Multiscale Exemplar Based Texture Synthesis by Locally Gaussian Models

Lara Raad(✉), Agnès Desolneux, and Jean-Michel Morel

CMLA, Ecole Normale Supérieure de Cachan, Cachan, France
lara.raad@cmla.ens-cahchan.fr

Abstract. In exemplar based texture synthesis methods one of the major difficulties is to synthesize correctly the wide diversity of texture images. So far the proposed methods tend to have satisfying results for specific texture classes and fail for others. Statistics-based algorithms present good results when synthesizing textures that have few geometric structures and are able to preserve a complex statistical model of the sample texture. On the other hand, non-parametric patch-based methods have the ability to reproduce faithfully highly structured textures but lack a mechanism to preserve its global statistics. Furthermore, they are strongly dependent on a patch size that is decided manually. In this paper we propose a multiscale approach able to combine advantages of both strategies and avoid some of their drawbacks. The texture is modeled at each scale as a spatially variable Gaussian vector in the patch space, which allows to fix a patch size fairly independent of the texture.

Keywords: Texture synthesis · Locally gaussian · Multiscale · Patch size

1 Introduction

Exemplar based texture synthesis is a well known problem that has many applications in computer graphics, computer vision and image processing, for example for fast scene generation, inpainting, and texture restoration. It is defined as the process of generating from an input texture sample a perceptually equivalent larger one. Texture synthesis algorithms are generally divided into two categories, the statistics-based [5,7,13] and the non-parametric patch-based [1–3,10–12,17]. The first category models a given texture sample by estimating statistical parameters that characterize the underlying stochastic process. Although these methods can faithfully reproduce some of the global statistics of the sample and synthesize micro and pseudo-periodic textures, they generally do not yield high quality visual results for more structured ones, in particular when the sample is small and contains large objects. The second category rearranges local neighbourhoods of the input sample in a consistent way. These methods provide efficient algorithms able to reproduce highly structured textures. Even though they yield visual satisfactory results, they often turn into practising verbatim copies of large parts of the input sample.

A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 435–443, 2015.
DOI: 10.1007/978-3-319-25751-8_52

Statistics-based methods are generally done in two steps: analysis and synthesis. The analysis step consists in identifying a set of global statistics from the input texture and the synthesis process generates an image satisfying the estimated set of statistics. These methods were inspired from Julesz [9], who discovered that many texture pairs having the same second-order statistics would not be preattentively discerned by humans. The success of Julesz first model can be checked in [5] where the authors propose to synthesize textures by randomizing the Fourier phase of the sample image while maintaining its Fourier modulus, thus preserving the second order statistics of the sample. These statistics are enough to synthesize micro-textures that can be characterized by their Fourier modulus but they fail for more structured ones as can be seen in [6]. Heeger and Bergen [7] initiated more sophisticated statistics-based methods describing the input sample by the histograms of its wavelet coefficients. A new texture is then created by enforcing these statistics on a white noise image. The results are satisfying for a small class of textures. Indeed the proposed statistics miss important correlations between scales and orientations, as can be verified in [8]. In [13] Portilla and Simoncelli extended [7] by estimating autocorrelations, cross-correlations and statistical moments of the wavelet coefficients of the texture sample. Compared to the previous statistical attempts, convincing results are observable on a very wide range of textures. Although this method represents the state of the art for psychophysically and statistically founded algorithms, the results nevertheless often present blur and phantoms effects.

Non-parametric patch-based methods were initialized by Efros and Leung [3] who extended to images Shannon's Markov random field model initially devised to simulate text. The synthesized texture is constructed pixelwise. For each new pixel, a patch centered at the pixel is compared to all patches with the same size in the input sample. The nearest matches help predict the pixel value in the reconstructed image. Several works [1,17] have extended and accelerated this method. Still these pixelwise algorithms are not always satisfactory. They are known to grow "garbage" when the compared patches are too small, or may lead to verbatim copies of significant parts of the input sample for large patches as can be verified in [4]. To overcome these drawbacks more recent methods stitch together entire patches instead of performing a synthesis pixel by pixel. The question then is how to blend a new patch in the existing texture. In [12] this is done by a smooth transition. Efros and Freeman [2] refined this process by stitching each new patch along a minimum cost path across its overlapping zone with the texture under construction. Kwatra et al. in [11] extended the stitching procedure of [2] by a graph cut approach redefining the edges of the patches. In [10] the authors propose to synthesize a texture image by sequentially improving the quality of the synthesis by minimizing a patch-based energy function. These non-parametric patch-based approaches often present satisfactory visual results. However, the risk remains of copying verbatim large parts of the input sample. Furthermore, a fidelity to the global statistics of the initial sample is not guaranteed, in particular when the texture sample is not stationary. See [18] for an extensive overview of the different neighbourhood-based methods.

