

Texture Synthesis: From Convolutional RBMs to Efficient Deterministic Algorithms

Qi Gao and Stefan Roth

Department of Computer Science, TU Darmstadt

Abstract. Probabilistic models of textures should be able to synthesize specific textural structures, prompting the use of filter-based Markov random fields (MRFs) with multi-modal potentials, or of advanced variants of restricted Boltzmann machines (RBMs). However, these complex models have practical problems, such as inefficient inference, or their large number of model parameters. We show how to train a Gaussian RBM with full-convolutional weight sharing for modeling repetitive textures. Since modeling the local mean intensities plays a key role for textures, we show that the covariance of the visible units needs to be sufficiently small – smaller than was previously known. We demonstrate state-of-the-art texture synthesis and inpainting performance with many fewer, but structured features being learned. Inspired by Gibbs sampling inference in the RBM and the small covariance of the visible units, we further propose an efficient, iterative deterministic texture inpainting method.

1 Introduction

Modeling prior knowledge on images and scenes is an important research problem in computer vision. Somewhat different challenges arise when only considering specific kinds of images or scenes, since the specific structure of the data needs to be taken into account. For example, visual textures, even though playing a large part in the composition of natural images, cannot be modeled well by directly applying the ideas for building generic image priors. The major reason is that generic image priors mainly consider the smoothness and continuity of the image, while texture models have to capture the specific textural structures.

To this end, the seminal FRAME texture model [16] uses Markov random fields (MRFs) with non-parametric, multi-modal potentials to allow for spatial structure generation. More recently, Heess *et al.* [5] suggested an MRF with parametric bi-modal potentials, which can be learned alongside the filters (features). Another class of probabilistic texture models extends restricted Boltzmann machines (RBMs) [6] toward capturing the spatial structure of textures, *e.g.* work by Kivinen *et al.* [7] and Luo *et al.* [8]. Common to these MRF- or RBM-based texture models is that they can be interpreted as a conditional Gaussian random field whose parameters are controlled by discrete latent variables. Moreover, all of them simultaneously perform regularization and generate textural structures through modeling the conditional covariance and mean, respectively. Due to their complex “mean+covariance” construction, these models are not

easy to train in practice. Some compromises toward stabilization of training, *e.g.* tiled-convolutional weight-sharing [7,8], can be detrimental to the quality of the generated textures. Moreover, the relative importance of the mean vs. the covariance component of these models is unclear in light of modeling textures.

In this paper we ask how important the regularization effect from the covariance component is for modeling textures. To that end, we explore the ability and efficiency of “mean” only RBMs for modeling Brodatz texture images (www.ux.uis.no/~tranden/brodatz.html). We learn full-convolutional Gaussian RBMs (cGRBs), for which the conditional Gaussian distribution has fixed identity covariance, and its mean is determined by the hidden variables. Block Gibbs sampling can be used for efficient learning and inference. Most importantly, we find that to learn good cGRB texture models, the covariance of the visible units needs to be sufficiently small, quite a bit smaller than the typical best practice [6]. Similarly, the coefficient norms of the convolutional features must be carefully constrained during learning. Our contributions are threefold: First, our learned convolutional RBMs have several favorable properties – simplicity, efficiency, spatial invariance, and a comparatively small number of structured, more interpretable features – yet they outperform more complex state-of-the-art methods in repetitive texture synthesis and inpainting. Second, as the conditional covariance of the visibles is a diagonal matrix with small variance, we show that the “mean” units actually take the most important role in modeling textures. Third, inspired by the procedure of inference through block Gibbs sampling, we further propose an efficient deterministic method for texture inpainting based on the learned features.

Other Related Work. Hao *et al.* [4] modified high-order Gaussian gated Boltzmann machines for texture modeling, and also directly model the dependencies between two visible units. They learned 1000 features with convolutional weight sharing and achieved good texture classification performance. The performance in texture generation are less satisfactory, however, with block artifacts appearing in texture inpainting results. Efros *et al.* [1,2] proposed non-parametric methods for texture synthesis. In [2] the synthesized image is grown by one pixel at a time according to the best matches between its neighbors and patches from the texture. [1] instead stitches together small patches directly.

