Neural Texture Synthesis: A Survey

Zeyang Huang

***Abstract*—In this survey, we make comparisons and give explanations on different state-of-the-art texture synthesis algorithms for both 2-dimensional images and 3-dimensional objects, and from statistical approaches to recent neural approaches, with an emphasis on methods using convolutional neural network and generative adversarial networks. This paper also discusses related works: the applications of texture synthesis on other fields as well as its relationship with similar fields are presented.**

***Index Terms*—Texture synthesis, geometric texture modeling, geometry processing, convolutional neural network, generative adversarial networks**

I. Introduction

LINGUISTS use “things” and “stuff” to differentiate countable nouns like a table and nouns of unspecified contents such as snow [1], while integrating “things” and “stuff”, or correspondingly, unifying instances and semantics, is likewise a major theme that the computer vision community would like to pursue. Material perception is categorized as discovering the “stuff” with preattentive features during the early stages of visual processing, as we may characterize surface variations of tree bark as “texture”, but the surface difference for eyes and mouth as “parts” on the face [2], [3]. Even though in an indiscrete fashion, since textures may acquire various identity-preserving transformations, we are able to infer whether an object is rough or smooth, wood or furry, soaked with water or dried out with the help from the natural processing of visual, tactile, and perhaps auditory sensors.

Textures are often considered as a subcategory of material surface properties along with glossiness and material recognition [2]. While the cross-modal interaction has been well studied in [4], [5], etc, the texture properties alone that describe regular pattern generated by bumps and dips of an object’s surface structure, and discriminated via the posterior collateral sulcus of our brain, has attracted much attention beyond neuroscience communities [6]. For materials like granite containing multiple reflective properties, the material-specific texture is characterized with repetitive illumination and color change, which can be modeled by a set of displacement rules and primitives deterministically (Figure 1). While for mass amounts of small grains such as sand, the reflectance model of each individual is simple, yet the spatial location together formalizes the overall stochastic texture property [7].

Regularity and randomness nature also has been reflected during cortical processing, as texture images could be decomposed in a pyramid representation into local orientations and spatial frequency signals, with high pass and low pass filters [8]. The regularity nature provides a basis for the synthesis process, while randomness with ambiguities makes the task non-trivial with a balance between local neighborhoods and global appearances.

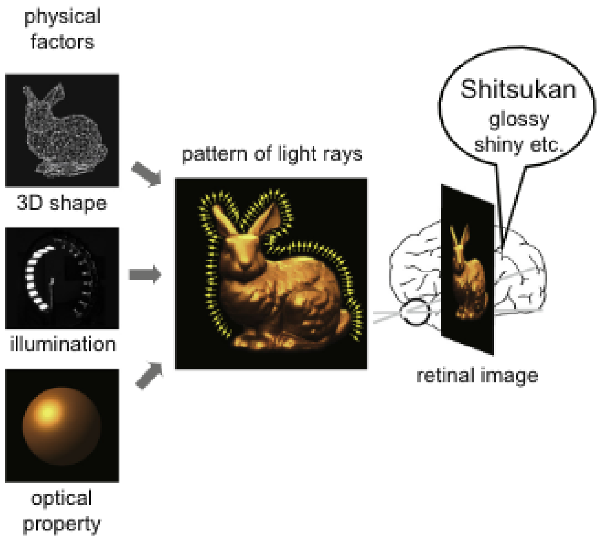


Fig. 1. Physical factors that being processed in forming a retinal image that gives Shitsukan (a Japanese word meaning ”sense of quality”) [2].

The regularity often infers low-dimensional structures, as a matrix-representation of an image can be defined as a discrete sampling of a continuous texture function on 2D domain [9]. Let function $I_o(x,y)$ describes a 2D texture in $\mathbb{R}^2$,

$$\begin{aligned} r \doteq dim(span\{I_o(x,y)|y \in \mathbb{R}\}) \leq k \\ for\ some\ k \in \mathbb{Z}^+ \end{aligned} \quad (1)$$

If r is finite, then I_o has a rank-r texture. And I_o is defined as a low-rank texture if the family of 1D functions $\{I_o(x,y)|y \in \mathbb{R}\}$ span a finite low-dimensional linear subspace [9].

The simplest checkerboard-like corners and contours has rank-1 texture, and the low-rank textures, in general, can give insights into global characteristics to recover a structured object when it has partial occlusion or corruption.

Two-dimensional sample-based texture synthesis, texture inpainting, or texture recovery, all more or less describes the task that a larger similar-looking image needs to be generated based on small patches of texture [10], spatial continuity [11], wavelet construction [12]. Other methods rely on local statistics based on the information provided by order of each pixel for nearest neighborhoods with pyramid coarse-to-fine searches [13], loopy belief propagation within blocks [14], etc [15].

Photometric appearances can be integrated with the 3d geometry of the shape, correspondingly, texture recovery along with shape recovery formalize the task of 3d reconstruction [15]. In the meantime, the computer graphics community has been widely and thoroughly studied texturing 3D models. Before neural networks’ prevalence, there are three main categories of 2d texture synthesis approaches, non-parametric resampling pixels or patches, statistical parametric texture descriptions, and probabilistic models to describe the texture. And when it comes to 3d, we also need to facilitate the texture mapping process that transfers 2d images onto 3d objects. Given the well-developed context of this field, in this survey, we are focusing on illustrating the relationship between texture synthesis and its similar works, and discussing the most recent approaches using neural networks.