More recently methods such as [15,16] combine patch-based and statistics-based methods to overcome the previous drawbacks. In [15] the author proposes to use a patch-based approach where all the patches of the synthesized image are created from a sparse dictionary learnt on the input sample. In [16] Tartavel et al. extend the work in [15] by minimizing an energy that involves a sparse dictionary of patches combined with constraints on the sample's Fourier spectrum. With the same motivation of avoiding verbatim-copies of the input sample and providing a statistical model of it, in [14] the authors proposed an algorithm that involves a local multivariate Gaussian texture model in the patch space.

Macro-textures show information at different scales that cannot be captured with a unique patch size. This motivates the extension of patch-based methods to a multiscale framework in the spirit of [1,16]. In this way the method is more robust to the patch size and avoids the blending step of patch-based approaches.

The rest of this paper is structured as follows. In Section 2 the multiscale approach is presented. In Section 3 two experiments are shown. The first one presents multiscale results and explores the impact of the scale interval on their efficiency. The second experiment shows how to combine two different synthesis methods: the multiscale locally Gaussian and the Portilla-Simoncelli statistical method [13]. Conclusions are presented in Section 4.

2 A Multiscale Algorithm

Macro textures have the particularity to present details at different scales: a coarse one containing the global structure and finer ones containing the details. On the one hand small patch sizes may capture the finer details of the input, yet if an algorithm is based only on them, the resulting texture will lack global coherence. On the other hand big patch sizes tend to a better respect of the global configuration but risk of a "copy-paste" effect. Furthermore, it becomes impossible to model the variability of large patches by curse of dimensionality: a texture sample will generally not contain enough patch samples. This is for example apparent in [14]. Multiscale approaches instead permit to contemplate several patch sizes within one synthesis (capture the different level of details).

In this section the potential of a multiscale approach is illustrated by improving the method in [14]. This approach can be summarized in a few sentences. The method begins by a synthesis in the coarsest scale ($k = K - 1$) using [14] where the quilting step is replaced by a simple average of the overlapping patches. For the remaining scales ($k = K - 2, \ldots, 0$) a synthesis is performed by using the result of the previous scale ($k + 1$) and the input of corresponding resolution. At each scale the synthesis is done patch by patch in a raster-scan order. Each new patch, added to the synthesized image, overlaps part of the previously synthesized patch and is the combination of a low resolution patch and a high resolution one sampled from a multivariate Gaussian distribution. The Gaussian distribution of the high frequencies of a given patch is estimated from the high frequencies of its m nearest neighbours in the corresponding scale input image. The synthesis result of the finer scale is the desired output image.

