Texture Synthesis via a Noncausal Nonparametric Multiscale Markov Random Field

Rupert Paget, *Member, IEEE* and I. D. Longstaff, *Senior Member, IEEE*

Abstract— **Our noncausal, nonparametric, multiscale, Markov random field (MRF) model is capable of synthesising and capturing the characteristics of a wide variety of textures, from the highly structured to the stochastic. We use a multiscale synthesis algorithm incorporating *local annealing* to obtain larger realisations of texture visually indistinguishable from the training texture.**

Keywords— **Markov random fields, Nonparametric estimation, Texture synthesis, Multi-resolution, Local annealing.**

I. Introduction

We present here a method of modelling texture which enables synthesis of texture visually indistinguishable from training textures. Our noncausal, nonparametric multiscale Markov Random Field model captures the high-order statistical characteristics of textures. We propose that if a model is capable of synthesising texture visually indistinguishable from its training texture, then it has captured *all* the visual characteristics of that texture and must therefore be unique to that particular texture. Given a set of unique statistical models for a set of training textures, it may then be possible to use these models to segment and classify textures in images that contain a myriad of textures including unmodelled textures. Classification could be achieved by using these unique statistical models to determine the statistical similarity of a region in the image to a training texture. Any region where there was no statistical similarity could then be labelled as an *unknown texture*.

The conventional approach to classifying texture is to develop texture models from textural features until all textures in a training set are correctly classified [17]. Texture models are refined by choosing the most discriminatory features from the training textures via a feature selection process such as linear discriminatory analysis. However, when a new texture is added to the training set, the features selected may *not* be those appropriate for distinguishing the new texture from the previously modelled textures. Therefore, for each new texture, the selection process has to be repeated to obtain a new set of discriminatory features. Another limitation of conventional models is they cannot be applied to images containing textures other than those in the training sets. Therefore, they cannot be used to classify complex images such as Synthetic Aperture Radar images of Earth's terrain which contain a myriad of textures.

Current texture models such as fractal models, auto-models, autoregressive models, moving average models, autoregressive moving average models, do not realistically reproduce natural textures [9], such as those in the Brodatz album [4]. This implies the models do not capture *all* the visual characteristics. Julesz [11] hypothesised that third- or higher-order models were required to model natural textures. The MRF model has the required statistical order [1], but the parametric versions are inheritly inaccurate for modelling high-order statistical characteristics over a data sparse multi-dimensional feature space. This happens to be less of a problem for the nonparametric model.

Our nonparametric MRF model captures sufficient higher order statistical characteristics of texture to synthesise realistic textures from the Brodatz album. Based on the only criteria available to judge the success of a model – visual comparison of synthesised textures with training textures

The authors are affiliated with the Department of Electrical and Computer Engineering, University of Queensland and the Cooperative Research Centre for Sensor, Signal, and Information Processing.

Rupert Paget is at the Department of Electrical and Computer Engineering, University of Queensland, QLD 4072 AUSTRALIA, Tel: (+61) 7 365 3697, Fax: (+61) 7 365 3684, email: paget@elec.uq.edu.au

Dennis Longstaff is at the Department of Electrical and Computer Engineering, University of Queensland, QLD 4072 AUSTRALIA, Tel: (+61) 7 365 3871, Fax: (+61) 7 365 4999, email: idl@elec.uq.edu.au

– we conclude that our model captures *all* of the visual characteristics of a texture. (The synthesis process is facilitated by incorporating a novel local annealing function into a multiscale synthesis algorithm.) For any texture it is applied to, the model provides a unique statistical model of that texture. This means it can be used for texture classification of images containing textures not in the training set. Thus, the model may be able to extend the application of texture classification to images such as Synthetic Aperture Radar images.

II. Markov Random Field Texture Model

MRF models have been used for texture synthesis, region segmentation and image restoration [6]. The property of an MRF is that a variable X_s, at site s on a lattice $S = \{s = (i,j) : 0 \le i,j < M\}$, can be equal to any value $x_s \in \Lambda_s$, but the probability of $X_s = x_s$ depends upon values x_r at sites *neighbouring* s. The neighbouring sites are defined as those sites $r \in \mathcal{N}_s \subset S$, where $\mathcal{N}_s$ represents the *neighbourhood* of s. The *neighbourhood system* is the set of all neighbourhoods $\mathcal{N} = \{\mathcal{N}_s \subset S, s \in S\}$. The MRF is then defined by the *local conditional probability density function* (LCPDF) with respect to the neighbourhood system $\mathcal{N}$ [1], [7]:

$$\begin{aligned} P(X_s = x_s | X_r = x_r, r \neq s) = \\ P(x_s | x_r, r \in \mathcal{N}_s) \; s \in S, \; x_s \in \Lambda_s. \end{aligned} \quad (1)$$