2 Texture Modeling and Mean Units

2.1 MRF Models Based on Filters and Potentials

A typical way of formulating a prior for a generic image \mathbf{x} is through modeling the response to some linear filters with potential functions [3]:

$$p_{\text{MRF}}(\mathbf{x}; \Theta) \propto \prod_{c,j} \phi(\mathbf{f}_j^T \mathbf{x}_c; \theta_j), \quad (1)$$

where \mathbf{x}_c denotes the pixels of clique c , \mathbf{f}_j are linear filters, and $\phi(\cdot; \theta_j)$ are the potential functions. As the filter responses usually have heavy-tailed empirical distributions around zero, the potential functions are also chosen to be



Fig. 1. Tiled-convolutional (left) and full-convolutional (right) weight sharing. Lines converging to the hidden units (shaded) are the filters; they share their parameters when indicated using the same color or line type.

heavy-tailed (*e.g.*, Student-*t*). Many heavy-tailed potentials can be formulated as Gaussian scale mixtures (GSMs) [10]. For better understanding and more efficient inference, such GSM potentials allow augmenting the prior with hidden variables $\mathbf{h} = (h_{jc})_{j,c}$, one for each filter \mathbf{f}_j and clique c , which represent the index of the Gaussian mixture component modeling the filter response [12]. It holds that $p_{\text{MRF}}(\mathbf{x}; \Theta) \propto \sum_{\mathbf{h}} p_{\text{MRF}}(\mathbf{x}, \mathbf{h}; \Theta)$. Given the hidden variables, the conditional distribution for the image is a zero-mean Gaussian

$$p_{\text{MRF}}(\mathbf{x}|\mathbf{h}; \Theta) \propto \mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma(\mathbf{h}, \Theta)). \quad (2)$$

As changing the hidden units only changes the conditional covariance, such basic image priors focus on modeling the covariance structure of the image, which is intuitive as they are primarily aimed at regularization.

Heess *et al.* [5] showed that such generic MRF priors for natural images are not suitable for textures, and propose to extend them using bi-modal potential functions. Multi-modal potentials can also be modeled with Gaussian mixtures, however the components may no longer all have zero means. Given the hidden units, the conditional distribution of such an MRF texture model

$$p_{\text{MRF}_t}(\mathbf{x}|\mathbf{h}; \Theta) \propto \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}(\mathbf{h}, \Theta), \Sigma(\mathbf{h}, \Theta)) \quad (3)$$

shows that bi-modal potentials capture not only the covariance structure, but also the local mean intensities. The seminal FRAME texture model [16] with its non-parametric potentials is also consistent with this observation. Comparing Eq. (3) with Eq. (2) suggests that modeling the conditional mean is a particular trait of texture models. The intuitive explanation is that the model does not “just” want to perform regularization, but instead generate textural structure.

Note that in these models the filters are applied convolutionally across the entire image. Since filters can be understood as weights connecting visible and hidden units, this is called convolutional weight sharing (*cf.* Fig. 1). Importantly, this keeps the number of parameters manageable, even on images of an arbitrary size, and also gives rise to the model’s spatial invariance.

Nonetheless, learning such a “mean+covariance” model is difficult in practice, since the hidden units affect both conditional mean and covariance in complex ways. Since the filters need to be sufficiently large to generate coherent structures, the resulting covariance matrix will furthermore be quite dense, making both learning and inference rather inefficient. Moreover, the learned texture filters from [5] lack clear structure (see Fig. 2), making them difficult to interpret.