Notations

- $u_0 : \Omega \to \mathbb{R}$: input texture image. $\Omega = I_M \times I_N$ of size $M \times N$ where I_c is the discrete interval $[0, \dots, c-1]$
- $w_0 : \Omega_r \to \mathbb{R}$: output texture image. $\Omega_r = I_{rM} \times I_{rN}$ of size $rM \times rN$
- r: ratio (size of output image w_0)/(size of input image u_0)
- $n \times n$: patch size
- m: number of nearest neighbours used to learn the Gaussian distribution
- K: number of scales (maximum factor of zoom out is $K-1$)
- $u_k : \Omega^k \to \mathbb{R}$: zoom out of u_0 by a factor 2^k. $\Omega^k = I_{2^{-k}M} \times I_{2^{-k}N}$ of size $2^{-k}M \times 2^{-k}N$ for $k = 1 \dots K-1$
- $w_k : \Omega_r^k \to \mathbb{R}$: synthesized texture at scale k. $\Omega_r^k = I_{r2^{-k}M} \times I_{r2^{-k}N}$ of size $r2^{-k}M \times r2^{-k}N$ for $k = 0 \dots K-1$
- $v_k : \Omega_r^k \to \mathbb{R}$: zoom in of w_{k+1} by a factor 2 for $k = 0 \dots K-2$
- G_σ: Gaussian kernel centered of standard deviation σ
- $L_{u_k} : \Omega^k \to \mathbb{R}$: low resolution of u_k. $L_{u_k} = u_k * G_\sigma,\ k = 0 \dots K-2$
- $L_{w_k} : \Omega_r^k \to \mathbb{R}$: low resolution of w_k. $L_{w_k} = w_k * G_\sigma,\ k = 0 \dots K-2$
- $H_{w_k} : \Omega_r^k \to \mathbb{R}$: high resolution of w_k. $H_{w_k} = w_k - w_k * G_\sigma,\ k = 0 \dots K-2$
- $p_u^{(x,y)}$: square patch of size $n \times n$ from an image u of size $M \times N$ at position (x, y). $p_u^{(x,y)} = \{u((x,y)+(i,j)),\ (i,j) \in [0, \dots, n-1]^2\},\ (x,y) \in \mathcal{V}_u = I_{M-n+1} \times I_{N-n+1}$
- $\mathcal{Z}_2^{\text{out}}(u)$: zoom out by a factor 2 of image u performed as a smooth frequency cutoff followed by a sub-sample of factor 2
- $\mathcal{Z}_2^{\text{in}}(u)$: zoom in by a factor 2 of image u performed by a zero padding of the discrete Fourier transform of u

Distance Between Patches

To estimate the parameters of the Gaussian distribution of the patch being processed, denoted by $p_{w_k}^{(x',y')}$, the set $\mathcal{U}$ of m nearest patches in u_k to $p_{w_k}^{(x',y')}$ is considered. These patches are those minimizing the distance to $p_{w_k}^{(x',y')}$ defined in (1) for $k = K-1$ and in (2) for the remaining scales $k = K-2, \dots, 0$.

The size of patch overlap is fixed to half the patch size $n/2$. Depending on the stage of the synthesis three different cases of overlap can be observed: vertical (first row of raster-scan)(VO), horizontal (first column of raster-scan)(HO) and L-shape (everywhere else)(LO). The overlap area of a patch $p_u^{(x,y)}$ is denoted as $Op_u^{(x,y)} = \{u((x,y)+(i,j)),\ (i,j) \in \mathcal{O}\}$ where

$$\mathcal{O} = \begin{cases} [0, \dots, n-1] \times \left[0, \dots, \frac{n}{2}-1\right] & \text{if VO} \\ \left[0, \dots, \frac{n}{2}-1\right] \times [0, \dots, n-1] & \text{if HO} \\ \left[0, \dots, \frac{n}{2}-1\right] \times [0, \dots, n-1] \cup \left[\frac{n}{2}, \dots, n-1\right] \times \left[0, \dots, \frac{n}{2}-1\right] & \text{if LO} \end{cases}$$

When $k = K-1$ the m nearest neighbours in u_{K-1} to the patch $p_{w_{K-1}}^{(x',y')}$ are those minimizing the L^2 distance restricted to the overlap area (1).

$$d(p_{u_{K-1}}^{(x,y)}, p_{w_{K-1}}^{(x',y')})^2 = \frac{1}{|\mathcal{O}|} \sum_{(i,j)\in\mathcal{O}} (u_{K-1}(x+i, y+j) - w_{K-1}(x'+i, y'+j))^2 \tag{1}$$

When $k = K-2, \ldots, 0$ the nearest neighbours in u_k to the patch $p_{w_k}^{(x',y')}$ are those minimizing a distance (2) similar to (1) with an additional term taking into account the low resolution v_k (the synthesis result of the previous scale $k+1$). Is is important to notice that when comparing $Op_{u_k}^{(x,y)}$ and $Op_{w_k}^{(x',y')}$ the low and the high resolution must be considered jointly, they are not independent.

$$\begin{aligned} d(p_{u_k}^{(x,y)}, p_{w_k}^{(x',y')})^2 =& \frac{1}{|\mathcal{O}|} \sum_{(i,j)\in\mathcal{O}} (u_k(x+i, y+j) - w_k(x'+i, y'+j))^2 \\ &+ \frac{1}{n^2} \sum_{i,j=0}^{n-1} (L_{u_k}(x+i, y+j) - v_k(x'+i, y'+j))^2 \end{aligned} \tag{2}$$