II. Related Work

A. BRDF and BTF Estimation

Bidirectional reflectance distribution function (BRDF) is a general model of light scattering associated with local coordinate surface frames consisting of four parameters describing two incident angles and two exiting angles [15]. Equation 2 calculate the amount of light exiting a surface at point p in $\hat{v}_r$ direction, with incoming light $L_i(\hat{v}_i;\lambda)$, spatially-varying BRDF f_r, incident direction $\hat{v}_i$, reflected direction $\hat{v}_r$, wavelength λ, normal vector $\hat{n}$, and the angle λ_i between $\hat{v}_i$ and $\hat{n}$ [15],

$$\begin{gathered} L_r(\hat{v}_r;\lambda) = \int L_i(\hat{v}_i;\lambda) f_r(\hat{v}_i,\hat{v}_r,\hat{n};\lambda) cos^+\theta_i d\hat{v}_i \\ where\ cos^+\theta_i = max(0, cos\theta_i) \end{gathered} \tag{2}$$

For most real surfaces, it seems that only when the viewing and reflecting directions are very similar, the reflexive term's value is significant [9].

From a perceptual point of view, we are still unclear on how many parameters our visual system putatively use, and there are cases that two arbitrarily similar BRDF would be perceived differently with specific-designed lighting, shape, and viewpoint combinations [16]. Even though the gloss properties sometimes are hard to be perceptually discriminable, but the linear combination of the parameters have been successfully approximating a broad range of real materials [17], [18], [15].

The small-scale geometry of most real material surfaces would cause mesoscopic effects. Dana *et al.* proposed Bidirectional Texture function (BTF) as an analogy to BRDF with the similar parameters of illumination and viewing direction to capture the appearance of textures in fine-scale spatial variations [19]. Often a tenser is used to represent BTF,

$$\begin{gathered} B \in R^{c \times n_l \times n_v \times w \times h} \\ f(B) \approx U \cdot V \end{gathered} \tag{3}$$

where c is the number of color channels, n_l is the number of sampled light, n_v is the number of view directions, w, h represent the size of bidirectional textures, and matrix representation of BTFs admit a low-rank factorization with

$$\begin{gathered} U \in R^{n_l \cdot n_v \times k}, V \in R^{k \times c \cdot w \cdot h} \\ U, V\ are\ matrices\ of\ rank\ k << n_l \cdot n_v \end{gathered} \tag{4}$$

The high dimensional BTF gives a better representation but the data-driven nature also requires demanding measurement setups and massive amounts of data like thousands of material samples under varying viewing angles and illumination conditions, leading to many research studies focusing on data collection, compression, and modeling [20]. Brok *et al* tackles this issue by inferring the high-resolution BTF from low-resolution measures with the help of single image super-resolution CNN architecture [21]. Zhang *et al.* combined the existing Conditional Generative Adversarial Network with the perceptual feature regressions to learn the BTF data and synthesize new BTF images with brand new illumination and viewing angles [22], [23].

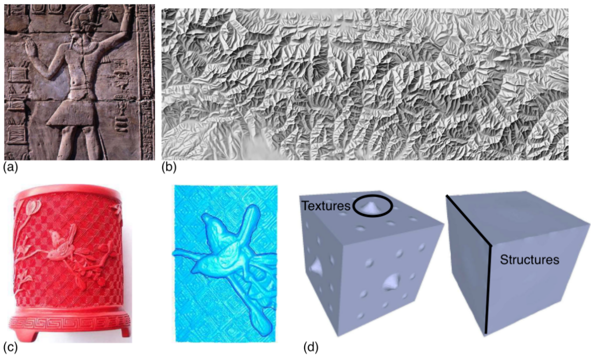


Fig. 2. (a) A cultural shallow bas-relief [28]. (b) Shaded relief of Caucasus Mountains created by neural network [29]. (c) Relief on a textured background for daily objects [30]. (d) Structure-preserving relief texture removal [31].

Rainer *et al.* proposed a neural net representation with autoencoders: the encoder compresses each texel to latent coefficients while the decoder combines lighting and viewing vectors to output an RGB vector indicating the reflectance values [24]. Maximov *et al*, further, proposed Deep Appearance Maps (DAMs) that models appear as a network directly so that the network is a 4D generalization over 2D reflectance maps. Specifically, an input image with n materials would output n different DAMs together with a segmentation network mapping every pixel to n weights, so that instead of extracting explicit reflectance and illumination parameters, the deep appearance representation would offer a generalization of reflectance maps in world space [25].

B. Geometric Relief Modeling

As an intermediate art category between paintings and sculptures, reliefs are sometimes being considered to be 2.5 dimensional and other times as a full 3d sculpture [26]. There are three common types of relief sculptures: bas-relief where objects are slightly projected from background surface; high-relief where objects are at least half or more from the background; and sunken relief where objects are carved below the surface [27]. Reliefs often can be found beyond the cultural heritage context with applications in crafts, cartographic reliefs showing terrain information, and in a broader sense as a geometric texture with an emphasis on bas-relief modeling (Figure 2).