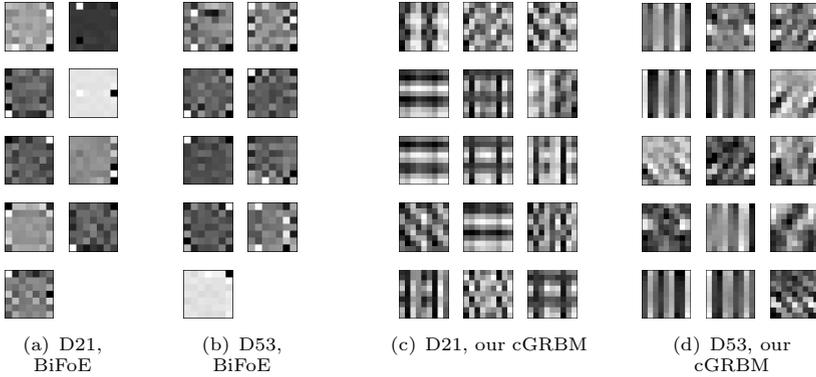


Fig. 2. Comparison of learned texture filters/features

2.2 Boltzmann Machine Models

Models derived from restricted Boltzmann machines (RBMs) take a different route. A Gaussian RBM [6] models an image by defining an energy function over visible units \mathbf{x} (here, the image pixels) and binary hidden units \mathbf{h} . The random variables have a Boltzmann distribution $p_{\text{RBM}}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp\{-E_{\text{RBM}}(\mathbf{x}, \mathbf{h})\}$, where Z is the partition function. Gaussian RBMs have the property that the conditional distribution of the visible units given the hidden ones is a Gaussian

$$p_{\text{RBM}}(\mathbf{x}|\mathbf{h}; \Theta) \propto \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}(\mathbf{h}, \Theta), \boldsymbol{\Sigma}), \quad (4)$$

in which only the conditional mean depends on the hidden variables.

More recent variants of Boltzmann machines for texture modeling [7,8] not only model the conditional mean, but also the covariance akin to Eq. (3). [7] experimentally compared three models: Gaussian RBMs, Products of Student-t (PoT) [15], and their combination, which corresponds to modeling conditional mean, covariance, and mean+covariance, respectively. The results revealed the importance of the conditional mean for texture synthesis and inpainting.

Note that both [7,8] adopt tiled-convolutional weight sharing [11] (*cf.* Fig. 1). The apparent reason is that the states of hidden units are less correlated, thus making training of the models easier. Unfortunately, tiled-convolutional models involve many parameters, since several sets of features (filters) must be learned. For example, [7,8] learn and use more than 300 features for every texture, which are moreover not spatially invariant. Consequently, tiled-convolutional weight sharing requires copious training data, which for textures is often not available.

3 Learning Full-Convolutional RBMs for Textures

Mean units appear to be an important component of many texture models. As these models also include covariance units and/or complex weight sharing, it is not clear how important the mean units are. We now investigate this and explore the capability of “mean-only” Gaussian RBMs for textures.

3.1 Convolutional Gaussian RBM

A spatially invariant model is obtained through applying the Gaussian RBM convolutionally to all overlapping cliques of a large texture image. The energy function of the convolutional Gaussian RBM (cGRBM) is then written as

$$E_{cGRBM}(\mathbf{x}, \mathbf{h}) = \frac{1}{2\gamma} \mathbf{x}^T \mathbf{x} - \sum_{c,j} h_{jc} (\mathbf{w}_j^T \mathbf{x}_c + b_j), \quad (5)$$

where we add a weight γ to the quadratic term. Here \mathbf{w}_j determine the interaction between pairs of visible units \mathbf{x}_c and hidden units h_{jc} . Thus \mathbf{w}_j are the features or filters, b_j are the biases, c and j are indices for all overlapping image cliques and filters, respectively. The conditional distribution of \mathbf{x} given \mathbf{h} is a Gaussian

$$p_{cGRBM}(\mathbf{x}|\mathbf{h}) \propto \mathcal{N}\left(\mathbf{x}; \gamma \sum_{c,j} h_{jc} \mathbf{w}_{jc}, \gamma \mathbf{I}\right), \quad (6)$$

where the vector \mathbf{w}_{jc} is defined as $\mathbf{w}_{jc}^T \mathbf{x} = \mathbf{w}_j^T \mathbf{x}_c$. The conditional of \mathbf{h} given \mathbf{x} is a simple logistic sigmoid function

$$p_{cGRBM}(h_{jc}|\mathbf{x}) \propto \text{logsig}(\mathbf{w}_{jc}^T \mathbf{x} + b_j). \quad (7)$$

3.2 Data

For a fair comparison with other models [5,7,8], we follow their use of the Brodatz texture images for training and testing our models. The images are rescaled to either 480×480 or 320×320 , while preserving the major texture features, and then are normalized to 0-mean and unit standard deviation. We also divide each image into a top half for training and a bottom half for testing.