The Gaussian Model and the Blending Process

Every patch $p_{w_k}^{(x',y')}$ in w_k for $k = 0, \ldots, K-1$ is sampled from a multivariate Gaussian distribution in the spirit of [14]. The parameters (μ, Σ) of the distribution of $p_{w_k}^{(x',y')}$ are estimated on the set $\mathcal{U} = \{p_{u_k}^{(x_1,y_1)}, \ldots, p_{u_k}^{(x_m,y_m)}\}$ as in (3). Here $p_{u_k}^{(x_i,y_i)}$, for $i = 1, \ldots, m$, are the m nearest patches to $p_{w_k}^{(x',y')}$ in u_k for the distances in (1) and (2).

$$\mu = \frac{1}{m} \sum_{i=1}^{m} q_{u_k}^{(x_i,y_i)}, \quad \Sigma = \frac{1}{m} QQ^t \tag{3}$$

In (3) $q_{u_k}^{(x_i,y_i)}$ is the patch $p_{u_k}^{(x_i,y_i)}$ in vector form and Q is the matrix whose columns are $(q_{u_k}^{(x_i,y_i)} - \mu)$, $i = 1, \ldots, m$. Sampling a patch $\tilde{p} \sim G(\mu, \Sigma)$ comes down to sampling m independent normal variables as can be seen in (4).

$$\tilde{q} = \frac{1}{m} Q^t QWDq' + \mu \tag{4}$$

Here $\tilde{q}$ is the vector form of $\tilde{p}$, $q' \sim G(\mathbf{0}, I_m)$, W is a matrix whose columns are the eigenvectors of Q^tQ and D is a diagonal matrix with its eigenvalues.

The blending process consists in simply averaging the values across the overlap area as in (5). This step is applied only for the synthesis at scale $k = K-1$.

$$w_k(x'+i, y+j') = \begin{cases} \frac{1}{2}\left(\tilde{p}(i,j) + p_{w_{K-1}}^{(x',y')}\right) & \text{if } (i,j) \in \mathcal{O} \\ \tilde{p}(i,j) & \text{if } (i,j) \in I_n^2 - \mathcal{O} \end{cases} \tag{5}$$

Synthesizing Patches at Scales $k = K-2, \ldots, 0$

At each scale k a patch $p_{w_k}^{(x',y')}$ is synthesized as the combination of a low resolution patch with a high resolution one. It can be decomposed as follows

$$\begin{aligned} p_{w_k}^{(x',y')} &= p_{w_k * G_\sigma}^{(x',y')} + (p_{w_k}^{(x',y')} - p_{w_k * G_\sigma}^{(x',y')}) = p_{v_k}^{(x',y')} + (p_{w_k}^{(x',y')} - p_{v_k}^{(x',y')}) \\ &= p_{L_{w_k}}^{(x',y')} + p_{H_{w_k}}^{(x',y')}. \end{aligned}$$

The set $\mathcal{U}$ define the Gaussian distribution of $p_{L_{w_k}}^{(x',y')} \sim G(\mu_L, \Sigma_L)$ and $p_{H_{w_k}}^{(x',y')} \sim G(\mu_H, \Sigma_H)$ and therefore the distribution of the patch $p_{w_k}^{(x',y')} \sim G(\mu, \Sigma)$ where $\mu = \mu_H + \mu_L$ and $\Sigma = \Sigma_L + \Sigma_H + \mathbb{E}(p_{L_{w_k}}^{(x',y')}(p_{H_{w_k}}^{(x',y')})^t) + \mathbb{E}(p_{H_{w_k}}^{(x',y')}(p_{L_{w_k}}^{(x',y')})^t)$. Instead of sampling $p_{L_{w_k}}^{(x',y')}$ from its Gaussian distribution, $p_{v_k}^{(x',y')} \sim G(\mu_L, \Sigma_L)$ is kept to conserve the low resolution synthesis from the previous scale. The high frequency patch $p_{H_{w_k}}^{(x',y')}$ is sampled form $G(\mu_H, \Sigma_H)$ and then added to $p_{v_k}^{(x',y')}$. In this way the correlations between high and low resolution pixels are respected, using the low resolution synthesis v_k as initialization.