Cignoni *et al.* proposed 2.5 dimensional height fields to represent the relief geometry [32]. Such representation is widely used with the following challenges of compression algorithms, height discontinuities at silhouettes, and depth ordering preservation. Weyrich *et al.* proposed a gradient-domain approach [33]. Rather than directly compressing into

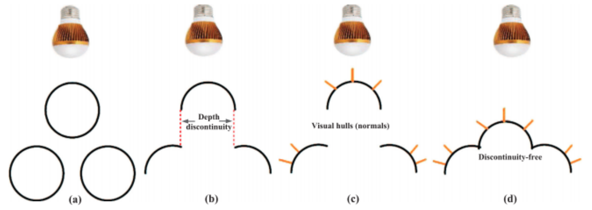


Fig. 3. The three circles represent surface normals. If the viewpoint is fixed above, we can calculate the normals independently per pixel; if the visual-hull scenario in (c) happens, it can shift the depth discontinuity from (b) to a discontinuity-free shape in (d) [28].

the 3d space, the height field is differentiated into the gradient domain, with large gradients being attenuated corresponding to steep scene elements, and on top of that, Gaussian filters [34], bilateral filters [35], etc, are used to highlight the key features and eliminate discontinuities [26] (Figure 3).

Wei *et al*. proposed from another direction with generalized surface-from-gradient methods to deform a 3d mesh to achieve the task of relief texture removal [31]. With a guided mesh normal filtering that calculates the relative total variation measure of geometric flatness within facets, adaptive patches with the maximal geometry flatness could be obtained.

More recently, surface normal representations of input appearance are widely used. Since bas-reliefs have normals of a different 3d scene resembling its thin surface normals, leading to a depth sensation from specific viewpoints. Schüller *et al*. proposed a mesh parameterization following strict depth constraints to be optimized globally for delivering the desired appearance. During the process, the meshes are being deformed but with preservation of the original connectivity. Since per-vertex depth constraint can be satisfied exactly, the input can be projected to arbitrary and disconnected shapes [36].

Since a typical 3d model can be decomposed into a piecewise smooth base layer and a detailed layer in the normal field, corresponding to the structure-preserving and detail-preserving needs for bas-relief modeling, Wei *et al*. proposed a discrete approach that first decompose with Gaussian Mixture Model assisting with the rolling-guidance normal filter, and then recompose with local discrete shaping and global blending to model bas-reliefs [28]. Wang *et al*. similarly, used a Domain Transfer Recursive Filter for geometry preservation, but applied also visual attention based mask with Deep Visual Attention Network providing highlighted area and LG-LSTM Network generating part segmentations to augment the richness of details [37].

C. Image Style Transfer

Image style transfer is often considered to be a generalized version of the texture synthesis task with texture extraction and transformation from one to another (Figure 4 (a)). Quantitative methods model and extract local texture features statistically with relatively weak spatial constraints. Portilla *et al*. proposed a global wavelet coefficient constraints with correspondence to basis functions at adjacent spatial locations, orientations, and scales, and images are projected iteratively to satisfy the constraints [12]. The style is transferred through the process of image abstraction enabled with low-rank textures, and combinations of filters such as Gaussian filter kernels [38], non-linear bilateral filters [39], [40], or multiple convolution filter banks [41] are applied to meet certain constraints and complete the style transfer task [42], [43].

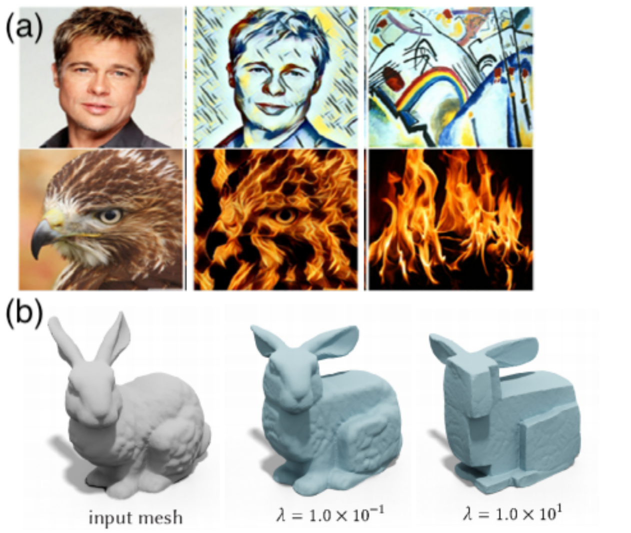
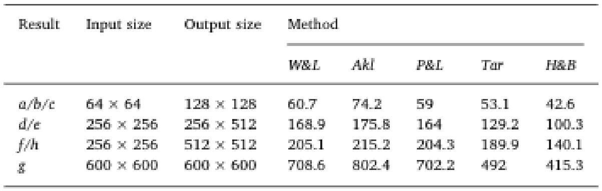


Fig. 4. (a) Image style transfer with the left one being the content, the right one being the style, and the middle one being the output after transformation [42] (b) 3D cubic stylization with λ parameter controling the cubeness [44]

On the other hand, neural style transfer gained lots of attention from Gatys *et al.*'s work on Convolutional Neural Networks with textures being represented by the feature map correlations of different layers of CNN [45]. The texture model in essence is similar to Portilla *et al*'s work on calculating spatial summary statistics on feature responses while the linear filter bank would be replaced by the correlations between feature responses in each layer of CNN [12], [45]. While the approaches overlap heavily with texture synthesis and often style transfer extends the work with an emphasis on constraining the content intact in selected regions, while producing a visually pleasing rich style in the target image. Specialized optimization adding for example a style swap layer [46], adaptive instance normalization layers [47], a downsampled semantic segmentation map [48], Markov random field acting as a dCNN feature pyramid [49], etc into the CNN model, or some post-processing such as joint bilateral upsampling filters for further refinement [39]. For a more comprehensive overview of neural style transfer, we would recommend [50], [51], [42].