3.3 Learning

As the partition function of the model is intractable, we perform approximate maximum likelihood (ML) learning based on persistent contrastive divergence (PCD) [13]. Model samples are obtained using standard, efficient block Gibbs sampling, which alternately samples the visibles \mathbf{x} or the hidden \mathbf{h} given the other. The parameters are updated using gradient ascent, *e.g.* for filters using

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + \eta \left[\left\langle \frac{\partial E(\mathbf{x})}{\partial \mathbf{w}_j} \right\rangle_{\mathbf{X}^{\text{PCD}}} - \left\langle \frac{\partial E(\mathbf{x})}{\partial \mathbf{w}_j} \right\rangle_{\mathbf{X}^0} \right], \quad (8)$$

where η is the learning rate, $\langle \cdot \rangle$ denotes the average over the training data \mathbf{X}^0 or the samples \mathbf{X}^{PCD} .

The standard learning procedure [6], however, does not ensure that a good convolutional RBM texture model is learned in practice. *E.g.* even the simple mean-only RBM baseline of [7] stabilizes learning using tiled-convolutional

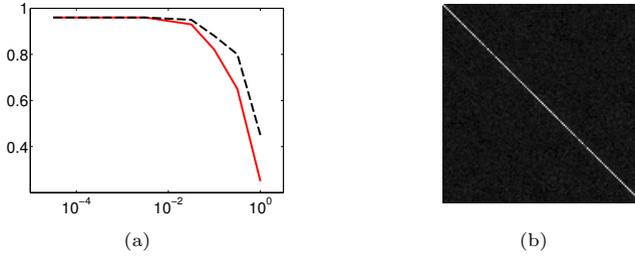


Fig. 3. (a) Texture similarity scores (TSS) of synthesized textures (black, dashed) and model samples (red, solid) vs. the choice of γ . (b) The covariance matrix of 1000 samples of \mathbf{h} when $\gamma = 0.03$. Results are based on D21.

weight sharing and a slowly mixing Hybrid Monte Carlo (HMC) sampler. Consequently, care must be taken to be able to train a cGRBM for textures.

Choice of Parameter γ . The typical best practice in Gaussian RBMs is to set the weight γ to 1 when the training data is normalized [6]. But we find that $\gamma = 1$ is far from the optimum and its value can greatly affect the generative properties of the trained texture models. Fig. 3(a) shows how γ changes the texture similarity score (TSS) [5] (see Sec. 3.4 for details) of model samples and synthesized textures. Actually, since a texture sample drawn with the Gibbs sampler is a sum of the conditional mean and *i.i.d.* Gaussian noise, $\gamma = 1$ will lead to the textural structures being dominated by noise. But even if we synthesize textures by taking the conditional mean of the final sampling iteration, as we do here, we see that γ can greatly affect the quality of the texture and that the previous best practice of $\gamma = 1$ does not work well. This may be the reason why other texture models considered more complex pixel covariances and/or rely on less well-mixing samplers for stabilizing training.

Although Fig. 3(a) suggests that smaller values of γ should be preferred, an overly small γ will lead to a small covariance for the Gaussian in Eq. (3) and consequently to slow mixing of the Gibbs sampler. We find $\gamma = 0.03$ to be a good trade-off. To illustrate this, Fig. 3(b) shows the covariance matrix computed from 1000 consecutive samples of \mathbf{h} corresponding to one feature. As it is close to a diagonal matrix, the variables in \mathbf{h} are approximately independent, thus the sampler mixes well. We use $\gamma = 0.03$ for all our experiments.