To model a digital image as an MRF, consider each pixel in the image as a site s on a lattice S, and the grey scale value associated with the pixel equal to the value x_s. The site value x_s is then contained in the *state space* $\Lambda \doteq \{0, 1, 2, \ldots, L-1\}$, where L is the number of *grey levels* in the image. The *configuration space* Ω for the set of variables $\mathbf{X} = \{X_s, s \in S\}$ is the set of all possible images $\mathbf{x} = (x_1, x_2, \ldots, x_{M\times M})$, where

$$\mathbf{x} \in \Omega = \Lambda^{M\times M}. \quad (2)$$

Let Π be the (joint) probability measure on Ω with $\Pi(\mathbf{X} = \mathbf{x}) > 0 \; \forall \mathbf{x} \in \Omega$. Besag [1, p 195] proved that the joint distribution $\Pi(\mathbf{x})$ is uniquely determined by its LCPDF. The "Hammersley-Clifford theorem" [1], [6], also known as the "Markov-Gibbs equivalence theorem", gives form to the LCPDF so as to define a valid joint distribution $\Pi(\mathbf{x})$. The theorem requires a condition of the neighbourhood system $\mathcal{N} = \{\mathcal{N}_s, s \in S\}$ such that $s \in \mathcal{N}_t \Leftrightarrow t \in \mathcal{N}_s$. This means the neighbourhoods must be symmetrical for a homogeneous MRF. The symmetrical neighbourhood systems employed in this paper are those of Geman and Geman [7] for which the neighbourhood system $\mathcal{N}^o = \{\mathcal{N}^o_s, s = (i,j) \in S\}$ is defined as

$$\mathcal{N}^o_s = \{r = (k,l) \in S : 0 < (k-i)^2 + (l-j)^2 \le o\}, \quad (3)$$

where o refers to the order of the neighbourhood system. Neighbourhood systems for $o = 1, 2$ and 8 are shown in Figs. 1 (a), (b), and (c) respectively.

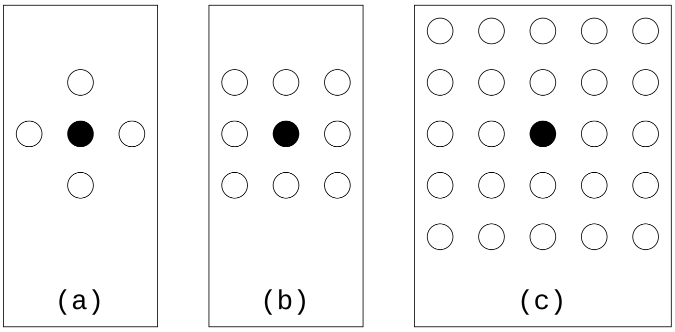

Fig. 1. Neighbourhoods. (a) The first order neighbourhood $o = 1$ or "nearest-neighbour" neighbourhood for the site $s = (i,j) =$ '•' and $r = (k,l) \in \mathcal{N}_s =$ '∘'; (b) second order neighbourhood $o = 2$; (c) eighth order neighbourhood $o = 8$.

III. Nonparametric MRF Model

Given a sample image $\mathbf{y} \in \Omega$ of a homogeneous texture, and a predefined neighbourhood system $\mathcal{N}$ defined on a lattice S_y, a nonparametric estimate of the LCPDF can be obtained by building a multi-dimensional histogram of $\mathbf{y}$. First, denote a pixel value as L_0, where $L_0 \in \Lambda$. Given that L_0 represents the pixel value at a site $p \in S_y$, denote pixel values $L_{n_r} \in \Lambda$ for each site $r \in \mathcal{N}_p$. The indices n_r are integers representing the relative position of r to p for which $1 \le n_r \le N$, where $N = |\mathcal{N}_p|$ is the number of sites in the neighbourhood $\mathcal{N}_p$. Then the set of pixel values $\{L_0, \ldots, L_N\}$ represents a realisation of a pixel and its neighbours irrespective of the pixel location $p \in S_y$.

Let $F(L_0, \ldots, L_N)$ denote the frequency of occurrence of the set of grey levels $\{L_0, \ldots, L_N\}$ in the image $\mathbf{y}$. The frequency is calculated from the image $\mathbf{y}$ as

$$F(L_0,\ldots,L_N) = \sum_{\substack{p\in S_y,\\ \mathcal{N}_p\subset S_y}} \delta(y_p - L_0) \prod_{r\in\mathcal{N}_p} \delta(y_r - L_{n_r}), \tag{4}$$

where δ is the Kronecker function. The set of frequencies $F(L_0,\ldots,L_N)\ \forall\ L_0,\ldots,L_N \in \Lambda$ forms the multi-dimensional histogram, where each $L_n, 0 \le n \le N$ is located on its own separate dimension (or axis) of the histogram. The total number of dimensions is the statistical order of the model and is equal to the size of the neighbourhood. The LCPDF is estimated from the multi-dimensional histogram as

$$\hat{P}(x_s|x_r, r\in\mathcal{N}_s) = \frac{F(L_0 = x_s, L_{n_r} = x_r, r\in\mathcal{N}_s)}{\sum_{L_0\in\Lambda} F(L_0, L_{n_r} = x_r, r\in\mathcal{N}_s)}. \tag{5}$$

