Current Biology*, 28(9):1405–1418, 2018.

[35] G. Okazawa, S. Tajima, and H. Komatsu. Image statistics underlying natural texture selectivity of neurons in macaque v4. *Proceedings of the National Academy of Sciences*, 112(4):E351–E360, 2015. eprint: `https://www.pnas.org/content/112/4/E351.full.pdf`.

[36] G. Okazawa, S. Tajima, and H. Komatsu. Gradual development of visual texture-selective properties between macaque areas v2 and v4. *Cerebral Cortex*, 27(10):4867–4880, 2017.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[38] G. Peyré, M. Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[39] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70, 2000. `http://dx.doi.org/10.1023/A:1026553619983`.

[40] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham. Probabilistic brains: knowns and unknowns. *Nature neuroscience*, 16(9):1170, 2013.

[41] L. Raad, A. Davy, A. Desolneux, and J.-M. Morel. A survey of exemplar-based texture synthesis. *Annals of Mathematical Sciences and Applications*, 3(1):89–148, 2018. `http://dx.doi.org/10.4310/AMSA.2018.v3.n1.a4`.

[42] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011.

[43] R. Rosenholtz. Capabilities and limitations of peripheral vision. *Annual Review of Vision Science*, 2:437–457, 2016. `https://doi.org/10.1146/annurev-vision-082114-035733`.

[44] R. Rosenholtz, J. Huang, A. Raj, B. Balas, and L. Ilie. A summary statistic representation in peripheral vision explains visual search. *Journal of vision*, 12(4):14–14, 2012. `https://dx.doi.org/10.1167/12.4.14`.

[45] R. Ruiters, R. Schnabel, and R. Klein. Patch-based texture interpolation. *Computer Graphics Forum (Proc. of EGSR)*, 29(4):1421–1429, June 2010. J. Lawrence and M. Stamminger, editors.

[46] L. G. Sanchez-Giraldo, M. N. U. Laskar, and O. Schwartz. Normalization and pooling in hierarchical models of natural images. *Current opinion in neurobiology*, 55:65–72, 2019.

[47] O. Schwartz, T. J. Sejnowski, and P. Dayan. Soft mixer assignment in a hierarchical generative model of natural scene statistics. *Neural computation*, 18(11):2680–2718, 2006.

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] G. Tartavel, Y. Gousseau, and G. Peyré. Variational texture synthesis with sparsity and spectrum constraints. *Journal of Mathematical Imaging and Vision*, 52(1):124–144, 2015. `https://doi.org/10.1007/s10851-014-0547-7`.

[50] G. Tartavel, G. Peyré, and Y. Gousseau. Wasserstein loss for image synthesis and restoration. *SIAM Journal on Imaging Sciences*, 9(4):1726–1755, 2016. `https://doi.org/10.1137/16M1067494`.

[51] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016. `http://proceedings.mlr.press/v48/ulyanov16.html`.

[52] I. Ustyuzhaninov, W. Brendel, L. A. Gatys, and M. Bethge. What does it take to generate natural textures? In *ICLR*, 2017.

[53] J. Vacher, A. I. Meso, L. U. Perrinet, and G. Peyré. Bayesian modeling of motion perception using dynamical stochastic textures. *Neural computation*, 30(12):3355–3392, 2018. `https://doi.org/10.1162/neco_a_01142`.

[54] J. Vacher and T. Briand. The portilla-simoncelli texture model: towards the understanding of the early visual cortex. *Image Processing On Line*, preprint, 2020.

[55] J. Vacher and R. Coen-Cagli. Combining mixture models with linear mixing updates: multilayer image segmentation and synthesis. *arXiv preprint arXiv:1905.10629*, 2019.

[56] J. D. Victor, M. M. Conte, and C. F. Chubb. Textures as probes of visual processing. *Annual review of vision science*, 3:275–296, 2017.

[57] M. J. Wainwright and E. P. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *Advances in neural information processing systems*, pages 855–861, 2000.

[58] T. S. A. Wallis, M. Bethge, and F. A. Wichmann. Testing models of peripheral encoding using metamerism in an oddity paradigm. *Journal of Vision*, 16(2):4–4, March 2016. eprint: `https://jov.arvojournals.org/arvo/content\_public/journal/jov/934904/i1534-7362-16-2-4.pdf`. `https://dx.doi.org/10.1167/16.2.4`.

[59] T. S. A. Wallis, C. M. Funke, A. S. Ecker, L. A. Gatys, F. A. Wichmann, and M. Bethge. A parametric texture model based on deep convolutional features closely matches texture appearance for humans. *Journal of vision*, 17(12):5–5, 2017. `https://dx.doi.org/10.1167/17.12.5`.

[60] T. S. A. Wallis, C. M. Funke, A. S. Ecker, L. A. Gatys, F. A. Wichmann, and M. Bethge. Image content is more important than bouma's law for scene metamers. *eLife*, 8:e42512, 2019. `https://doi.org/10.7554/eLife.42512`.

[61] G.-S. Xia, S. Ferradans, G. Peyré, and J.-F. Aujol. Synthesizing and mixing stationary gaussian texture models. *SIAM Journal on Imaging Sciences*, 7(1):476–508, 2014.

[62] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays. Texturegan: controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2018.

[63] Z. Xue and Z. Wang. Texture mixing by interpolating deep statistics via gaussian models. *IEEE Access*, 8:60747–60758, 2020.

[64] N. Yu, C. Barnes, E. Shechtman, S. Amirghodsi, and M. Lukac. Texture mixer: a network for controllable synthesis and interpolation of texture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12164–12173, 2019.

[65] C. M. Ziemba, J. Freeman, A. J. Movshon, and E. P. Simoncelli. Selectivity and tolerance for visual texture in macaque v2. *Proceedings of the National Academy of Sciences*, 113(22):E3140–E3149, 2016. `https://doi.org/10.1073/pnas.1510847113`.

[66] C. M. Ziemba, J. Freeman, E. P. Simoncelli, and A. J. Movshon. Contextual modulation of sensitivity to naturalistic image structure in macaque v2. *Journal of neurophysiology*, 120(2):409–420, 2018. `https://doi.org/10.1152/jn.00900.2017`.