3 Experiments

All the texture examples in Figures 1 and 2 can be found at http://dev.ipol.im/~lraad/ciarp_2015/. The experiments shown in Figure 1 compare the multiscale method using one, two and three scales. This is performed for micro- and macro-textures. For all the experiments the side patch size is fixed to $n = 20$ and the number of nearest neighbours to $m = 20$. Figure 1 shows that using a single scale is not enough to reproduce faithfully the global structure of the input example. Naturally to achieve satisfying synthesis results for $K = 1$ a bigger patch size should be considered. Still this would lead to limitations on the Gaussian model [14]. A fix patch size was sufficient to achieve satisfying results on all examples shown. Another positive aspect of using smaller patches is that one can find more reliable examples in the input sample to build the multivariate Gaussian distributions. Finally complex quilting steps like those used in [2,10,11] is no longer necessary. It can be replaced by an average of the values along the overlap zone for the synthesis of the coarsest scale. This is possible since at $k = K-1$ the images are smoother and an average is then well suited. In general it is enough to average the overlapping parts only at the coarser scales.

In Figure 1 the experiments show that for the three different cases ($K = 1, 2, 3$) the Gaussian synthesis entails a slight blur. To recover these fine details at scale $k = 0$ an additional step can be applied to the multiscale synthesis result. The output image w_0 is combined to Portilla and Simoncelli's method [13]. In [13] the synthesis image is initialized with a white noise. For this experiment Portilla and Simoncelli's method is initialized with the result of the multiscale method instead of a random noise. In Figure 2 the result of the multiscale approach

is compared to the results of combining the algorithms and to those of [13]. They show that the granularity of the input texture is globally recovered. The resulting texture respects the global statistics of the input imposed by [13] while maintaining the structures that are lost if the method is initialized with a white noise. Some of the example images in Figures 1 and 2 were provided from [13].

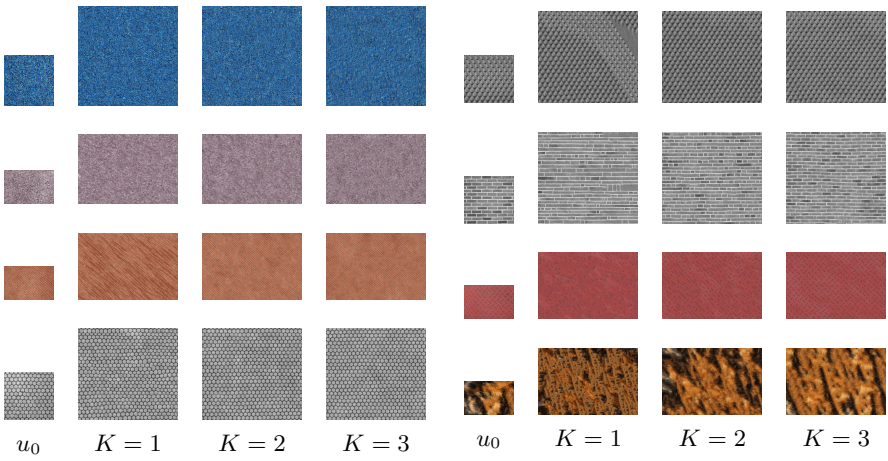

Fig. 1. Synthesis results for $K = 1, 2, 3$ scales. The parameters were fixed to $n = 20$ and $m = 20$. *It is recommended to zoom in the images by a factor 400% to evaluate texture details.*

