Texel-Based Texture Synthesis with Bunch Sampling

Dongxiao Zhou and Georgy Gimel'farb

CITR, Tamaki Campus, Department of Computer Science,
The University of Auckland,
Auckland, New Zealand

Abstract. Texture is frequently considered as a repetitive spatial arrangement of texels, the primitive elements of texture. We define the texel as a bunch of image signals that has a particular geometric structure (shape and size). This provides for fast synthesis of a spatially homogeneous texture by bunch sampling. First, the structure of the texels and a placement grid to spatially arrange them are estimated from a training image using a generic Gibbs random field model of the texture. Then at the synthesis stage, the structure serves as a sampling mask to capture the texels from the training image. Random positions for replicating texels to form a synthetic large-size texture are selected according to the placement grid.

1 Introduction

Most of the known texture synthesis methods use Markov random field (MRF) models to describe spatially homogeneous textures with translation invariant signal statistics. With respect to sampling techniques, there are two main groups of such methods, namely, model-based probabilistic synthesis and non-parametric sampling. Model-based methods [1, 3, 4, 7, 17] construct an explicit MRF model of a given training image. The model is specified by a joint Gibbs probability distribution (GPD) of image signals. New textures are generated by iterative Markov Chain Monte Carlo (MCMC) probabilistic sampling of the GPD based pixel-wise stochastic relaxation e.g., the Gibbs or Metropolis sampler. These methods are successful in modelling a broad range of textures, but are inefficient in realistic texture synthesis because the stochastic relaxation is too computationally complex.

Non-parametric sampling [5, 6, 11] achieves much faster texture synthesis by avoiding slow pixel-wise relaxation. The MRF model of texture is assumed only implicitly, and the training texture itself is used as a sampling source. To preserve local features of the training texture, the sampling in [6], for instance, chooses sampling candidates by minimising the distance (e.g., in the L_2 metric) between the pixel neighbourhoods in the training image and the like neighbourhood in the synthetic texture. As each sampling candidate is copied to the synthetic texture, the pixel neighbourhood structure and thus the texture feature is expected to be inherited by the synthetic texture. However, without an explicit texture model,

Fig. 1. Brodatz’s training textures D4 and D34 (128×128), the corresponding MBIMs and the structure of texels. The MBIMs for the 65×65 search window are scaled up for better visual perception.

the non-parametric sampling lacks knowledge of internal structure of signal dependencies varying from texture to texture. This hinders adaptive selection of the shape and size of the pixel neighbourhood for a particular texture. Also, the distance-based pixel-wise copying may accumulate local errors. More stable block- or patch-based copying in [5, 11] results in false borders between adjacent permuted blocks (patches) and in verbatim replicas of each training singularity. Most of these problems are overcome by “smart copying” [12] that cuts from the training image arbitrary shaped patches forming a “borderless” synthetic texture rather than attempts to post-fix the false borders.

Our paper describes an alternative method, called bunch sampling in [9, 16], for realistic texture synthesis. It combines the strengths of both the model-based probabilistic synthesis and the non-parametric sampling to provide fast synthesis of large-size textures. Compared to [9, 16], this paper introduces more elaborated techniques for estimating the geometric structure of texels and their placement grid for each particular texture. The same bunch sampling is used here also to rectify grayscale and colour textures with at least weak translational periodicity. The single texel for a rectified prototype of the periodic texture is estimated using marginal posterior probabilities of the corresponding signals in all the superposed training bunches.

2 Bunch Sampling

Structural approach to texture analysis [10] considers a homogeneous texture as the result of regular and repetitive spatial arrangement of its primitive elements or texels. Bunch sampling synthesises textures in line with this idea, that is, a new texture is generated by permuting and replicating the texels. The replication is guided by a placement grid specifying the periodic spatial arrangement of texels in the texture. Experiments in [16] have shown that the bunch sampling forms realistic large-scale synthetic textures for two separate classes of images, namely, periodic or almost periodic mosaics and aperiodic stochastic textures. The classes have different notions of the texels and placement grids. Textures of the first class contain spatially uniform repetitive subimages (graphical patterns) that may be considered as noisy or transformed replicas of a single texel prototype or a few such prototypes. The grid puts strict limits on possible relative positions of the neighbouring texels. Textures of the second class has spatially uniform marginal distributions of signal co-occurrences in the bunches. The texel-based description is still possible, but now it involves a very large set of different texels, and there are no too strict limitations on their relative placement except for the image continuity (that is, the absence of “false borders”).