Other Best Practices. To obtain structured filters and – in our experience – also better texture models, we have to impose some constraints on the filters. As usual, the filters are initialized with random values, but their coefficient norms are ensured to be small initially. During training they are moreover constrained to have 0-mean and limited to not increase above an empirical threshold of $0.05/\gamma$. Otherwise the filters will often get stuck in poor local optima without any clear structure. Since the biases do not change significantly during learning, we fix them to $b_j = -\frac{1}{3}\|\mathbf{w}_j\|$, similar to [9]. The bias depends on the current norm of filter coefficients to keep a reasonable portion of the hidden units being “on” (*cf.* Eq. 7).

Also note that the typical whitening of the training data cannot be applied for textures, even if it is common for natural image priors. Since whitening will remove the major structural pattern of a single texture, it is in our experience difficult for the RBM to represent the remaining spatial patterns.

The Learned Models. We trained cGRBM models for several Brodatz textures, each of which is trained based on 40 patches of size 76×76 , randomly cropped from the corresponding preprocessed training image. As all the training images are rescaled, we simply fix the filter size to 9×9 for all models. The models for textures D6, D21 and D53 consist of 15 learned filters, while the model for D77 has 20 filters due to its slightly more complex pattern. Examples of the learned filters are shown in Fig. 2; we observe clearly apparent structure, *e.g.* unlike [5].

3.4 Generative Properties

To evaluate the generative performance of our learned cGRBM texture models, we quantitatively compute texture similarity scores (TSS) [5] between the synthesized textures and real texture patches, which is defined based on the maximum normalized cross correlation $TSS(\mathbf{s}, \mathbf{x}) = \max_i \frac{\mathbf{s}^T \mathbf{x}_{(i)}}{\|\mathbf{s}\| \cdot \|\mathbf{x}_{(i)}\|}$, where \mathbf{s} is the synthesized texture and $\mathbf{x}_{(i)}$ denotes the patch of the same size as \mathbf{s} within the texture image \mathbf{x} , at location i .

We collect 100 samples of size 76×76 for each model (each texture) using Gibbs sampling. Since in Gibbs sampling the texture samples are obtained by summing the final conditional mean and *i.i.d.* Gaussian noise, we use the conditional means from the last sampling step as the synthesized texture. For computing the TSS, only the center 19×19 pixels are considered (the same size as in [5,7,8]). Tab. 1 shows the means and standard deviations of TSS. Thanks to the full-convolutional weight sharing scheme, our simple cGRBM models only require 15 (or 20 for D77) features (filters) to exceed the generative properties of much more complex Boltzmann machine models [7,8] with many more (> 300) features. Note the considerable performance difference between our learned cGRBMs and the Gaussian RBM baseline “Tm” of [7], which is based on a standard learning procedure and tiled-convolutional weight sharing. Our cGRBMs even outperform the deep belief networks (DBNs) of [8]. Meanwhile, the 9×9 filter size of our models is also smaller than that of [7,8] (11×11). The BiFoE models [5] only use 9 filters of size 7×7 , but the paper argues that more and larger filters do not lead to a large difference in model quality, but greatly reduce the efficiency of inference.

3.5 Texture Inpainting

In a texture inpainting application, following previous work [7,8], we take 76×76 patches from the testing texture images and create a 54×54 square hole in the middle of each patch by setting the intensity values to zero. The task is to generate texture in the square hole that is consistent with the given boundary.