An example of a two dimensional histogram for the neighbourhood system $\mathcal{N} = \{\mathcal{N}_s = \{s-1\}\}$ (Fig. 2(a)) is shown in Fig. 2(b).

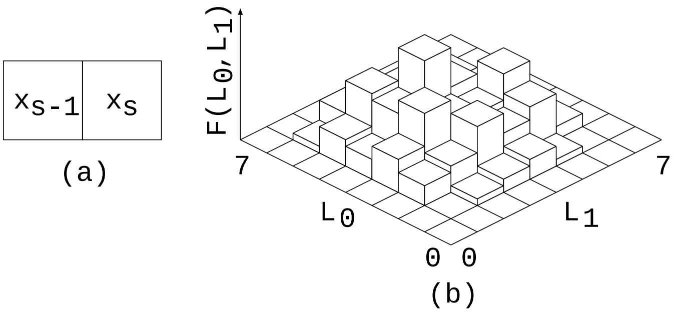

Fig. 2. Neighbourhood and its 2-D histogram

The true LCPDF needs to be estimated from the multi-dimensional histogram. When the sample data is sparsely dispersed over the histogram space, nonparametric estimation tends to be more reliable than parametric estimation if the underlying true distribution is unknown [15]. However with nonparametric estimation, the LCPDF may no longer define a valid joint distribution Π. Whereas with parametric estimation, the underlying true shape of the distribution may be compromised by trying to fit the parametric function to the data.

A. Parzen-Window Density Estimator

The Parzen-window density estimator [15] has the effect of spreading each sample datum into a smooth multi-dimensional histogram over a larger area. Denoting the sample data as $\mathbf{Z}_p = \mathrm{Col}[y_p, y_r, r\in\mathcal{N}_p]\ p\in S_y, \mathcal{N}_p \subset S_y$, the Parzen-window density estimated frequency $\hat{F}(\mathbf{z} = \mathrm{Col}[L_0 = x_s, L_{n_r} = x_r, r\in\mathcal{N}_s])$ of F in (5) is:

$$\hat{F}(\mathbf{z}) = \frac{1}{nh^d} \sum_{p\in S_y, \mathcal{N}_p\subset S_y} K\left\{\frac{1}{h}(\mathbf{z} - \mathbf{Z}_p)\right\}, \tag{6}$$

where n is the number of sample data $\mathbf{Z}_p$, h is the window parameter, and $d = |\mathcal{N}_s|+1$ is the number of elements in the vector $\mathbf{z}$ [15, p 76].

The shape of the smoothing is defined by the kernel function K. We chose K as the standard multi-dimensional Gaussian density function,

$$K(\mathbf{z}) = \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\mathbf{z}^{\mathbf{T}}\mathbf{z}). \tag{7}$$

The size of K is defined by the window parameter h. The aim is to choose h so as to obtain the best estimate of the frequency distribution $\hat{F}$ for the LCPDF. Silverman [15, p 85] provides an optimal window parameter:

$$h_{opt} = \sigma\left\{\frac{4}{n(2d+1)}\right\}^{1/(d+4)}, \tag{8}$$

where σ^2 is the average marginal variance. In our case, marginal variance is the same in each dimension and therefore σ^2 equals the variance associated with the one-dimensional histogram.

IV. Multiscale Texture Synthesis

There have been quite a few attempts to synthesise textures with a multiscale algorithm [12], but none have successfully reproduced natural textures [4]. Two noteworthy attempts were by Popat and Picard [14] and Heeger and Bergen [10]. Although the stochastic textures synthesised by Heeger and Bergen are impressive, their model only incorporated second order statistics which we believe to be inadequate for synthesising highly structured natural textures. Popat and Picard used higher-order statistics with much greater success. In fact, our approach is similar to theirs, but whereas their synthetic textures suffered from *phase discontinuity*, we used our method of *local annealing* to synthesise highly representative examples of natural textures.

We perform texture synthesis via a multiscale synthesis algorithm incorporating *local annealing* in the form of a novel pixel temperature function. As part of the synthesis process, the pixel temperature function reduces the dimensionality of the multi-dimensional histogram which, in turn, alleviates the problem associated with estimating the model in a high-dimensional space. This means we are able to use large neighbourhood systems to represent the texture.