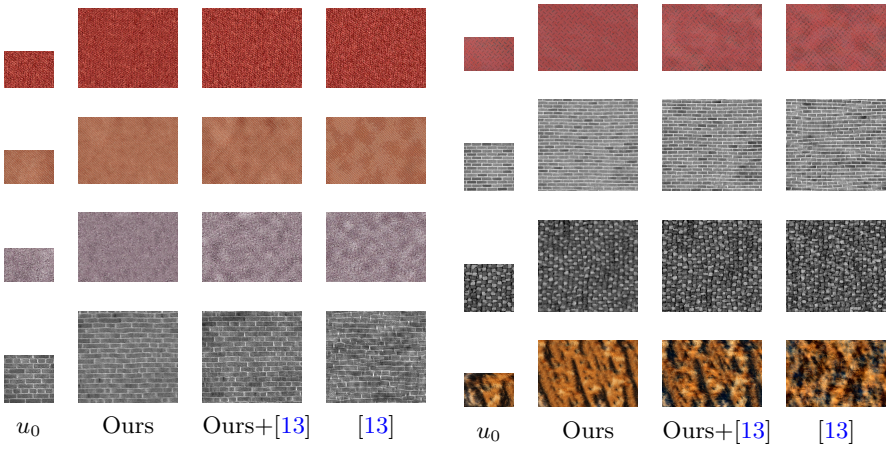

Fig. 2. Comparison of several texture synthesis algorithms: ours, ours combined to [13] and [13]. The parameters were fixed to $n = 20$, $m = 20$ and $K = 3$. *It is recommended to zoom in the images by a factor 400% to evaluate texture details.*

4 Conclusion

In this paper a multiscale approach of the locally Gaussian texture synthesis algorithm [14] was proposed. A first synthesis is performed at the coarsest scale to generate the global structure of the synthesized image. For the remaining scales the corresponding finer details are added on the low resolution result of the previous scale and so on until the finer scale is reached. The experiments showed that a single patch size can be used for different type of textures achieving satisfying visual results. A second observation is that due to the use of Gaussian models the synthesis results lose some resolution compared to the input sample. To recover its granularity the multiscale algorithm was combined with Portilla and Simoncelli's method [13]. The results showed that this combination is able to preserve the strong geometric structures and at the same time respect the global statistics of the sample that are imposed with [13].

Acknowledgments. Work partly founded by the European Research Council (advanced grant Twelve Labours) and the Office of Naval research (ONR grant N00014-14-1-0023).

References

1. Ashikhmin, M.: Synthesizing natural textures. In: Proceedings of the 2001 Symposium on Interactive 3D graphics, pp. 217–226. ACM (2001)
2. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH, pp. 341–346 (2001)
3. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE ICCV, pp. 1033–1038 (1999)
4. Aguerrebere, C., Gousseau, Y., Tartavel, G.: Exemplar-based Texture Synthesis: the Efros-Leung Algorithm. IPOL **3**, 223–241 (2013). http://dx.doi.org/10.5201/ipol.2013.59
5. Galerne, B., Gousseau, Y., Morel, J.-M.: Random phase textures: Theory and synthesis. IEEE Transactions in Image Processing (2010)
6. Galerne, B., Gousseau, Y., Morel, J.-M.: Micro-Texture Synthesis by Phase Randomization. IPOL (2011). http://dx.doi.org/10.5201/ipol.2011.ggm_rpn
7. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: SIGGRAPH, New York, NY, USA, pp. 229–238 (1995)
8. Briand, T., Vacher, J., Galerne, B., Rabin, J.: The Heeger and Bergen Pyramid Based Texture Synthesis Algorithm. IPOL **4**, 276–299 (2014). http://dx.doi.org/10.5201/ipol.2014.79
9. Julesz, B.: Visual pattern discrimination. IEEE Trans. Inf. Theory **8**(2), 84–92 (1962)
10. Kwatra, V., Essa, I., Bobick, A., Kwatra, N.: Texture optimization for example-based synthesis. In: ACM Transactions on Graphics (TOG), vol. 24, pp. 795–802. ACM (2005)
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. In: ACM Transactions on Graphics (TOG), vol. 22, pp. 277–286. ACM (2003)

12. Liang, L., Liu, C., Xu, Y.-Q., Guo, B., Shum, H.-Y.: Real-time texture synthesis by patch-based sampling. ACM Transactions on Graphics **20**(3), 127–150 (2001)
13. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. IJCV **40**(1), 49–70 (2000)
14. Raad, L., Desolneux, A., Morel, J.M.: Locally gaussian exemplar based texture synthesis. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 4667–4671. IEEE, October 2014
15. Peyré, G.: Sparse modeling of textures. Journal of Mathematical Imaging and Vision **34**(1), 17–31 (2009)
16. Tartavel, G., Gousseau, Y., Peyré, G.: Variational texture synthesis with sparsity and spectrum constraints. Journal of Mathematical Imaging and Vision (2014)
17. Wei, L.-Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: SIGGRAPH, pp. 479–488 (2000)
18. Wei, L.Y., Lefebvre, S., Kwatra, V., Turk, G.: State of the art in example-based texture synthesis. In: Eurographics 2009, State of the Art Report, EG-STAR, pp. 93–117. Eurographics Association (2009)