Table 1. Means and standard deviations of TSS of the synthesized textures

Model	D6	D21	D53	D77
BiFoE [5]	0.757 ± 0.059	0.871 ± 0.032	0.827 ± 0.087	0.646 ± 0.022
Tm [7]	0.930 ± 0.021	0.890 ± 0.079	0.849 ± 0.061	0.866 ± 0.008
TmPoT [7]	0.933 ± 0.036	0.896 ± 0.070	0.853 ± 0.056	0.870 ± 0.008
TssRBM [8]	0.937 ± 0.047	0.948 ± 0.025	0.941 ± 0.022	0.841 ± 0.012
DBN [8]	0.952 ± 0.016	0.947 ± 0.032	0.950 ± 0.026	0.864 ± 0.160
cGRBM (ours)	0.963 ± 0.005	0.961 ± 0.008	0.965 ± 0.004	0.875 ± 0.013

Table 2. Means and standard deviations of MSSIM scores of the inpainted textures

Model	D6	D21	D53	D77
Tm [7]	0.858 ± 0.016	0.866 ± 0.019	0.849 ± 0.023	0.764 ± 0.027
TmPoT [7]	0.863 ± 0.018	0.874 ± 0.012	0.860 ± 0.023	0.767 ± 0.032
TssRBM [8]	0.888 ± 0.023	0.912 ± 0.014	0.916 ± 0.024	0.763 ± 0.031
DBN [8]	0.889 ± 0.025	0.906 ± 0.016	0.924 ± 0.029	0.774 ± 0.023
cGRBM (ours)	0.909 ± 0.017	0.928 ± 0.012	0.933 ± 0.010	0.783 ± 0.027
Efros & Leung [2]	0.827 ± 0.028	0.801 ± 0.029	0.863 ± 0.018	0.632 ± 0.041
Deterministic (ours)	0.899 ± 0.019	0.918 ± 0.014	0.926 ± 0.016	0.775 ± 0.034

Inpainting is done through sampling conditioned on the given boundaries. This procedure is quite efficient when using a block Gibbs sampler. For each texture, we use 20 different inpainting frames and perform inpainting 5 times with different initializations, leading to 100 inpainting results. The quality of the inpainted texture is measured by the mean structural similarity (MSSIM) score [14] between the inpainted region and the ground truth. Fig. 4 shows examples of inpainting results and Tab. 2 gives a quantitative comparison with other models¹, which we outperform considerably despite a simpler model architecture.

4 Deterministic Texture Synthesizer

From Sec. 3 we know that the value for γ in our cGRBM model is small. Looking at Eq. (6), this on the one hand means that the sample will not deviate significantly from the conditional mean. Moreover, the norms of filter coefficients must be large to balance the small γ , which implies that most values of $\mathbf{w}_{jc}^T \mathbf{x} + b_j$ will fall outside of the smooth transition area of the logistic sigmoid in Eq. (7). This suggests that, in applications, it may be possible to use deterministic functions to replace sampling the two conditionals. In particular, we apply a unit step function on $\mathbf{w}_{jc}^T \mathbf{x} + b_j$, then use the obtained binary auxiliary variable to modulate the filters to reconstruct the image, and repeat the procedures until convergence (Alg. 1). Note that this is equivalent to a block coordinate descent on the model energy Eq. (5). Since this scheme only works well if some reference pixels are given, such as in texture inpainting, we use it in this context. While slightly worse than sampling the cGRBM, the performance of the deterministic approach is still

¹ Our implementation of Efros & Leung [2] uses a window size of 15×15 .

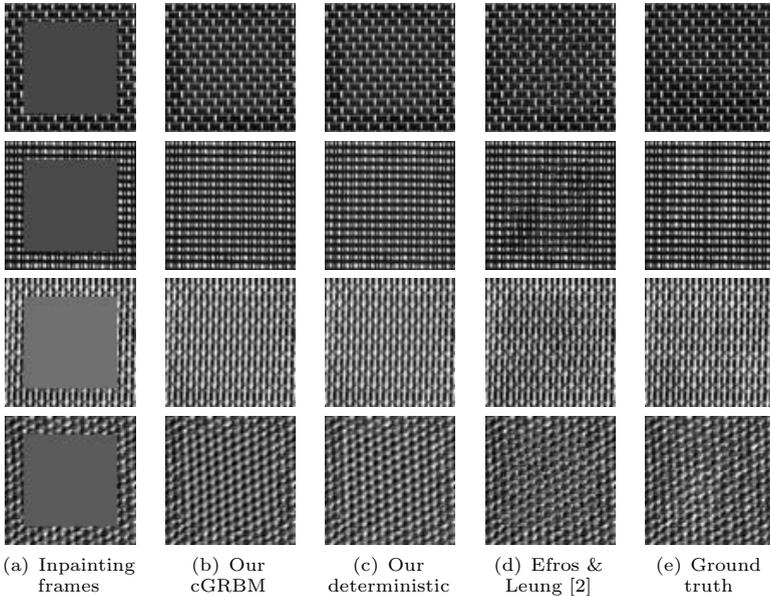


Fig. 4. Examples of inpainting results. From top to bottom: D6, D21, D53, D77.