We can synthesise a texture from an MRF model by a method known as stochastic relaxation (SR) [7]. This is done by starting with any image and iteratively updating pixels in the image with respect to the LCPDF. This generates a sequence of images $\{\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(n)\}$ with the property,

$$\lim_{n\to\infty} P(\mathbf{x}(n)|\mathbf{x}(0)) = \Pi(\mathbf{x}(n)) \quad \forall\ \mathbf{x} \in \Omega. \qquad (9)$$

A well-known SR algorithm is the Gibbs sampler [7]. Besag [2] also introduced a deterministic relaxation algorithm called *Iterative Conditional Modes* (ICM). Either algorithm is adequate for synthesising texture.

A. Multiscale Relaxation

A problem with the single-scale relaxation process is that global image characteristics evolve indirectly in the relaxation process [8], [16]. Global image characteristics are typically only propagated across the image lattice by local interactions and therefore evolve slowly, requiring long relaxation times to obtain equilibrium, as defined by equation (9). With multiscale relaxation (MR), we attempt to overcome this problem by implementing SR, first at a low resolution, and then at progressively higher resolutions [8]. The information obtained from SR at one resolution is used to constrain the SR at the next highest resolution. By this method, global image characteristics that have been resolved at a low resolution are infused into the relaxation process at the higher resolutions. This helps reduce the number of iterations required to obtain equilibrium [16]. MR also helps the ICM algorithm converge to an image closer to the global maximum of the joint distribution Π [3], [5].

The multiscale model may be best described by a multigrid representation of the image, as shown in Fig. 3. The grid at level $l = 0$ represents the image at the original resolution, where each intersection point '$\bullet$' is a site $s \in S$. The lower resolutions, or higher grid levels $l > 0$, are decimated versions of the image at level $l = 0$. The multigrid representation of the image $\mathbf{x}$ is then the set of images $\mathbf{x}^l$, for grid levels $l \geq 0$. The image $\mathbf{x}^l$ is defined on the lattice $S^l \subset S$, where

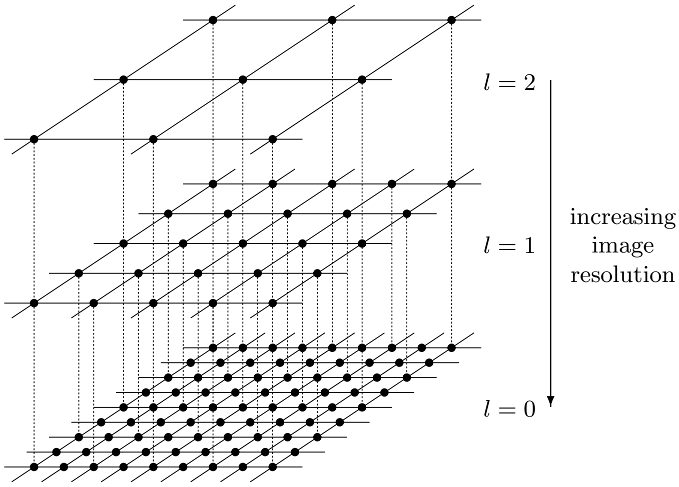

Fig. 3. Grid organisation for MR via decimation.

$$S^l = \left\{ s = (2^l i, 2^l j) : 0 \leq i, j < \frac{M}{2^l} \right\}. \qquad (10)$$

The set of sites S^l at level l represents a decimation of the previous set of sites S^{l-1} at the lower grid level $l - 1$. On this multiscale lattice representation, we need to redefine the neighbourhood system for each grid level $l \geq 0$ with respect to order o as

$$\mathcal{N}^l_{s=(2^l i, 2^l j)} = \left\{ r = (2^l p, 2^l q) \in S^l : 0 < (i-p)^2 + (j-q)^2 \leq o \right\}. \qquad (11)$$

We use the MR algorithm proposed by Gidas [8], which maintains the constraint imposed by the image $\mathbf{x}^{l+1}$ through the entire SR process at level l and proceeding levels. The constraint is such that, at any point through the SR process at level l, the image $\mathbf{x}^{l+1}$ may still be obtained from $\mathbf{x}^l$ by the multigrid representation. As we used decimation to form each grid level, the MR constraint is maintained at level l by *not* performing SR on those sites $s \in S^{l+1} \subset S^l$, as defined by equation (10).

To better incorporate the MR constraint into the SR process, we introduce a novel pixel temperature function, which defines how the MR constraint is imposed on the SR process. The function produces an equilibrium state which can be used to determine when to terminate the SR process at one level and start it at the next level. The novel pixel temperature function also reduces phase discontinuities [13].