For 3d stylization, many generative models have been applied, and among which, Liu *et al*. published an interesting work with respect to cubic stylization that aims at deforming 3d objects into cubes while still preserving textures and geometric features using energy-based geometric flows and mesh filters [44] (Figure 4 (b)). Chen *et al*. proposed a Decor-GAN that uses a 3D CNN generator for upsampling course models and a 3D PatchGAN discriminator to enforce similarities in style [10].



Result	Input size	Output size	Method				
			W&L	*Akl*	*P&L*	*Tar*	*H&B*
a/b/c	64 × 64	128 × 128	60.7	74.2	59	53.1	42.6
d/e	256 × 256	256 × 512	168.9	175.8	164	129.2	100.3
f/h	256 × 256	512 × 512	205.1	215.2	204.3	189.9	140.1
g	600 × 600	600 × 600	708.6	802.4	702.2	492	415.3

Fig. 5. Simulation time (in seconds) for the results texture synthesizing results. a/b/c/d/e/f/g/h corresponds to specific texture input labels. Method abbreviation's corresponding names can be found on Table I [54]

III. Methods for Texture Synthesis

2D texture synthesis has been intensively studied around the 1990s. There are two main categories: parametric texture modeling that describes a texture with sets of spatial summary statistics, and non-parametric texture modeling with Markov Random Fields (MRFs) that models a texture as a stationary local random process, featured with Efros and Leung's work that use MRF for finding similar neighborhoods [52]. In 2009, Wei *et al*. present a classic survey of the state-of-the-art for exemplar-based texture synthesis [53].

Akl *et al*. conducted multiple experiments and evaluated the efficiency between non-parametric and parametric algorithm (Table I) with results shown in Figure 5, and it is demonstrated that those parametric methods, in general, requires less time. Also, Akl *et al*. did multiple measurements within those methods and visualized the results in Figure 6 so that comparisons can be made between the methods output qualities [54].

names	year	abbreviations	categories
Wei and Levoy [13]	2000	W&L	non-parametric
Efros and Leung [52]	1999	E&L	non-parametric
Akl et al. [55]	2015	Akl	non-parametric
Efros andFreeman [52]	2001	E&F	non-parametric
Paget and Longstaff [56]	1998	P&L	non-parametric
Portilla and Simoncelli [12]	2000	P&S	parametric
Galerne et al. [57]	2011	Gal	parametric
Tartavel et al. [58]	2014	Tar	parametric
Heeger and Bergen [7]	1995	H&B	parametric
Gatys et al. [45]	2015	Gatys	parametric

TABLE I
Abbreviation and work correspondence for Figure 5 and Figure 6

Along with their work, we could get an overview that there are three subtasks for image synthesis: texture acquisition, texture mapping, and texture rendering. While for texturing a surface, the two main things are creating an orientation field over the surface, and performing texture synthesis according to the orientation field. Since then, the synthesis tasks are focused on more complex views and light fields, or focusing on specific domains such as art brushes, rich materials, etc.

A. Convolutional Neural Networks

Image optimization is more like the statistical parametric approaches aiming at feature matching between newly generated images and the example image. CNN plays an important role in giving characterization and feature selection of input images with the feature maps from its internal layers, and many studies have been done towards better matching measurements for the optimization process.

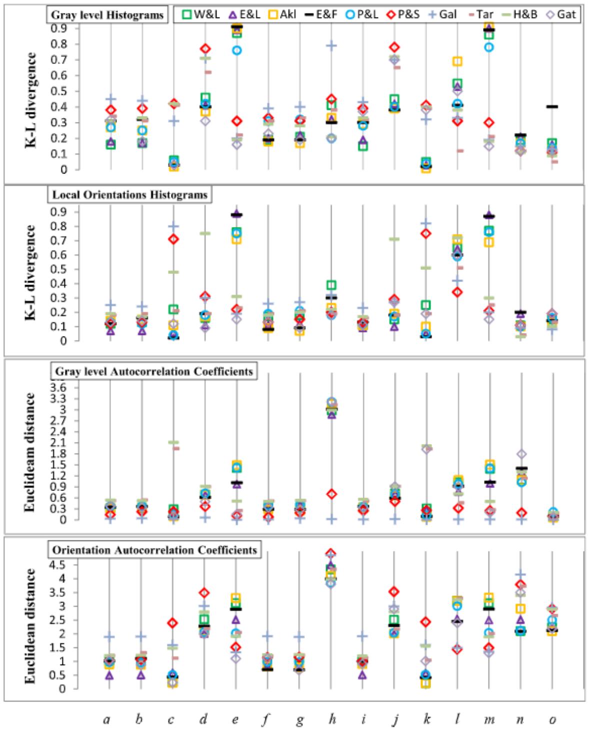


Fig. 6. Objective indicators calculated. a/b/c/d/e/f/g/h/i/j/k/l/o corresponds to specific texture input labels. Method abbreviation's corresponding names can be found on Table I [54]