Loosely speaking, in both cases the texel is a bunch of image signals having a geometric structure with particular shape, orientation, and size. Due to assumed translational invariance, all the texels share the same geometric structure but may differ by particular signal combinations. Spatial repetitiveness results in non-uniform frequencies of signal co-occurrences. Thus the bunch structure can be approximately related to most characteristic pairwise gray level cooccurrences in a generic Gibbs random field (GGRF) model of translation invariant textures [7, 8]. Once that the structure is found for a given training image, the placement grid is specified accordingly in terms of relative spatial positions of the texels.

The GGRF texture model allows to rank families of translation invariant pixel pairs according to their partial interaction energies in a particular texture. Let $\mathbf{R}$ denote an arithmetic lattice supporting digital images. Let $\mathbf{C}_{\xi,\eta} = \{(x, y), (x + \xi, y + \eta) : (x, y) \in \mathbf{R}, (x + \xi, y + \eta) \in \mathbf{R}\}$ be a family of pixel pairs having the same relative spatial displacement. Then the first approximation of the partial energy for these pixel pairs in the training image $\mathbf{g}$ is proportional to the variance of the normalised grey level co-occurrence histogram (GLCH) collected over the family $\mathbf{C}_{\xi,\eta}$ [7, 8]:

$$E_{\xi,\eta}(\mathbf{g}) \propto (\mathbf{F}_{\xi,\eta}(\mathbf{g}) - \mathbf{F}_{\mathbf{IRF}}) \bullet \mathbf{F}_{\xi,\eta}(\mathbf{g}) \quad (1)$$

where $\mathbf{F}_{\xi,\eta}(\mathbf{g})$ and $\mathbf{F}_{\mathbf{IRF}}$ are the normalised GLCHs for the image and the independent random field, respectively, and $\bullet$ denotes the dot product.

The partial energy of Eq. (1) specifies the contribution of the family $\mathbf{C}_{\xi,\eta}$ to the overall interaction energy, that is, to the probability of the training image. The higher the partial energy, the more probable the image and the more characteristic the family itself for this texture. The top-rank partial energies specify the most characteristic structure of pairwise interactions [8].

The partial energies of all the families in a large search set of displacements, $\mathbf{W} = \{(\xi, \eta) : |\xi| \leq \Delta, |\eta| \leq \Delta\}$ of the size $(2\Delta + 1) \times (2\Delta + 1)$, specify a symmetric model based interaction map (MBIM) [8]. Points (ξ, η) and $(-\xi, -\eta)$ in the MBIM correspond to the family $\mathbf{C}_{\xi, \eta}$, and their scalar values represent the relative partial energy $E_{\xi, \eta}(\mathbf{g})$ in Eq. (1). Figure 1 demonstrates the MBIMs with the search window $\mathbf{W}$ of the size 65×65 for the Brodatz's stochastic texture D4 and periodic mosaic D34.

2.1 Geometric Structure of Texels

As shown in Fig. 1, the top-rank partial energies form specific clusters in the MBIMs. These most energetic clique families determine both statistical features of the texture and the geometric structure of the texels. Stochastic textures (e.g. D4) with dominant close-range pixel interactions have only one central cluster. The like central cluster for periodic textures such as D34 relates mainly to uniform background. The periodic structure of these textures is reflected by the peripheral clusters. Thus, the MBIM allows bunch sampling to distinguish between the aperiodic and periodic textures and estimate the geometric structure of texels using different strategies. For a stochastic texture, the central cluster in the MBIM specifies the shape of the texel (usually it is a connected region). For a periodic mosaic, the peak points of the peripheral clusters being the nearest neighbours to the MBIM's centre form the geometric structure of the texels. These texels are of the relatively small size, e.g., only six pixels for the texture D34. As shown later, even these simple structures are adequate for realistic texture synthesis.

The following algorithm is used to analyse MBIMs.