B. Pixel Temperature Function

Our aim in incorporating the pixel temperature function is to define the degree of *confidence* that a pixel has the correct value. Each pixel is given its own temperature t_s, representing the confidence associated with the pixel x_s. The confidence is expressed as a value $0 \leq t_s \leq 1$, where 0 represents complete confidence, and 1 none at all. We have chosen this representation of the pixel temperature function to relate it to the global temperature function T, as used in stochastic annealing [7]. In fact, the function of our local pixel temperature may be regarded as an implementation of *local annealing* in the relaxation process.

In the MR algorithm, the confidence or temperature associated with each pixel is used to condition the LCPDF so as to be strongest on those pixels with $t_s = 0$, and weakest for those with $t_s = 1$. The pixel temperature is incorporated into the LCPDF by modifying the form of $(\mathbf{z} - \mathbf{Z}_p)$ in equation (6). Given that the LCPDF is estimated for an image $\mathbf{x}$ at the site $s \in S$, the vector $\mathbf{z} = \text{Col}[x_s, x_r, r \in \mathcal{N}_s]$. The sample data $\mathbf{Z}_p$ is taken from a sample image $\mathbf{y}$ defined on the lattice S_y, where $\mathbf{Z}_p = \text{Col}[y_p, y_r, r \in \mathcal{N}_p]$ $p \in S_y$, and

$$(\mathbf{z} - \mathbf{Z}_p) = \text{Col}[x_s - y_p, (x_r - y_r)(1 - t_r), r \in \mathcal{N}_s], \tag{12}$$

where the pixel temperature t_r is from the same site as the pixel value $x_r, r \in S$ for the image $\mathbf{x}$.

Before the SR algorithm starts at level l, those pixels which were relaxed at the previous level, $l + 1$, are given a pixel temperature $t_s = 0\ \forall\ s \in S^{l+1}$, *i.e.* complete confidence. The other pixels have their pixel temperatures initialised to $t_s = 1\ \forall\ s \notin S^{l+1}$, *i.e.* no confidence. After a pixel is relaxed, it has its temperature updated. We relate pixel temperature to pixel confidence, where pixel confidence is associated with the probability that x_s is the correct pixel value for the site s. Full pixel confidence occurs when x_s is sampled from an LCPDF at equilibrium, or when the LCPDF is completely conditional on its neighbouring pixel values. This occurs when $t_r = 0,\ \forall\ r \in \mathcal{N}_s$. The confidence associated with the pixel value x_s is then dependent on the pixel temperatures $t_r, r \in \mathcal{N}_s$. Therefore, the formula

$$t_s = \max\left\{0, \frac{\xi + \sum_{r \in \mathcal{N}_s} t_r}{|\mathcal{N}_s|}\right\},\ \xi < 0 \tag{13}$$

is used to describe the confidence associated with a pixel value x_s after it has been relaxed.

Initially, only those sites that had their values relaxed at the previous grid level are used in the LCPDF. However, as the SR iterations progress, more sites gain a degree of confidence. When $t_s = 0,\ \forall\ s \in S^l$, we can say the SR process has reached an equilibrium state, indicating that the image can be propagated to the next lower grid level. The process is then repeated at the next grid level $l-1$ and so on until the final grid level is reached.

C. Parallel Implementation

In our experiments, we synthesised textures on a multiprocessor machine, DECmpp 12000 (MasPar) ®, with 16384 processors in a parallelised array. This is very useful for image processing applications, as each processor in the parallelised array can be dedicated to a single pixel in the image. Up to 16384 pixels can be relaxed in one iteration. This is an advantage when applying SR with our nonparametric LCPDF, as the LCPDF has to be derived directly from the sample data for each pixel iteration, which in itself is computationally intensive.

A relaxation algorithm may be parallelised if the relaxation of a single pixel is conditionally independent of other pixels. Only those sites

$$S_{i.i.d.} = \{s \in S : r \notin S_{i.i.d.}\ \ \forall\ r \in \mathcal{N}_s\} \tag{14}$$