After Gatys *et al*. proposed the CNN architecture for texture synthesis in 2015, Snelgrove used Gram matrix as a spatially invariant matching method that revealing the correlations between feature maps instead of statistical properties between the CNN layers, so that similar texture can be generated when the Gram matrix is close to the examples. Equation 5 defines the normalized Gram matrix G_l at layer l, e difference between the same layer of a source image and synthesized image E_l, and the minimization objectives,

$$
\begin{aligned}
G^l_{ij} &= \frac{1}{M_l N_l} \sum_k F^l_{ik} F^l_{jk} \\
E_l &= \sum_{i,j} (\hat{G}^l_i j - G^l_{ij})^2 \\
\mathcal{L}(\hat{\vec{x}}, \vec{x}) &= \sum_{l=0}^{L} w_l E_l
\end{aligned}
\tag{5}
$$

- l: layer index
- N_l: the number of feature maps at layer l
- M_l: vectoried size
- F_l: feature matrix for layer l with $F_l \in \mathbb{R}^{N_l \times M_l}$
- F^l_{jk}: activation at layer l of the j^th feature in position k
- G_l: Gram matrix at layer l summing across every neuron in that map with $G_l \in \mathbb{R}^{N_l \times N_l}$
- E_l: difference between the source image's Gram matrix G by taking squared Frobenius norm
- w_l: a selectable weighting factor

– $\mathcal{L}(\hat{\vec{x}}, \vec{x})$: L-BFGS optimizer[59] with analytic gradients since every operation is differentiable

Besides, layers across different scales are matching in a Gaussian pyramid resulting in a higher-resolution output [45], [60].

Similar to a Gram loss, Sendik and Cohen-Or introduced structural energy that captures the self-similarities with Deep Correlation (DCor) loss aiming at large-scale regularity, since the deep features rather than direct pixel values would be more adaptive to minor pixel-level local differences that has similar semantics [61]. Li and Wand added MRF priors that assume independence between pixels that are separated by more than a patch diameter, along with Gram Matrix under the CNN architecture, the nonparametric losses and local pattern constraints significantly improve the quality of synthesis [49].

The iterative optimization process is often replaced by feed-forward networks with the same minimization objectives because they are around 3 orders of magnitude faster [47]. Wang *et al*. proposed coarse-to-fine architecture to learn large brush strokes and maintain correct texture scales to enhance the granularity of feed-forward style transfer with a multi-resolution architecture. Perceptual factors are controlled during the stylization process by defining two perceptual loss terms that measure dissimilarity between generated image and the target with hieratical sub networks being trained at different style scales [62].

Ulynanov and Vedaldi proposed an instance normalization layer such that simply applying normalization to a single image rather than a batch of images leads to significant improvement in convergence type and stylization quality, since the desired style of each content is directly normalized so that the objective is simpler [63]. Nam *et al*. further explicitly manipulated style or texture information with batch-instance normalization which will be selective in the normalization process and only preserves useful textures [64].

Since the batch error may increase significantly when the size becomes smaller and factors like shape, frequency, and textures may lead to grouping, Wu and He, further, proposed group normalization that segments feature maps into groups and compute the mean and variance within each group to increase accuracy. Huang *et al*. proposed decorrelated batch normalization that added zero-phase component analysis data whitening to accelerate the training process [65]. Kim *et al*. applied the batch normalization into their neural net architecture to make the training process to be more stable given the complex and time-consuming earth texture with morphological geometry [66]. Huang *et al*. proposed another method to shorten the training time with texture feature distribution extraction then use the clustering algorithm so that discrete texture elements can be separated based on their image feature distribution before applying the texture transfer CNN [67].

For faster interactions, one might use lazy synthesis methods to only synthesize specific voxels compared to the whole volume. Dong *et al*. proposed such a method with a careful pre-processing selection of 2d neighborhoods forming a consistent strip [68]. It is suitable for real-time interactions with a parallelized algorithm that uses the time to trade-off for visual quality.

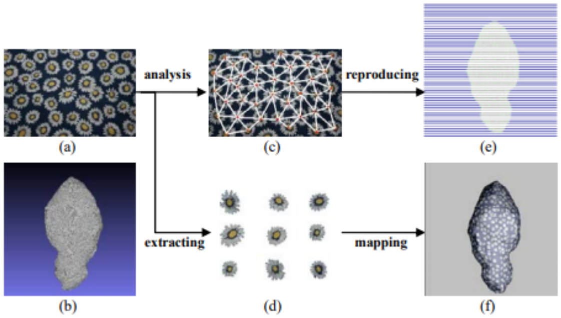


Fig. 7. An overview of synthesizing of discrete texture elements over the 3D surface (a) material (b) 3D mesh to apply the material on (c) connectivity of textures (d) extracted feature elements (e) synthesized element distribution (f) output result [72]

Deep pre-trained convolutional networks often suffer from the need for pre-trained large datasets and generalizability to non-local structures, while the conditional generative convolutional neural network (cgCNN) combines the deep texture statistics with a probabilistic framework. The addons to regular CNN are that it can learn the weights of the CNN for each input exemplar so that the textures can be synthesized in an integrated way and non-local structures could also be synthesized without extra constraints [69].