1. Find the threshold that separates the most energetic families of pixel pairs from others, using the algorithm described in [14]. The energy histogram for the MBIM is usually unimodal, with only one main peak at the lower end representing the majority of families with the low partial energy.
2. Segment the MBIM using the obtained threshold and identify the energy clusters using a general connected component labeling technique.
3. Find the peak of every cluster (it can be approximated by the gravity centre of the cluster if the partial energies are considered as the weights).

2.2 Placement Grid for Texels

Assuming that non-overlapping bunches are conditionally independent, bunch sampling tessellates the textured image with a guiding grid derived according to the estimated geometric structure of the texel. Each cell of the grid is a compact parallelogram enclosing the texel's structure. The parallelogram is specified with four parameters, $(\theta_x, \theta_y, m, n)$ where the angles θ_x and θ_y give the guiding orientation of the cell sides with respect to the image coordinate axes, and

the side sizes m and n are the maximum spans of the texel along the guiding directions(Fig. 3).The enclosing parallelogram is computed here using the Voss–Suesse’s algorithm in [15].

The guiding grid reflects the underlying structure that forms the repetitive pattern of a homogeneous texture. It defines a reference coordinate system so that the spatial dependencies between the texels are specified in terms of the relative positional shift of each texel with respect to the closest grid cell. As shown in Fig. 4, the relative shift, (δ_x, δ_y) , for the texel centred on point $p = \{x, y\}$ in the image is as follows:

$$\begin{aligned}\delta_x &= (y \cdot \sin \theta_y + x \cdot \cos \theta_x) \bmod m \\ \delta_y &= (y \cdot \cos \theta_y - x \cdot \sin \theta_x) \bmod n\end{aligned}\tag{2}$$

2.3 Texture Synthesis with Bunch Sampling

Given a training texture $\mathbf{g}^0$, texture synthesis begins with a blank goal image, $\mathbf{g}^1$, that can be of arbitrary size. For each pixel, p , in the goal image, bunch sampling computes its relative shift (δ_x, δ_y) with respect to the placement grid by Eq. (2), and then searches the training image for a proper sampling position p_s with the same relative shift. Usually there are multiple candidate sampling positions satisfying the condition, so one of them is chosen randomly. With p_s as the centre and the geometric texel structure as the sampling mask, a bunch of image signals is picked up from the training image, and then it is copied into the goal image $\mathbf{g}^1$ with the centre on p . During the synthesis, signal collisions happen when a new bunch has to be placed into an area that has been occupied by a previously placed bunch. A simple heuristic rule that preserves the already placed signals to resolve the collisions [9] results in the most of cases in realistic borderless textures.

The pixels in the goal image can be generated in an arbitrary order, and the texture synthesis completes after all the pixels are covered. The synthesis has linear computational complexity with respect to the size of the synthetic image $|\mathbf{g}^1|$. It depends linearly on the size of the bunch, too.

Fig. 2. Synthetic textures D4 and D34 (360×360).

Fig. 3. Parameters of the placement grid for the texture D34.

Fig. 4. Tessellation of D34 and relative positions of the bunches: $(0, 0)$ for Bunch a and (δ_x, δ_y) for Bunch b.

Fig. 5. Training Brodatz's and MIT VisTex textures 128×128 .

3 Results and Discussion

Bunch sampling has been tested on a variety of textures, mainly from the digitised Brodatz album [2] and the MIT VisTex texture database [13]. Figure 2 shows the synthetic textures D4 and D34 generated with the texels in Fig 1. Figure 6 shows synthetic textures for the training images in Fig. 5. In these cases, the synthetic and training textures have good visual similarity.

But deviations from periodicity may hinder the synthesis. Figure 7,a shows the visually unsatisfactory synthetic Brodatz's texture D3. Because the bunch sampling assumes the fixed geometric structure of each texel, it fails under local deformations of such a weakly homogeneous texture. As a result, random distortions appear in the synthetic texture due to mismatches between the estimated fixed structure and its local deformation at the sampling position.

However, in this case, the bunch sampling can at least rectify the distorted training image and produce its idealised periodic prototype. The approximate Bayesian estimate of a single texel based on the maximum marginal posterior (MMP) probabilities of signals in each pixel is easily obtained by superposing all

signals with the same relative shift with respect to the placement grid. Figure 7,b shows the resulting synthetic prototype. In spite of less noise and better visual appearance, it is far from a realistic D3-like texture. Such a “homogenised” prototype allows to recover local geometric deformations of the training image, and their spatial model can be later used to convert the prototype into a more realistic texture.