should be simultaneously relaxed. In other words, no neighbouring sites should be simultaneously relaxed. If all sites were simultaneously relaxed, Besag [2] suggests that oscillations in the site representation may result. In fact for the Ising model [1], we found simultaneous relaxation of all sites to be detrimental.

```
Algorithm 1: Nonparametric multiscale MRF texture synthesis
Input:
  $\mathbf{Y} \leftarrow$ the textured image to be modelled
  $M_y \times M_y \leftarrow$ size of image $\mathbf{Y}$
  $M_x \times M_x \leftarrow$ size of synthetic image $\mathbf{X}$
  $o \leftarrow$ the order of the neighbourhood system
begin
1. Define number of grid levels $M \leq 1 + \log_2 (\min \{M_x, M_y\})$.
2. Define image $\mathbf{X}$ as being on a set of sites $S = \{s = (i, j) : 0 \leq i, j < M_x\}$.
3. Define the multigrid representation of image $\mathbf{X}$ as the set of subset of sites $S^l \subseteq S$ for $0 \leq l < M$ as given by (10).
4. Similarly, define image $\mathbf{Y}$ as being on a set of sites, $S_y = \{p = (i, j) : 0 \leq i, j < M_y\}$, with a multigrid representation as the set of subset of sites $S_y^l \subseteq S_y$ for $0 \leq l < M$ as given by (10).
5. Initialise pixel temperatures $t_s = 1, \ \forall \ s \in S$.
6. For $l = M - 1$ to 0 do
6.1 Define neighbourhood $\mathcal{N}_s^l$ w.r.t. order $o$ as given by (11).
6.2 While $t_s \neq 0, \ \forall \ s \in S^l$ do
6.2.1 choose a set of i.i.d. sites $S_{i.i.d.} \subset S^l$ from (14) for which $t_s > 0$.
6.2.2 For all $s \in S_{i.i.d.}$ in parallel do
6.2.2.1 Estimate the LCPDF for $x_s$ via (6), with $(\mathbf{z} - \mathbf{Z}_p)$ defined by (12).
6.2.2.2 Choose a new $x_s$ by sampling its LCPDF, as in the Gibbs Sampler, or ICM algorithm.
6.2.2.3 Update $t_s$ via (13).
6.2.3 done
6.3 done
7. done
end
```

Fig. 4. Nonparametric multiscale MRF texture synthesis algorithm

V. Results

The textures presented in Figs 5 and 6 were synthesised with the multiscale texture synthesis algorithm outlined in Fig. 4. In Fig. 5 we show the progressive realisations for the MR algorithm on a Brodatz texture at each grid level. This shows how the MR algorithm infuses the global to the local characteristics of a sample texture into a synthetic texture. Fig. 6 demonstrates the wide range of textures – from the stochastic to the well structured – that we were able to synthesise. Best results for the structured textures were obtained with the higher order neighbourhood $\mathcal{N}^{18}$. In all cases the sample texture images were of size 128×128 pixels which we used to estimate the LCPDF from which we synthesised images of size 256×256. In this way, we confirmed that the characteristics of the texture from the sample image had indeed been captured by the model.

VI. Conclusion

We have shown that the nonparametric MRF model presented here can synthesise complex textures ranging from the stochastic to the well structured. Although an excellent technique for synthesising texture, its application may be limited for now by high computational load.

The range of textures synthesised, and the visual similarity of synthesised to sample textures, indicates that the nonparametric MRF model captures *all* of the visual characteristics of textures. However, it may also capture superfluous noise characteristics. To perform segmentation and classification of images containing unknown textures, the model should only use those characteristics which identify particular texture classes. Otherwise, it will separate out textures with different noise characteristics that are of the same texture class. This is a problem we will address with future work.

VII. Acknowledgement

The authors would like to thank the University of Melbourne for the long computer times on the DECmpp 12000 (MasPar) ®. The authors would also like to thank the following people for contribution to research: Christine Graffigne, from the Université de Paris-Sud, for advise on Multiscale analysis; and Garry Newsam and David Howard, from the Centre for Sensor Signal and Information Processing (CSSIP), for advice on Markov random fields.

References

[1] Julian E. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society, series B*, vol. 36, pp. 192–326, 1974.

[2] Julian E. Besag, “On the statistical analysis of dirty pictures,” *Journal Of The Royal Statistical Society*, vol. B–48, pp. 259–302, 1986.

[3] Charles Bouman and Bede Liu, “Multiple resolution segmentation of textured images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 99–113, 1991.

[4] P. Brodatz, *Textures – a photographic album for artists and designers*, Dover Publications Inc., New York, 1966.

[5] Haluk Derin and Chee-Sun Won, “A parallel image segmentation algorithm using relaxation with varying neighbourhoods and its mapping to array processors,” *Computer Vision, Graphics, and Image Processing*, vol. 40, pp. 54–78, 1987.

[6] D. Geman, “Random fields and inverse problems in imaging,” in *Lecture Notes in Mathematics*, vol. 1427, pp. 113–193. Springer–Verlag, 1991.