B. *Generative Adversarial Networks*

Generating textures over surfaces of 3D shapes rather than image textures requires a little bit of more work. For solid texture synthesis, procedural methods are widely used if there exist explicit mathematical formulations and functions for those highly structured textures such as Perlin and Gabor noise [70], [71], [57]. However, creating them with consideration of aesthetics is a trial and error process considering the methods' randomness and spatial coherence [72] (Figure 7).

Galerne *et al*. used a precomputed Kernel Texton containing the target spectrum to convolve with random points [73]. Even though the cost of generation is greatly reduced for by-example random-phase noise synthesis, it is hard to be generalized to general photorealistic textures, while for other methods, the surface geometric and connectivity integrity may be hardly preserved.

Most neural 3D surface texture synthesis is via Generative Adversarial Network (GAN) with feed-forward evaluation of the generating network but different training strategies, as CNNs are not scaled well to acceptable-resolution 3d textures. Neural networks that train on a single image seem to overfit, but for specific tasks like texture analysis, it can learn a useful representation of internal statistics to improve photorealism.

Portenier *et al*. proposed GramGAN to generate infinite high-quality 3D texture based on 2D exemplar image produced by a non-linear combination of noise frequencies learned from neural models. The synthesis network is trained to match the Gram matrices of deep features via a discriminator network, and generalization can be obtained via an additional extrapolation strategy that would fine-tune specific parameters in the model [74].

Henzler *et al*. focused on generating stochastic variations of textures, and proposed a multilayer perceptron that formulates and parameterizes the synthesis problem as a point operation so that memory efficiency can be achieved, and it is trained to match a similar VGG convolutional network statistics. And the model is defined to be a function of noise frequencies to enable diversity [75].

Zhou *et al*. applied a similar approach of GANs to tackle non-stationary texture synthesis that is spatially variant and inhomogenous. The generators learn how to expand the small subarea of the exemplar and inject new content into larger windows, and the style loss is computed with the pre-trained VGG model and Gram matrix is also calculated for comparing feature map outputs [76].

Yu *et al*. proposed a GAN model that goes beyond synthesizing but with by-example controllability and mixer of textures via realistic and smooth interpolation to the original image domain [77]. Two subtasks under this are reconstruction and interpolation, with the former making sure the texture would be similar after autoencoders, while the latter one ensuring the interpolated latent sensor encodes plausible textures with the use of a discriminator. Those image-specific GANs (inGAN) are able to acquire the same internal patch distribution regardless of significant changes in image size and shape, leading to integration between textures and natural image [78].

Hertz *et al*. applied GAN not on images, but on local patches of a single mesh to synthesize indistinguishable local mesh patches given a reference mesh with geometric textures [79]. Instead of 2D displacement maps, the depth features are constrained with the triangular mesh topology such that every triangle is within a fixed-sized convolutional neighborhood. Starting with a low-resolution mesh, the optimization process is in a scale hierarchy by subdividing faces with additional refinement on geometry, along with the input features describing the neighboring face relations [80]. Because of this setup texture synthesis no longer requires parameterization while still can make the transition between objects of different genus numbers.

GANs are powerful yet recently, Granot *et al*. revisited the classical patch-based methods (GPNN) and proposed a simple optimization-free framework that starts with course guess and searches in a coarse-to-fine nearest-neighbor fashion. They claimed that this approach is faster with superior results. Although with limitations in generalization and local blur without pixel-level gradients, It would be interesting to see further applications of this method. The discrete nature of GPNN might provide insight into optimizing some procedures for synthesizing discrete texture elements onto 3D surfaces. Also, for the single-image GANs with a small enough dataset, there might be scenarios for GPNN to come to the stage, as the author stated, it would generate samples with higher fidelity to the input [81].

IV. Conclusion

Addressing the issues of running performances, output quality, user control, generalizability, diversity, semantics understanding, and spatial plus temporal consistency are more or less appearing in every approach. Given a mature field of texture synthesis, the neural network seems to expand the scope of the task, and making the boundaries between other fields to be less visible. Among those methods and applications, transferring textures and synthesizing textures from 3D objects to 3D objects seems to be relatively less studied yet quite intriguing. If more time could be given, it would be interesting to start the search from those two papers ”Cubic Stylization” by Liu *et al*. [44] and ”Deep Geometric Texture Synthesis” by Hertz *et al*. [79], as well as including researches associated with maximum entropy methods and optical texture representation.

References

[1] E. H. Adelson, “On seeing stuff: the perception of materials by humans and machines,” in *Human vision and electronic imaging VI*, vol. 4299. International Society for Optics and Photonics, 2001, pp. 1–12.

[2] H. Komatsu and N. Goda, “Neural mechanisms of material perception: Quest on shitsukan,” *Neuroscience*, vol. 392, pp. 329–347, 2018.

[3] R. Rosenholtz, “Texture perception,” *Oxford handbook of perceptual organization*, vol. 167, p. 186, 2014.

[4] V. Hulusic, C. Harvey, K. Debattista, N. Tsingos, S. Walker, D. Howard, and A. Chalmers, “Acoustic rendering and auditory–visual cross-modal perception and interaction,” in *Computer Graphics Forum*, vol. 31, no. 1. Wiley Online Library, 2012, pp. 102–131.

[5] S. Malpica, A. Serrano, M. Allue, M. G. Bedia, and B. Masia, “Cross-modal perception in virtual reality,” *Multimedia Tools and Applications*, vol. 79, no. 5, pp. 3311–3331, 2020.