The GGRF model is usually restricted to only the second order GLCHs as its sufficient statistics. Moreover, to restrict the computational complexity we deal in practice with only 16 grey levels. Nevertheless, the bunch sampling still

Fig. 6. Examples of synthetic textures (360×360) generated with bunch sampling.

Fig. 7. Synthetic textures D3 (360×360) generated under different selection of texels.

handles colour textures fairly well due to the separate analysis and synthesis stages. At the analysis stage, a colour texture is reduced to the greyscale one for extracting the texel structure and placement grid. During the synthesis, the signals sampled from the original colour images are used to generate new textures. Figure 8 shows examples of the training and rectified synthetic colour textures.

4 Conclusions

Bunch sampling uses a “placement rule + texels” scheme to model homogeneous textures. The size, shape, and placement rule for texels are derived from image statistics of the sample texture using a GGRF model. During the synthesis, texels are sampled from the training texture and are directly moved to the synthetic one guided by the placement rule. In such a way, bunch sampling provides a fast model-based texture synthesis. However, bunch sampling is less efficient in handling inhomogeneous deformation in texture because the assumption of the uniform geometric structure for texels is too rigid for weakly homogeneous and inhomogeneous textures. Our future work is to relax the geometric structure of texels with some geometric adaption so that we can model the local deformation in those textures more effectively.

Fig. 8. Synthesis of colour textures, “windows” and “cans”.

Acknowledgements

This work was partially supported by the RSNZ Marsden Fund grant UOA122.

References

1. J. E. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, Vol. B48, pp. 259–302, 1986.
2. P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.
3. R. Chellappa. Two dimensional discrete Gaussian Markov random field models for image processing. In *Progress in Pattern Recognition*, Vol. 2, pp. 79–112. Elsevier Science Publishers B. V., 1985.
4. G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.5, pp. 25–39, 1983.
5. A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001, Computer Graphics Proceedings*, pp. 341–346. ACM Press / ACM SIGGRAPH, 2001.
6. A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. Int. Conf. on Computer Vision (ICCV'99), Kerkira, Corfu, Greece, 20–25 September 1999*, Vol. 2, pp.1033–1038, 1999.
7. G. L. Gimel'farb. Texture modeling with multiple pairwise pixel interactions. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, pp. 1110–1114, 1996.
8. G. L. Gimel'farb. *Image Textures and Gibbs Random Fields*. Dordrecht, Kluwer Academic Publishers, 1999.
9. G. L. Gimel'farb and D. Zhou. Fast synthesis of large-size textures using bunch sampling. In *Proc. Image and Vision Computing New Zealand (IVCNZ 2002), Auckland, New Zealand, 26–28 November 2002*, pp. 215–220, 2002.
10. R. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, Vol. 67, no.5, pp. 786–804, 1979.
11. L. Liang, C. Liu, and H. Y. Shum. *Real-time texture synthesis by patch-based sampling*. Technical Report MSR-TR-2001-40, Micorsoft Research, 2001.
12. A. Neubeck, A. Zalesny, and L. van Gool. Cut-primed smart copying. In *Proc. 3rd Int. Workshop "Texture 2003"*, Nice, France, October 17, 2003. Heriot-Watt Univ., pp. 119–124, 2003.
13. R. Picard, C. Graszyk, S. Mann, *et al.* *VisTex Database*. MIT Media Lab, Cambridge, Massachusettes, 1995.
14. P.L. Rosin. Unimodal thresholding. *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.
15. K. Voss and H. Suesse. Invariant fitting of planar objects with primitives. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, pp. 80–84, 1997.
16. D. Zhou and G. L. Gimel'farb. Bunch sampling for fast textutre synthesis.. In *Proc. Int. Conf. Computer Analysis of Images and Patterns, Groningen, The Netherlands, 25–27 August 2003. (Lecture Notes in Computer Science 2756)*, Springer-Verlag, Berlin, pp. 124–131, 2003.
17. S. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (FRAME): To a unified theory for texture modeling. *International Journal of Computer Vision*, Vol. 27, no. 2, pp. 107–126, 1998.