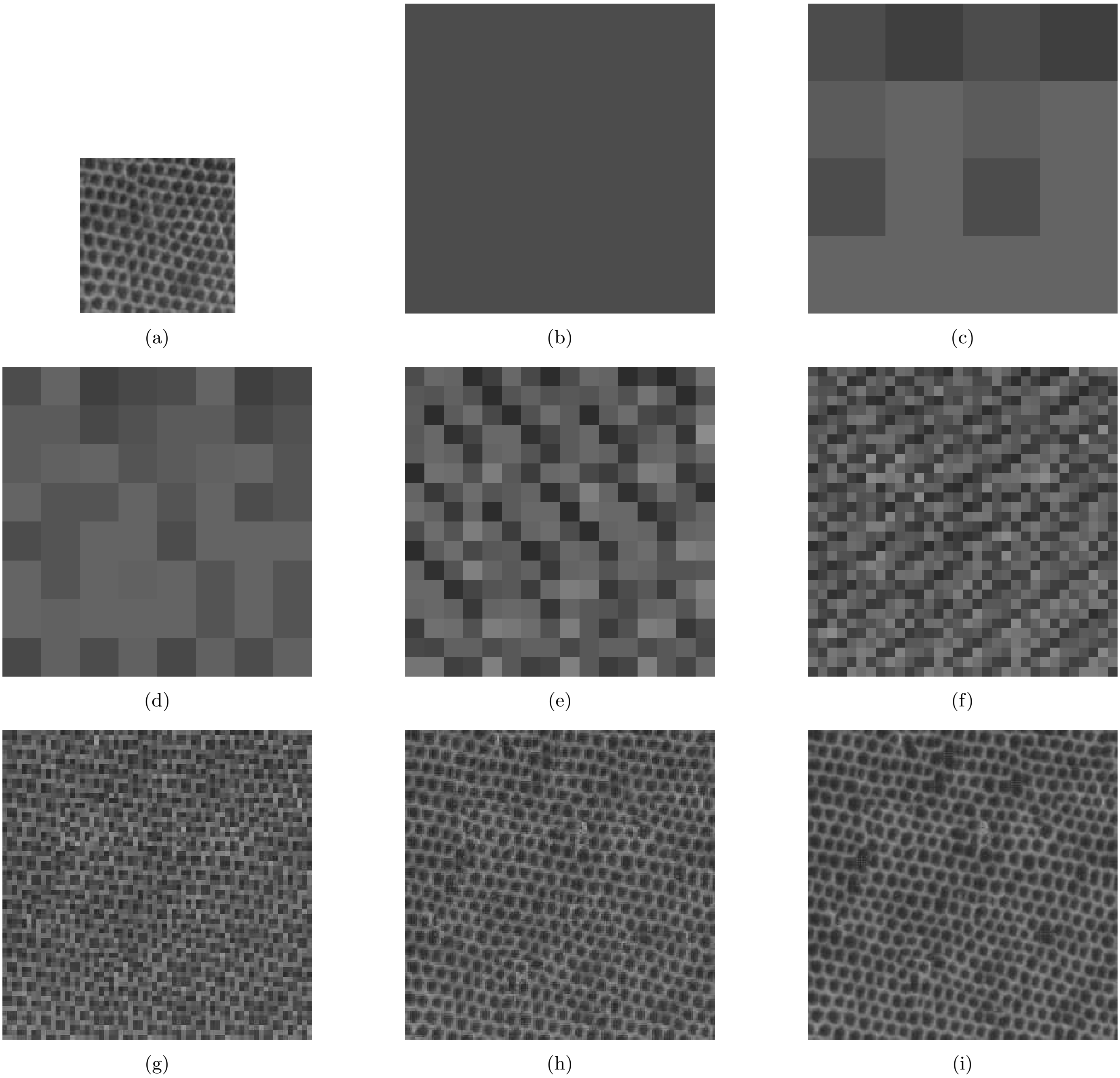

Fig. 5. Multiscale texture synthesis of Brodatz D22 (reptile skin) with neighbourhood order $c = 8$. (a) Original textured image; (b) Level 7; (c) Level 6; (d) Level 5; (e) Level 4; (f) Level 3; (g) Level 2; (h) Level 1; (i) Level 0.

[7] Stuart Geman and Donald Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[8] Basilis Gidas, "A renormalization group approach to image processing problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 164–180, 1989.

[9] Michal Haindl, "Texture synthesis," *CWI Quarterly*, vol. 4, pp. 305–331, 1991.

[10] David J. Heeger and James R. Bergen, "Pyramid–based texture analysis/synthesis," in *Proceedings ICIP–95: 1995 International Conference on Image Processing*, Washington, D.C., 1995, pp. 648–651.

[11] Bela Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, pp. 91–97, Mar. 1981.

[12] Rafael Navarro and Javier Portilla, "Robust method for texture synthesis–by–analysis based on a multiscale Gabor scheme," in *SPIE Electronic Imaging Symposium, Human Vision and Electronic Imaging '96*, Bernice Rogowitz and Jan Allebach, Eds., San Jose, Calfornia, 1996, vol. 2657, pp. 86–97.