better than the state of the art. In our inpainting experiment, our deterministic method only needed 30–50 iterations to reach convergence, while sampling the cGRBM usually required ~ 100 iterations. It is moreover quite efficient, because the computation in each iteration is very simple. By contrast, nonparametric methods (*e.g.* [2]) are often not as efficient due to the necessary matching step.

5 Summary

In this paper we analyzed the role of the conditional mean in modeling visual repetitive textures. We showed that simple Gaussian RBMs trained in a convolutional fashion are able to outperform much more complex state-of-the-art texture models, in which the latent variables also control the conditional covariance. We showed that the covariance of the cGRBM must actually be rather

Algorithm 1. Deterministic Texture Inpainting

Require: Image \mathbf{x} to be inpainted

repeat

$$h_{jc} \leftarrow H(\mathbf{w}_{jc}^T \mathbf{x} + b_j)$$

\triangleright where H is a unit step function

$$\mathbf{x} \leftarrow \gamma \sum_{j,c} h_{jc} \mathbf{w}_{jc}$$

until no change of \mathbf{x} (or \mathbf{h})

return \mathbf{x}

small to enable high texture quality, and suggest new best practices for modeling repetitive textures with RBMs. Our model requires only a small number of learned features, with a clearly emerging structure, and is spatially invariant. Inspired by efficient RBM inference using block Gibbs sampling, we further propose a fast, iterative deterministic texture synthesis method.

An open question for future work is to automatically determine the number of filters based on the complexity of the respective texture. Moreover, although a mean-only model can capture repetitive textures well, covariances might have to be considered for general textures. We leave this for future work.

References

1. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH 2001 (2001)
2. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: ICCV 1999, vol. 2, pp. 1033–1038 (1999)
3. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. *IEEE T. Pattern Anal. Mach. Intell.* 14(3), 367–383 (1992)
4. Hao, T., Raiko, T., Ilin, A., Karhunen, J.: Gated Boltzmann Machine in Texture Modeling. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) ICANN 2012, Part II. LNCS, vol. 7553, pp. 124–131. Springer, Heidelberg (2012)
5. Heess, N., Williams, C.K.I., Hinton, G.E.: Learning generative texture models with extended Fields-of-Experts. In: BMVC 2009 (2009)
6. Hinton, G.: A practical guide to training restricted Boltzmann machines. Tech. Rep. UTML TR 2010–003, University of Toronto (2010)
7. Kivinen, J.J., Williams, C.K.I.: Multiple texture Boltzmann machines. In: AIS-TATS (2012)
8. Luo, H., Carrier, P.L., Courville, A., Bengio, Y.: Texture modeling with convolutional spike-and-slab RBMs and deep extensions. In: AISTATS (2013)
9. Norouzi, M., Ranjbar, M., Mori, G.: Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In: CVPR 2009 (2009)
10. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE T. Image Process.* 12(11), 1338–1351 (2003)
11. Ranzato, M., Mnih, V., Hinton, G.E.: Generating more realistic images using gated MRF’s. In: NIPS 2010 (2010)
12. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on MRFs in low-level vision. In: CVPR 2010 (2010)
13. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In: ICML 2008 (2008)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE T. Image Process.* 13(4), 600–612 (2004)
15. Welling, M., Hinton, G.E., Osindero, S.: Learning sparse topographic representations with products of Student-t distributions. In: NIPS 2002, pp. 1359–1366 (2002)
16. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Comput. Vision* 27(2), 107–126 (1998)