[6] C. Cavina-Pratesi, R. Kentridge, C. Heywood, and A. Milner, “Separate processing of texture and form in the ventral stream: evidence from fmri and visual agnosia,” *Cerebral Cortex*, vol. 20, no. 2, pp. 433–446, 2010.

[7] 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*, 1995, pp. 229–238.

[8] M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *International journal of computer vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[9] J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2021.

[10] Z. Chen, V. Kim, M. Fisher, N. Aigerman, H. Zhang, and S. Chaudhuri, “Decor-gan: 3d shape detailization by conditional refinement,” *arXiv preprint arXiv:2012.09159*, 2020.

[11] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma, “Face recognition with contiguous occlusion using markov random fields,” in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1050–1057.

[12] J. Portilla and E. P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *International journal of computer vision*, vol. 40, no. 1, pp. 49–70, 2000.

[13] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 479–488.

[14] N. Komodakis and G. Tziritas, “Image completion using efficient belief propagation via priority scheduling and dynamic pruning,” *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.

[15] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[16] R. W. Fleming, “Material perception,” *Annual review of vision science*, vol. 3, pp. 365–388, 2017.

[17] R. L. Cook and K. E. Torrance, “A reflectance model for computer graphics,” *ACM Transactions on Graphics (ToG)*, vol. 1, no. 1, pp. 7–24, 1982.

[18] M. Ashikhmin and P. Shirley, “An anisotropic phong brdf model,” *Journal of graphics tools*, vol. 5, no. 2, pp. 25–32, 2000.

[19] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, “Reflectance and texture of real-world surfaces,” *ACM Transactions On Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999.

[20] M. Haindl and V. Havlíček, “Transfer learning of mixture texture models,” in *International Conference on Computational Collective Intelligence*. Springer, 2020, pp. 825–837.

[21] D. den Brok, S. Merzbach, M. Weinmann, and R. Klein, "Per-image super-resolution for material btfs," in *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2020, pp. 1–10.

[22] X. Zhang, J. Dong, Y. Gan, H. Yu, and L. Qi, "Btf data generation based on deep learning," *Procedia computer science*, vol. 147, pp. 233–239, 2019.

[23] Y. Gan, H. Chi, Y. Gao, J. Liu, G. Zhong, and J. Dong, "Perception driven texture generation," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 889–894.

[24] G. Rainer, W. Jakob, A. Ghosh, and T. Weyrich, "Neural btf compression and interpolation," in *Computer Graphics Forum*, vol. 38, no. 2. Wiley Online Library, 2019, pp. 235–244.

[25] M. Maximov, L. Leal-Taixé, M. Fritz, and T. Ritschel, "Deep appearance maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8729–8738.

[26] Y.-W. Zhang, J. Wu, Z. Ji, M. Wei, and C. Zhang, "Computer-assisted relief modelling: A comprehensive survey," in *Computer Graphics Forum*, vol. 38, no. 2. Wiley Online Library, 2019, pp. 521–534.

[27] M. Wang, J. Chang, J. Kerber, and J. J. Zhang, "A framework for digital sunken relief generation based on 3d geometric models," *The Visual Computer*, vol. 28, no. 11, pp. 1127–1137, 2012.

[28] M. Wei, Y. Tian, W.-M. Pang, C. C. Wang, M.-Y. Pang, J. Wang, J. Qin, and P.-A. Heng, "Bas-relief modeling from normal layers," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 4, pp. 1651–1665, 2018.

[29] B. Jenny, M. Heitzler, D. Singh, M. Farmakis-Serebryakova, J. C. Liu, and L. Hurni, "Cartographic relief shading with neural networks," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[30] S. Liu, R. R. Martin, F. C. Langbein, and P. L. Rosin, "Segmenting geometric reliefs from textured background surfaces," *Computer-Aided Design and Applications*, vol. 4, no. 5, pp. 565–583, 2007.

[31] M. Wei, Y. Feng, and H. Chen, "Selective guidance normal filter for geometric texture removal," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[32] P. Cignoni, C. Montani, and R. Scopigno, "Computer-assisted generation of bas-and high-reliefs," *Journal of graphics tools*, vol. 2, no. 3, pp. 15–28, 1997.

[33] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, "Digital bas-relief from 3d scenes," *ACM transactions on graphics (TOG)*, vol. 26, no. 3, pp. 32–es, 2007.

[34] J. Kerber, A. Belyaev, and H.-P. Seidel, "Feature preserving depth compression of range images," in *Proceedings of the 23rd spring conference on computer graphics*, 2007, pp. 101–105.

[35] J. Kerber, A. Tevs, A. Belyaev, R. Zayer, and H.-P. Seidel, "Feature sensitive bas relief generation," in *2009 IEEE International Conference on Shape Modeling and Applications*. IEEE, 2009, pp. 148–154.

[36] C. Schüller, D. Panozzo, and O. Sorkine-Hornung, "Appearance-mimicking surfaces," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–10, 2014.

[37] M. Wang, L. Wang, T. Jiang, N. Xiang, J. Lin, M. Wei, X. Yang, T. Komura, and J. Zhang, "Bas-relief modelling from enriched detail and geometry with deep normal transfer," *Neurocomputing*, 2020.

[38] J. J. Van Wijk, "Spot noise texture synthesis for data visualization," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, 1991, pp. 309–318.