[13] Rupert Paget and Dennis Longstaff, "A nonparametric multiscale Markov random field model for synthesising natural textures," in *Fourth International Symposium on*

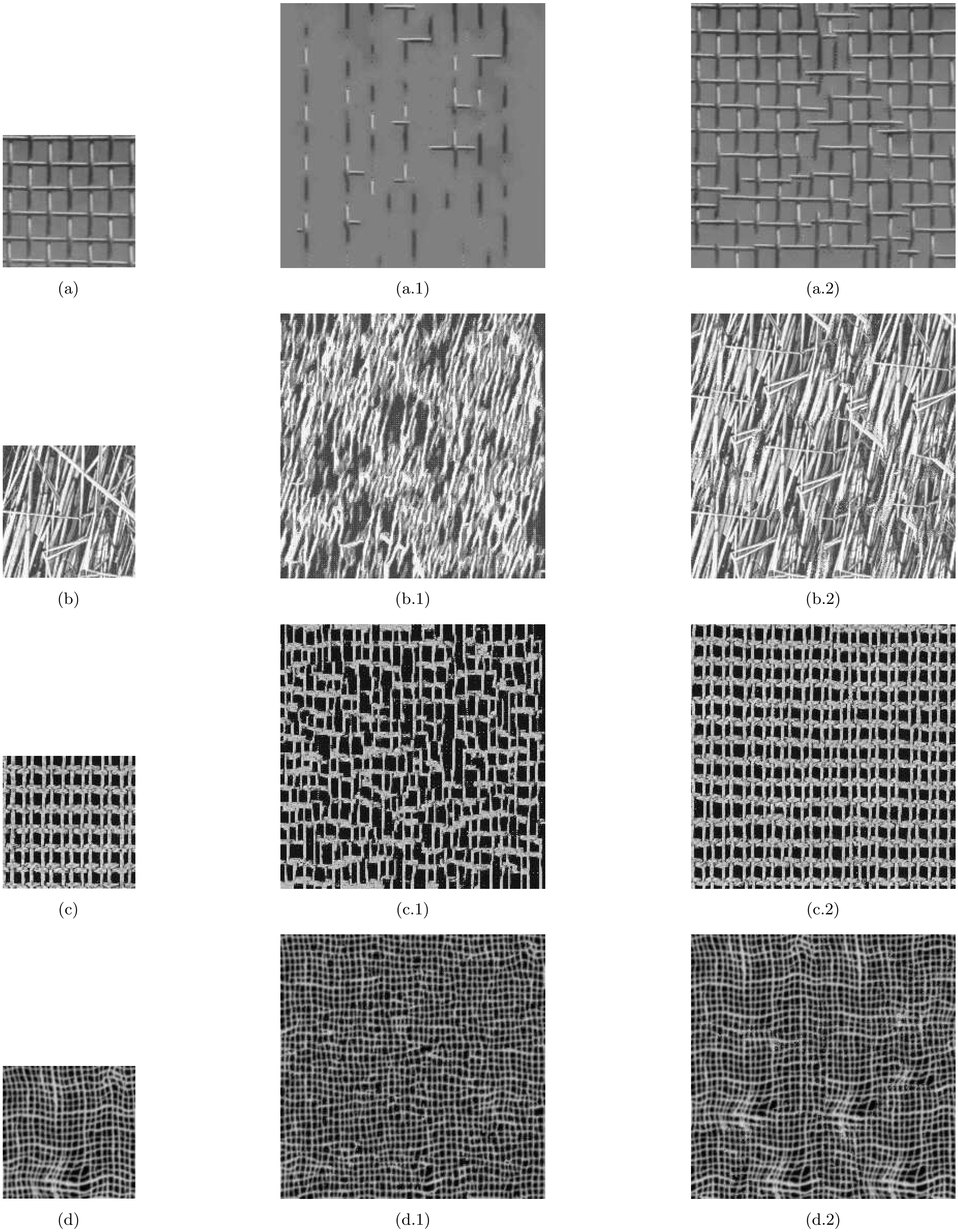

Fig. 6. Brodatz textures: (a) D1 - aluminium wire mesh; (b) D15 - straw; (c) D20 - magnified French canvas; (d) D103 - loose burlap; (?.1) synthesised textures - using neighbourhood order $o = 8$; (?.2) synthesised textures - using neighbourhood order $o = 18$.

Signal Processing and its Applications, Gold Coast, Australia, Aug. 1996, vol. 2, pp. 744–747, ISSPA 96, `http://www.vision.ee.ethz.ch/~rpaget`.

[14] Kris Popat and Rosalind W. Picard, "Novel cluster–based probability model for texture synthesis, classification, and compression," in *Proceedings SPIE visual Communications and Image Processing*, Boston, 1993.

[15] B. W. Silverman, *Density estimation for statistics and data analysis*, Chapman and Hall, London, 1986.

[16] Demetri Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 2, pp. 129–139, 1986.

[17] Mihran Tuceryan and Anil K. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds., pp. 235–276. World Scientific, Singapore, 1993.