[39] A. Semmo, M. Trapp, J. Döllner, and M. Klingbeil, "Pictory: combining neural style transfer and image filtering," in *ACM SIGGRAPH 2017 Appy Hour*, 2017, pp. 1–2.

[40] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009.

[41] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1897–1906.

[42] L. Liu, Z. Xi, R. Ji, and W. Ma, "Advanced deep learning techniques for image style transfer: a survey," *Signal Processing: Image Communication*, vol. 78, pp. 465–470, 2019.

[43] C. Zhao, "A survey on image style transfer approaches using deep learning," in *Journal of Physics: Conference Series*, vol. 1453, no. 1. IOP Publishing, 2020, p. 012129.

[44] H.-T. D. Liu and A. Jacobson, "Cubic stylization," *arXiv preprint arXiv:1910.02926*, 2019.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[46] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," *arXiv preprint arXiv:1612.04337*, 2016.

[47] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[48] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," *arXiv preprint arXiv:1603.01768*, 2016.

[49] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2479–2486.

[50] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[51] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the" art": A taxonomy of artistic stylization techniques for images and video," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 5, pp. 866–885, 2012.

[52] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 1033–1038.

[53] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009, pp. 93–117.

[54] A. Akl, C. Yaacoub, M. Donias, J.-P. Da Costa, and C. Germain, "A survey of exemplar-based texture synthesis methods," *Computer Vision and Image Understanding*, vol. 172, pp. 12–24, 2018.

[55] ——, "Structure tensor based synthesis of directional textures for virtual material design," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4867–4871.

[56] R. Paget and I. D. Longstaff, "Texture synthesis via a noncausal nonparametric multiscale markov random field," *IEEE transactions on image processing*, vol. 7, no. 6, pp. 925–931, 1998.

[57] B. Galerne, Y. Gousseau, and J.-M. Morel, "Random phase textures: Theory and synthesis," *IEEE Transactions on image processing*, vol. 20, no. 1, pp. 257–267, 2010.

[58] G. Tartavel, Y. Gousseau, and G. Peyré, "Variational texture synthesis with sparsity and spectrum constraints," *Journal of Mathematical Imaging and Vision*, vol. 52, no. 1, pp. 124–144, 2015.

[59] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[60] X. Snelgrove, "High-resolution multi-scale neural texture synthesis," in *SIGGRAPH Asia 2017 Technical Briefs*, 2017, pp. 1–4.

[61] O. Sendik and D. Cohen-Or, "Deep correlations for texture synthesis," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 5, pp. 1–15, 2017.

[62] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5239–5247.

[63] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6924–6932.

[64] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," *arXiv preprint arXiv:1805.07925*, 2018.

[65] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 791–800.

[66] S. E. Kim, H. Yoon, and J. Lee, "Fast and scalable earth texture synthesis using spatially assembled generative adversarial neural networks," *arXiv preprint arXiv:2011.06776*, 2020.

[67] Z. Huang, X. Lin, and C. Chen, "Fast texture synthesis for discrete example-based elements," *IEEE Access*, vol. 8, pp. 76 683–76 691, 2020.

[68] Y. Dong, S. Lefebvre, X. Tong, and G. Drettakis, "Lazy solid texture synthesis," in *Computer Graphics Forum*, vol. 27, no. 4. Wiley Online Library, 2008, pp. 1165–1174.

[69] Z.-M. Wang, M.-H. Li, and G.-S. Xia, "Conditional generative convnets for exemplar-based texture synthesis," *IEEE Transactions on Image Processing*, vol. 30, pp. 2461–2475, 2021.

[70] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[71] A. Lagae, S. Lefebvre, G. Drettakis, and P. Dutré, "Procedural noise using sparse gabor convolution," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–10, 2009.

[72] J. Gutierrez, J. Rabin, B. Galerne, and T. Hurtut, "On demand solid texture synthesis using deep 3d networks," in *Computer Graphics Forum*, vol. 39, no. 1. Wiley Online Library, 2020, pp. 511–530.

[73] B. Galerne, A. Leclaire, and L. Moisan, “Texton noise,” in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 205–218.

[74] T. Portenier, S. Bigdeli, and O. Göksel, “Gramgan: Deep 3d texture synthesis from 2d exemplars,” *arXiv preprint arXiv:2006.16112*, 2020.

[75] P. Henzler, N. J. Mitra, and T. Ritschel, “Learning a neural 3d texture space from 2d exemplars,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8356–8364.

[76] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang, “Non-stationary texture synthesis by adversarial expansion,” *arXiv preprint arXiv:1805.04487*, 2018.

[77] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 164–12 173.

[78] A. Shocher, S. Bagon, P. Isola, and M. Irani, “Ingan: Capturing and retargeting the” dna” of a natural image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4492–4501.

[79] A. Hertz, R. Hanocka, R. Giryes, and D. Cohen-Or, “Deep geometric texture synthesis,” *arXiv preprint arXiv:2007.00074*, 2020.

[80] T. R. Shaham, T. Dekel, and T. Michaeli, “Singan: Learning a generative model from a single natural image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4570–4580.

[81] N. Granot, A. Shocher, B. Feinstein, S. Bagon, and M. Irani, “Drop the gan: In defense of patches nearest neighbors as single image generative models,” *arXiv preprint arXiv:2103.15545*, 2021.